Pawan Lingras Marcin Wolski
Chris Cornelis Sushmita Mitra
Piotr Wasilewski (Eds.)

Rough Sets and Knowledge Technology

LNAI 8171

8th International Conference, RSKT 2013
Halifax, NS, Canada, October 2013
Proceedings

Springer

Lecture Notes in Artificial Intelligence 8171

Subseries of Lecture Notes in Computer Science

Pawan Lingras Marcin Wolski
Chris Cornelis Sushmita Mitra
Piotr Wasilewski (Eds.)

Rough Sets and Knowledge Technology

8th International Conference, RSKT 2013
Halifax, NS, Canada, October 11-14, 2013
Proceedings

Springer

Volume Editors

Pawan Lingras
Saint Mary's University, Halifax, NS, Canada
E-mail: pawan@cs.smu.ca

Marcin Wolski
Maria Curie-Skłodowska University, Lublin, Poland
E-mail: maarten.wolski@gmail.com

Chris Cornelis
University of Granada, Spain
E-mail: chriscornelis@ugr.es

Sushmita Mitra
Indian Statistical Institute, Kolkata, India
E-mail: sushmita@isical.ac.in

Piotr Wasilewski
University of Warsaw, Poland
E-mail: piotr@mimuw.edu.pl

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-41298-1 e-ISBN 978-3-642-41299-8
DOI 10.1007/978-3-642-41299-8
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013949005

CR Subject Classification (1998): I.2, H.2.8, G.1, I.5, F.4, I.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 8th Rough Sets and Knowledge Technology (RSKT) conference, which, along with the 14th international conference Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC), 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 RSKT and RSFDGrC, JRS 2013 also hosted the 4th Rough Set Theory Workshop (RST) and the Rough Set Applications Workshop (RSA), both held on October 10, 2013.

The RSKT conference series is a meeting for academic researchers and industry practitioners interested in knowledge-related technologies. It primarily aims at providing state-of-the-art scientific results, encouraging academic and industrial interactions, and promoting collaborative research in rough set theory and its applications to knowledge technology problems. The RSKT conference has taken place annually since 2006, when the first conference was organized in Chongqing, China. It has provided an important international forum for discussion of current research trends, exchange of ideas, original research results and development experience so as to make further progress in the fields of data mining, knowledge discovery, and knowledge-based systems.

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 the papers accepted for the conference RSKT 2013 and the invited papers of historical character written by the leading researchers in the field (including Hiroshi Sakai, Michinori Nakata, Yiyu Yao, JingTao Yao, Jerzy Grzymała-Busse, Guoyin Wang, and Michael Wong). The proceedings are enriched by a contribution from Marcin Szczuka, who gave one of the JRS tutorials. 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 who assured the high standards of accepted papers. We hope that the resulting proceedings are evidence of the high-quality and exciting RSKT 2013 program. This program also included two

special/thematic sessions: History and Future of Rough Sets (invited papers), and Three-Way Decisions and Probabilistic Rough Sets (regular papers).

We would like to express our gratitude to the special session chairs (Hong Yu, Bing Zhou, Dun Liu, Fan Min, Xiuyi Jia, Huaxiong Li) 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 (Yiyu Yao, Dominik Ślęzak, Guoyin Wang, Davide Ciucci, Yuhua Qian, Masahiro Inuiguchi, 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. for being the industry sponsor of the entire event.

Our immense gratitude goes once again to Davide Ciucci for his invaluable help and support throughout the preparation of this volume and the conference RSKT 2013.

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

Pawan Lingras
Marcin Wolski
Chris Cornelis
Sushmita Mitra
Piotr Wasilewski

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 Yao

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

Tutorial

Using Domain Knowledge in Initial Stages of Knowledge Discovery in Databases 1
Marcin Szczuka

History and Future of Rough Sets

Non-deterministic Information in Rough Sets: A Survey and Perspective 7
Hiroshi Sakai, Mao Wu, Naoto Yamaguchi, and Michinori Nakata

Granular Computing and Sequential Three-Way Decisions 16
Yiyu Yao

A Scientometrics Study of Rough Sets in Three Decades 28
JingTao Yao and Yan Zhang

Generalizations of Approximations 41
Patrick G. Clark, Jerzy W. Grzymała-Busse, and Wojciech Rząsa

Expression and Processing of Uncertain Information 53
Guoyin Wang, Changlin Xu, and Hong Yu

Early Development of Rough Sets - From a Personal Perspective 66
S.K.M. Wong

Foundations and Probabilistic Rough Sets

Contraction to Matroidal Structure of Rough Sets 75
Jingqian Wang and William Zhu

Optimal Approximations with Rough Sets 87
Ryszard Janicki and Adam Lenarčič

Partial Approximation of Multisets and Its Applications in Membrane Computing 99
Tamás Mihálydeák and Zoltán Ernő Csajbók

A Formal Concept Analysis Based Approach to Minimal Value Reduction 109
Mei-Zheng Li, Guoyin Wang, and Jin Wang

Comparison of Two Models of Probabilistic Rough Sets 121
Bing Zhou and Yiyu Yao

Empirical Risk Minimization for Variable Precision Dominance-Based Rough Set Approach 133
Yoshifumi Kusunoki, Jerzy Błaszczyński, Masahiro Inuiguchi, and Roman Słowiński

Formulating Game Strategies in Game-Theoretic Rough Sets 145
Nouman Azam and JingTao Yao

Rules, Reducts, Ensembles

Sequential Optimization of Approximate Inhibitory Rules Relative to the Length, Coverage and Number of Misclassifications 154
Fawaz Alsolami, Igor Chikalov, and Mikhail Moshkov

Robustness Measure of Decision Rules 166
Motoyuki Ohki and Masahiro Inuiguchi

Exploring Margin for Dynamic Ensemble Selection 178
Leijun Li, Qinghua Hu, Xiangqian Wu, and Daren Yu

Evaluation of Incremental Change of Set-Based Indices 188
Shusaku Tsumoto and Shoji Hirano

Recent Advances in Decision Bireducts: Complexity, Heuristics and Streams 200
Sebastian Stawicki and Dominik Ślęzak

Studies on the Necessary Data Size for Rule Induction by STRIM 213
Yuichi Kato, Tetsuro Saeki, and Shoutarou Mizuno

Applying Threshold SMOTE Algorithm with Attribute Bagging to Imbalanced Datasets 221
Jin Wang, Bo Yun, Pingli Huang, and Yu-Ao Liu

New Trends in Computing

Parallel Reducts: A Hashing Approach 229
Minghua Pei, Dayong Deng, and Houkuan Huang

A Parallel Implementation of Computing Composite Rough Set Approximations on GPUs 240
Junbo Zhang, Yun Zhu, Yi Pan, and Tianrui Li

GPU Implementation of MCE Approach to Finding Near Neighbourhoods 251
Tariq Alusaifeer, Sheela Ramanna, Christopher J. Henry, and James Peters

FPGA in Rough Set Based Core and Reduct Computation 263
Tomasz Grześ, Maciej Kopczyński, and Jarosław Stepaniuk

Fast Approximate Attribute Reduction with MapReduce 271
Ping Li, Jianyang Wu, and Lin Shang

Three-Way Decision Rough Sets

Three-Way Decision Based Overlapping Community Detection 279
Youli Liu, Lei Pan, Xiuyi Jia, Chongjun Wang, and Junyuan Xie

Three-Way Decisions in Dynamic Decision-Theoretic Rough Sets 291
Dun Liu, Tianrui Li, and Decui Liang

A Cluster Ensemble Framework Based on Three-Way Decisions 302
Hong Yu and Qingfeng Zhou

Multistage Email Spam Filtering Based on Three-Way Decisions 313
Jianlin Li, Xiaofei Deng, and Yiyu Yao

Cost-Sensitive Three-Way Decision: A Sequential Strategy 325
Huaxiong Li, Xianzhong Zhou, Bing Huang, and Dun Liu

Two-Phase Classification Based on Three-Way Decisions 338
Weiwei Li, Zhiqiu Huang, and Xiuyi Jia

A Three-Way Decisions Model Based on Constructive Covering Algorithm 346
Yanping Zhang, Hang Xing, Huijin Zou, and Shu Zhao

Learning, Predicting, Modeling

A Hierarchical Statistical Framework for the Extraction of Semantically Related Words in Textual Documents 354
Weijia Su, Djemel Ziou, and Nizar Bouguila

Anomaly Intrusion Detection Using Incremental Learning of an Infinite Mixture Model with Feature Selection 364
Wentao Fan, Nizar Bouguila, and Hassen Sallay

Hybridizing Meta-heuristics Approaches for Solving University Course Timetabling Problems 374
Khalid Shaker, Salwani Abdullah, Arwa Alqudsi, and Hamid Jalab

Weight Learning for Document Tolerance Rough Set Model 385
Wojciech Świeboda, Michał Meina, and Hung Son Nguyen

A Divide-and-Conquer Method Based Ensemble Regression Model for Water Quality Prediction 397
Xuan Zou, Guoyin Wang, Guanglei Gou, and Hong Li

A Self-learning Audio Player That Uses a Rough Set and Neural Net Hybrid Approach 405
Hongming Zuo and Julia Johnson

Author Index 413

Using Domain Knowledge in Initial Stages of Knowledge Discovery in Databases

Tutorial Description

Marcin Szczuka*

Institute of Mathematics, The University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
szczuka@mimuw.edu.pl

Abstract. In this tutorial the topic of data preparation for Knowledge Discovery in Databases (KDD) is discussed on rather general level, with just few detailed descriptions of particular data processing steps. The general ideas are illustrated with application examples. Most of examples are taken from real-life KDD projects.

Keywords: KDD, data cleansing, data quality, data mining.

1 Introduction

The process of Knowledge Discovery in Databases (KDD) is traditionally presented as a sequence of operations which, applied iteratively, lead from the raw input data to high-level, interpretable and useful knowledge. The major steps in KDD process are typically: Selection, Preprocessing, Transformation, Data Mining (DM), and Iterpretation/evaluation. The tutorial is focused on first three of these steps. The goal is to demonstrate how we can improve the entire KDD process by using background (domain) knowledge in these phases.

During the selection and preprocessing phases of the KDD cycle the original raw data is sampled, cleansed, normalized, formatted and stored in a convenient way. The original, raw data is first turned into target data (selection) and then converted into preprocessed, analytic data (preprocessing). At this point we already have the data ready for mining and analysis, however, further transformation may be required, if the data mining and analysis algorithms are to run

* The author is 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) - grants: 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" and 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". The applications presented as examples in the tutorial were developed as a part of implementation of the mentioned research programmes.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 1–6, 2013.

efficiently. By utilizing various kinds (layers) of knowledge about the problem, the nature and structure of data, the objective, and available computational tools, we want to improve both the processing speed and the overall quality of the KDD results. In general case, not much can be done to optimize the quality of data mining step beforehand, since the knowledge needed to do that is not discovered yet. However, in particular applications we can at least prepare the transformed data for data mining algorithms in such a way that computational effort needed to manage data and obtain results is decreased and the chance to discover meaningful knowledge is increased.

In the tutorial the general task of data preparation for data mining is narrowed down to cases that meet some additional criteria. We assume, that it is necessary (required) to use data representation that involves creation and processing (usage) of compound (complex) data objects. Such a complex data object can be a structured text (document), a set of images, and so on. The main feature that defines such object is the existence of internal, non-trivial structure that can be used to preprocess and transform data entity for the purposes of data mining algorithms. Another condition for the problem to fit our scheme is the complexity of the problem as a whole. We want to address situations such that there is a room for significant improvement. Therefore, we are mostly interested in using knowledge to deal with data sets that are large and/or complicated. Last, but not the least, we mostly (but not exclusively) deal with situations, when storage and processing of data entities involves Relational Database Management System (RDBMS). The use of RDBMS imposes some additional constraints, but at the same time provides more tools for data manipulation.

The tutorial describes experiences in constructing and using large data warehouses to form a set of hints (guidelines) for a practitioner who needs to deal with tasks that require storing and processing of big data represented with use of compound (complex) data objects. It provides some insights into the ways of utilizing various kinds of knowledge about the data and the application domain in the process of building a data-warehouse-based solution. Several examples of practical projects are used to demonstrate what kind of knowledge and how, can be utilized to improve data processing in KDD process when compound data objects are involved. An explanation is provided about the kinds of compound/complex objects one may encounter. One of major steps is the presentation of general approach (framework) used to characterize data processing tasks by the way they handle such compound objects.

2 Organization of the Presentation

The tutorial is organized as follows. First, it is explained what constitutes a complex data object, what kinds of operations we want to perform and what improvements we want to achieve. Then, it is demonstrated how the knowledge can used to improve (optimize) data processing at initial stages of KDD process. Next, an illustration of the proposed approach using several examples of practical projects (see [1–4]) is provided. The subsections below correspond to main parts of the tutorial and provide few more details about the content.

2.1 Compound/Complex Objects in KDD Process

Storage and processing of data entities representing compound objects, with use of domain knowledge, needs to be considered in many aspects. To begin with, by an *object* we understand any element of the real world that can be stored as a data object (database entity) represented using a chosen, formal *ontology*. We assume that the ontologies used to define (construct) objects are given as a part of the domain knowledge. Now, a *compound object* is an object that combines several (at least two) objects into an ontology-definable object. Compound objects can be characterized by two crucial properties. Firstly, a compound object can always be decomposed into at least two parts and each of these parts is a proper ontology-definable object. Secondly, all components that make the compound object are bound by relation(s) from ontology. In other words, the compound object is something more than just a container, it has an internal structure.

A compound object as a whole may posses certain properties that are specific for a given domain (given context). It may also be related to other compound objects, not necessarily from the same domain. Using these relations we may construct more compound objects from existing ones. Properties and attribute values of a compound object may also be derived by examining its structure and sub-objects it contains in relation to other objects, e.g., by measuring the amount of common sub-objects.

To select, store, preprocess and transform data that contains compound objects, so that they can be used in further steps of KDD process one has to consider the most probable data processing scenarios that we will have to perform. Then, we have to design data structures and algorithms in such a way that they are efficient and produce high quality output. At this point, using all available domain knowledge may be crucial for the overall success of KDD. Since we usually have to facilitate the storage and processing that includes both data entities and relations the choice of RDBMS technology comes quite naturally. Since at the same time we are aiming at really complex KDD tasks, that are usually accompanied by large amounts of data, we rely on technologies that are dedicated for use in large data warehouses.

2.2 Outline of the Framework for Domain Knowledge Incorporation

We claim that the use of domain knowledge in the process of designing and using data structures for compound objects may bring several benefits. In order to advocate this claim we introduce a framework consisting of several overlapping categories (non-disjoint layers) of domain knowledge. This construct, as the tutorial aims to demonstrate, might be utilized for optimizing storage and processing of complex objects.

Proposed layers of knowledge:

Layer 1. Knowledge about the underlying, general problem to be solved with use of available resources and the collection of compound objects we have

gathered. This kind of knowledge includes also such elements as: optimization preferences (e.g., storage or computation time), number of end–users of the system, availability of data, etc.

Layer 2. Knowledge about objects, their internal structure, importance of attributes, orders on them, their types (including knowledge about measurement errors), relations between objects as well as the knowledge about computational methods used to process them. This type of knowledge includes also knowledge of probable computation scenarios: typical queries and processing methods, potential high-level bottlenecks, most frequent elementary data operations.

Layer 3. Knowledge about the technologies, models, and data schemes that are used to store the information about objects within database. One can utilize high level knowledge – of general assets and shortcomings of particular technologies as well as some low level aspects of knowledge specific to chosen technology, e.g., about physical representation of objects inside database, such as Infobright's column-wise data packages.

It shall be emphasized that, while designing data-based process the optimization steps, one needs to take into consideration all the levels mentioned above. These levels are inter-connected and only by considering all of them we may achieve significant improvements in terms of the speed (computational cost) and accuracy of algorithms.

2.3 Examples

The general ideas are illustrated with examples taken from large, real-life KDD&DM projects. This includes the following three applications:

Object Comparators in Identification of Contour Maps
This example shows practical implementation of comparator theory and its application in the commercial project aimed at visualization of the results of the 2010 Polish local elections and 2011 Polish general elections (see [2]). It demonstrates a path leading from identification of domain knowledge through its skillful use leading to optimization of the implemented solution. Thanks to layer methodology for knowledge incorporation described in the previous section it was possible to significantly improve the overall performance of the system. The main cost of this solution is associated with search and comparison of complex objects. Through knowledge-driven optimization of the reference set it was possible to reduce the computational effort and perform some steps concurrently, which led to further speed-up.

Knowledge Driven Query Sharding
The SYNAT (`www.synat.pl`) project is a large, national R&D program of Polish government. Within its framework our research group designs and implements a solution allowing the user to search within repositories of scientific information

using their semantic content. This sub-system – called SONCA[1] (cf. [5]) – is also meant to be a platform allowing search for new solutions in the field of semantic measures. During the research, we tried to develop a new measure of semantic similarity between documents used to group them into semantically coherent clusters. For that purpose we had to incorporate a lot of domain knowledge in the form of external ontologies and knowledge bases (e.g., MeSH [2]). We also had to perform several optimization steps in query processing that led to use of domain knowledge in design of algorithms (methods) for data processing and query answering in data warehouses.

Data Cleansing of Fire and Rescue Reporting System
In this example we describe how the domain knowledge makes it possible to perform data sampling and data cleansing. This example is associated with recently started major R&D project aiming at creation of tools for decision support for commanders of the State Fire Service of Poland during Fire&Rescue (F&R) operations in buildings [6–8].

After every F&R action a report is created in EWID – the reporting system of State Fire Service of Poland (PSP). The system currently contains nearly 6 million reports and around 1500 new entries are created every day. The concern is that over the years this large corpus has been collected with limited validation of the input. In the example we show a mostly automatic, iterative process of data cleansing and interpretation, supervised by domain experts and incorporating the domain (expert) knowledge.

3 Prerequisites and Target Audience

As the time allotted for the presentation of the tutorial is relatively short (90 min.), it is assumed that the audience is familiar with some basics. In order to present the more advanced (and entertaining) topics it is necessary to assume that the members of the audience have some experience in:

- Knowledge Discovery in Databases (KDD) at least at the level of understanding basic concepts and processes.
- Project management, in particular with DM projects. Ideally, the members of the audience have participated in such project, but the general idea will do as well.
- Underlying technologies, such as database management, data warehousing, data quality assurance and so on.

The person with completely no background in information systems and computational intelligence will probably feel lost.

[1] Abbreviated: **S**earch based on **ON**tologies and **C**ompound **A**nalytics.
[2] Medical Subject Headings `www.nlm.nih.gov/pubs/factsheets/mesh.html`

4 Conclusion

Our approach to utilization of domain (background) knowledge in the initial stages of KDD process, as presented during the tutorial, is not an answer to each and every problem. The area of KDD is too diversified and complicated for any methodology to always work equally well. Our approach to selection, preprocessing and transformation steps in KDD is an attempt to identify characteristic features that may be used to select the right, knowledge-based tool for the task at hand. Sometimes the results of these attempts may be difficult to ascertain, as we only operate on initial steps of KDD. It may be hard, if not impossible to know in advance if all operations performed on initial stages of KDD will bring significant improvement after data mining and result interpretation is concluded. It is, after all, a discovery process, and we usually don't know what exactly we shall expect to be discovered.

The series of examples presented in the paper illustrates both the variety of issues that have to be addressed and the apparent existence of the overall scheme behind. It supports the claim that, by properly identifying the level of complication of the task and the kind of domain knowledge we posses, one can achieve significant improvements in efficiency and quality of the solution. It has to be stated that the knowledge-based methods shown in application examples are demonstrating improvements on very diversified scale.

References

1. Sosnowski, Ł., Ślęzak, D.: RDBMS framework for contour identication. In: [9], pp. 487–498
2. Sosnowski, Ł., Ślęzak, D.: Comparators for compound object identification. In: Kuznetsov, S.O., Ślęzak, D., Hepting, D.H., Mirkin, B.G. (eds.) RSFDGrC 2011. LNCS, vol. 6743, pp. 342–349. Springer, Heidelberg (2011)
3. Krasuski, A., Szczuka, M.: Knowledge driven query sharding. In: Popova-Zeugmann, L. (ed.) CS&P 2012. Informatik Berichte, vol. 225, pp. 182–190. Humboldt Univeristät zu Berlin, Berlin (2012)
4. Szczuka, M., Sosnowski, Ł., Krasuski, A., Kreński, K.: Using domain knowledge in initial stages of KDD: Optimization of compound object processing. Fundamenta Informaticae TBA (to be published, 2013)
5. Nguyen, H.S., Ślęzak, D., Skowron, A., Bazan, J.G.: Semantic search and analytics over large repository of scientific articles. In: Bembenik, R., Skonieczny, L., Rybiński, H., Niezgodka, M. (eds.) Intelligent Tools for Building a Scient. Info. Plat. SCI, vol. 390, pp. 1–8. Springer, Heidelberg (2012)
6. 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)
7. Krasuski, A., Ślęzak, D., Kreński, K., Łazowy, S.: Granular knowledge discovery framework: A case study of incident data reporting system. In: Pechenizkiy, M., Wojciechowski, M. (eds.) New Trends in Databases & Inform. Sys. AISC, vol. 185, pp. 109–118. Springer, Heidelberg (2012)
8. Kreński, K., Krasuski, A., Łazowy, S.: Data mining and shallow text analysis for the data of State Fire Service. In: [9], pp. 313–321
9. Szczuka, M., Czaja, L., Skowron, A., Kacprzak, M. (eds.): Proceedings of the International Workshop on CS&P 2011, Pułtusk, Poland. Białystok University of Technology (2011)

Non-deterministic Information in Rough Sets: A Survey and Perspective

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 been coping with issues connected with *non-deterministic information* in rough sets. Non-deterministic information is a kind of incomplete information, and it defines a set in which the actual value exists, but we do not know which is the actual value. If the defined set is equal to the domain of attribute values, we may see this is corresponding to a *missing value*. We need to pick up the merits in each information, and need to apply them to analyzing data sets. In this paper, we describe our opinion on non-deterministic information as well as incomplete information, some algorithms, software tools, and its perspective in rough sets.

Keywords: Rough sets, Non-deterministic information, Incomplete information, Survey, Information dilution, Privacy-preserving.

1 Introduction

In our previous research, we coped with rule generation in *Non-deterministic Information Systems* (*NISs*) [12,17,19]. In contrast to *Deterministic Information Systems* (*DISs*) [16,20], *NISs* were proposed by Pawlak [16], Orłowska [13] and Lipski [9] in order to better handle information incompleteness in data. In *NISs*, we have defined *certain* and *possible* rules, and recently we proved an algorithm named *NIS-Apriori* is *sound* and *complete* for defined rules. We have also implemented *NIS-Apriori* [18] and a web software *getRNIA* [24]. This paper describes the role of non-deterministic information and its survey according to Figure 1. Since Figure 1 consists of five decades, we sequentially survey important work in each decade.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 7–15, 2013.

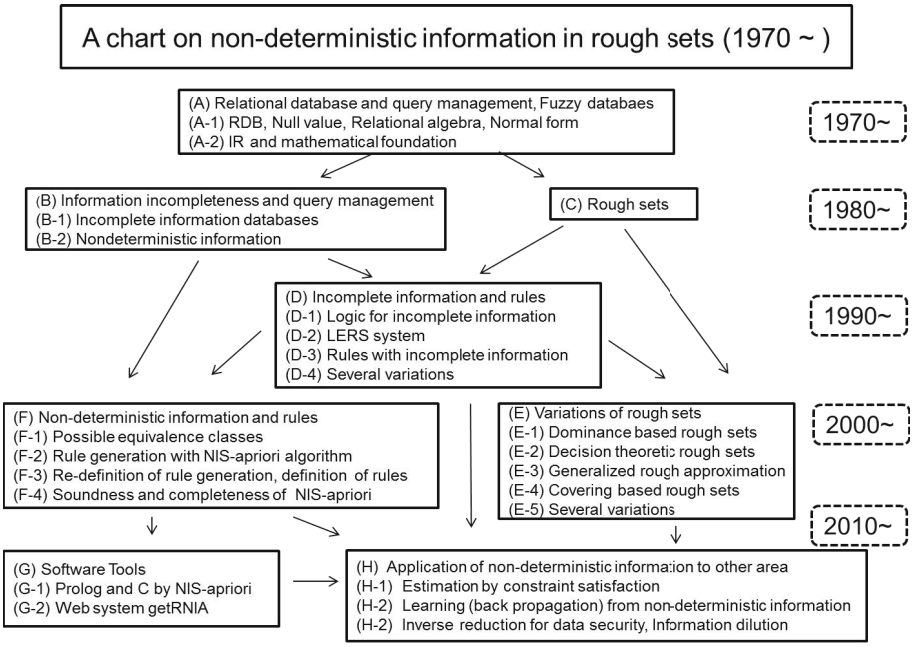


Fig. 1. A chart on non-deterministic information in rough sets

2 In 1970's: Relational Databases and Query Management

In 1970's, relational algebra, normal forms, null values are investigated [3], and Marek and Pawlak clarified mathematical foundations of information retrieval [11]. We think that each research coped with query management, and it supports current development of relational databases.

3 In 1980's: Information Incompleteness, Query Management and Rough Sets

In 1980's, information incompleteness in databases was investigated by Lipski [9,10]. Table 1 is an example of Lipski's incomplete information database cited from [9]. For *Age* whose domain is $(0, \infty)$, information about two persons x_3 and x_5 is definite. Information on three persons x_1, x_2 and x_4 is indefinite. For each of these cases, information is given as an interval. For *Dept#*, each attribute value is not an interval but a subset of a set of all department numbers. In Lipski's framework, we see the concept of non-deterministic information.

Table 2 is an example of nondeterministic information system cited from [13], where each attribute value is given as a set of possible values. We see the keyword *nondeterministic information* [13] by Orłowska and Pawlak, and *many-valued*

Table 1. An example of Lipski's Incomplete Information Database [9]. The age of x_2 is either 52, 53, 54, 55 or 56 years old, which is non-deterministic information.

OB	Age	$Dept\#$	$Hireyear$	Sal
x_1	$[60, 70]$	$\{1, \cdots, 5\}$	$\{70, \cdots, 75\}$	$\{10000\}$
x_2	$[52, 56]$	$\{2\}$	$\{72, \cdots, 76\}$	$(0, 20000]$
x_3	$\{30\}$	$\{3\}$	$\{70, 71\}$	$(0, \infty)$
x_4	$(0, \infty)$	$\{2, 3\}$	$\{70, \cdots, 74\}$	$\{22000\}$
x_5	$\{32\}$	$\{4\}$	$\{75\}$	$(0, \infty)$

information [15] by Pawlak. We think that query management and mathematical logic in databases with incomplete information were investigated in each research.

Table 2. A nondeterministic information system [13]. Each attribute value is given as a set of possible values.

OB	a_1	a_2
D_1	$\{v_1, v_3\}$	$\{u_1, u_2, u_3\}$
D_2	$\{v_2, v_5\}$	$\{u_1\}$
D_3	$\{v_1, v_3, v_4\}$	$\{u_1, u_2\}$
D_4	$\{v_1\}$	$\{u_1, u_2\}$
D_5	$\{v_1, v_3\}$	$\{u_1\}$
D_6	$\{v_5\}$	$\{u_1\}$

Rough sets were proposed by Pawlak in [14], and we think that the topic was moving from query management to knowledge discovery. Clearly, one of the roles of rough sets is to define a framework of rule generation in databases. In query management, the interpretation of a query (the conjunction of descriptors) was investigated, and the equivalence classes defined by the conjunction of descriptors have been employed in rough sets.

4 In 1990's: Information Incompleteness and Rule Generation

From 1990's, rule generation has been very important topic, and the relation between information incompleteness and rule generation has been investigated. In [4], theoretical aspect including logic and the complexity on incomplete information is described.

We also see other research, namely research for implementing real application systems. In [6], $LERS$ system was implemented by Grzymała-Busse, and this system is applied to several area. We understand the overview of $LERS$ system

as follows: At first, variations of the equivalence classes for handling missing values are defined as *blocks*. Each block is connected with a conjunction of descriptors $\wedge_i[A_i, val_i]$, respectively. For a target set T defined by the decision attribute value (this will be defined by a descriptor $[Dec, val]$), $LERS$ tries to cover T by using blocks. For block B_1, if $B_1 \subseteq T$, we obtain a *certain* rule $\wedge_i[A_i, val_i] \Rightarrow [Dec, val]$. For block B_2, if block $B_2 \not\subseteq T$ and $B_2 \cap T \neq \emptyset$, we obtain a *possible* rule $\wedge_i[A_i, val_i] \Rightarrow [Dec, val]$.

In [8], rules are defined according to incomplete information systems by Kryszkiewicz. In these two researches, missing values ($*$ in Table 3) are employed instead of non-deterministic information. According to the missing values, extended equivalence classes are defined, and the consistency of a rule is examined by the variations of inclusion relation $[x]_{CON} \subseteq [x]_{DEC}$. Table 3 is an incomplete information system, and Table 4 is a possible case of information system cited from [8].

Table 3. An example of incomplete information system Φ [8]

OB	a	b	c	d
1	1	1	1	1
2	1	$*$	1	1
3	2	1	1	1
4	1	2	$*$	1
5	1	$*$	1	2
6	2	2	2	2
7	1	1	1	2

Table 4. A possible information system from Φ [8]

OB	a	b	c	d
1	1	1	1	1
2	1	1	1	1
3	2	1	1	1
4	1	2	1	1
5	1	1	1	2
6	2	2	2	2
7	1	1	1	2

5 In 2000's: Variations of Rough Sets and Rough Non-deterministic Information Analysis

In 2000's or much earlier, we have several variations of rough sets, for example *dominance-based rough sets* [5], *decision-theoretic rough sets* [25], *generalized rough approximation* [2], *covering-based rough sets* [26], etc.

We understand the overview of dominance-based rough sets as follows: Each domain of attribute values has a total order, and rules are implications which are preserving the order of condition attribute values and the decision attribute values. The criterion 'consistency' in [16] is replaced with 'order-preserving', and the new framework is proposed.

We understand the overview of decision-theoretic rough sets as follows: This is the combined framework of rough sets and decision theory. Therefore, some possible choices with probability are given as well as a table, and the total expected value is employed for deciding a choice. This will be an extension from the typical decision theory and game theory.

We understand the overview of generalized rough approximation as follows: This work connects fuzzy sets with rough sets, and proposes the combination, namely the notion of fuzzy rough set. We think the combined framework will be more robust and general framework for approximation theory.

We understand the overview of covering-based rough sets as follows: This is also the combined framework of rough sets and algebraic structure, and this framework tries to include several types of data, i.e., Boolean-valued data, numeric and mixed data.

We described our opinion to four variational works on rough sets. We think that these four works cope with the theoretical aspects.

As for the implementation and real application, we see *Infobright* technology [21] based on rough sets. Even though this technology handles deterministic information, we are considering non-deterministic information on *Infobright* [22].

We also started the research named *Rough Non-deterministic Information Analysis (RNIA)*. Table 5 is an example of a $NIS\ \Psi$ [19].

Table 5. An exemplary $NIS\ \Psi$ for the suitcase data sets. Here, VAL_{Color}=$\{red, blue, green\}$, VAL_{Size}=$\{small, medium, large\}$, VAL_{Weight}=$\{light, heavy\}$, VAL_{Price}=$\{high, low\}$.

$Object$	$Color$	$Size$	$Weight$	$Price$
x_1	{red,blue,green}	{small}	{light,heavy}	{low}
x_2	{red}	{small,medium}	{light,heavy}	{high}
x_3	{red,blue}	{small,medium}	{light}	{high}
x_4	{red}	{medium}	{heavy}	{low,high}
x_5	{red}	{small,medium,large}	{heavy}	{high}
x_6	{blue,green}	{large}	{heavy}	{low,high}

In Table 5, $g(x_1, Color)$=$\{red, blue, green\}$ is equal to VAL_{Color}, therefore we may see $g(x_1, Color)$ is a missing value. However, $g(x_3, Color)$=$\{red, blue\}$ is different from VAL_{Color}. It is impossible to express $g(x_3, Color)$ by using a missing value. If Tom is a typical student in a graduate school, we will figure his age will be 22, 23, 24 or 25 years old. We do not figure his age will be 10's nor 50's. Intuitively, non-deterministic information may be seen as a missing value with the restricted domain.

We follow the way like Table 3 and Table 4, and we named a possible table a *derived DIS from a NIS* Ψ, and let $DD(\Psi)$ denote a set of all derived DIS from Ψ. In a NIS, we need to pay attention to the number of derived $DISs$. For example, there are 2304 ($=2^8 \times 3^2$) derived $DISs$ even in Table 5. The number of derived $DISs$ increases exponentially, therefore it will be hard to enumerate each derived DIS sequentially. In Hepatitis and Mammographic data sets in UCI machine learning repository [23], the number of derived $DISs$ are more than 10 power 90 [18].

We defined two rules, and proposed *NIS-Apriori* algorithm for handling rules in the following: [17,19].

(Certain rule τ) $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ hold in each $\phi \in DD(\Psi)$.
(Possible rule τ) $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ hold in some $\phi \in DD(\Psi)$.

The definition of two rules depends upon $|DD(\Psi)|$, however the computational complexity of *NIS-Apriori* does not depend upon $|DD(\Psi)|$, and its complexity is almost the same as the original *Apriori* algorithm [1].

6 In 2010's: Real Application, Perspective of Non-deterministic Information

Recently, we have implemented a software *getRNIA* [24] in Figure 2 and 3. This *getRNIA* employs *NIS-Apriori* for rule generation and granules for association

Fig. 2. An overview of the getRNIA

rules as data structure. Since this software is open, anyone can access to this site, and apply it to analyzing data sets.

Certain and possible Rule Generation(Reducted): back

Support: 0.100000
Accuracy: 0.300000

CON,p -> DEC,q

	CON,p -> DEC,q	minsupp	minacc	maxsupp	maxacc	Lower/Upper
1	temperature:normal->flu:yes	0	0	0.375	0.75	Upper
2	temperature:high->flu:yes	0.125	0.25	0.625	0.833	Upper
3	temperature:very_high->flu:yes	0	0	0.375	1	Upper
4	headache:yes->flu:yes	0.25	0.4	0.625	1	Lower
5	headache:no->flu:yes	0	0	0.25	0.667	Upper
6	nausea:yes->flu:yes	0.125	0.2	0.5	0.667	Upper
7	nausea:no->flu:yes	0.125	0.25	0.625	1	Upper
8	temperature:normal->flu:no	0.125	0.25	0.375	1	Upper
9	temperature:high->flu:no	0.125	0.167	0.375	0.75	Upper
10	temperature:very_high->flu:no	0	0	0.125	1	Upper
11	headache:yes->flu:no	0	0	0.375	0.6	Upper
12	headache:no->flu:no	0.125	0.333	0.375	1	Lower
13	nausea:yes->flu:no	0.25	0.333	0.5	0.8	Lower
14	nausea:no->flu:no	0	0	0.375	0.75	Upper

Fig. 3. An example of the execution of getRNIA

Now, we consider the role of non-deterministic information as well as incomplete information. In most of research on non-deterministic information and incomplete information, we usually suppose a data set with information incompleteness is given, and we coped with what conclusions are obtainable.

We think that the inverse of this research may be new topic, namely we intentionally add noisy attribute values to a table with keeping some rough set-based constraints. Let us consider Table 6, which is a simple table, and the degree of data dependency $age \Rightarrow sex$ is 1.0. We added some noise to Table 6, and we generated Table 7. In Table 7, information of age is changed. The original

Table 6. An example of deterministic information system ψ

OB	age	sex
Tom	25	$male$
$Mary$	24	$female$

Table 7. A revised non-deterministic information system from Ψ

OB	age	sex
Tom	$\{25, 26, 27\}$	$\{male\}$
$Mary$	$\{23, 24\}$	$\{female\}$

information is diluted, however the data dependency is still 1.0 in each derived DIS.

By diluting each information, the original information is hidden. Such information dilution may be applicable to keep the security of original data sets. In data mining, we usually do not open the original data sets for keeping the privacy-preserving. However, if we can dilute a DIS ψ to a NIS Ψ with keeping the obtainable rules, we may open a data set Ψ. Because, some important attribute values can be diluted for keeping the privacy-preserving. The actual example of information dilution will be in the proceedings of JRS2013.

In rough sets, we have several work on reduction with keeping some constraints. Similarly to reduction, we will have several work on information dilution with keeping some constraints. We may also say this *inverse-reduction*.

7 Concluding Remarks

We have briefly surveyed non-deterministic information and incomplete information in rough sets. We have investigated what is concluded according to a given table with information incompleteness, like query management and rule generation. However, the inverse of the previous research, namely we intentionally dilute a DIS ψ to a NIS Ψ related to privacy-preserving, may be new topic on rough sets. The inverse-reduction may also be a new topic.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. of VLDB, pp. 487–499 (1994)
2. Ciucci, D., Flaminio, T.: Generalized rough approximations in PI 1/2. Int. Journal of Approximate Reasoning 48(2), 544–558 (2008)
3. Codd, E.: A relational model of data for large shared data banks. Communication of the ACM 13, 377–387 (1970)
4. Demri, S., Orłowska, E.: Incomplete Information: Structure, Inference, Complexity. Monographs in Theoretical Computer Science. Springer (2002)
5. Greco, S., Matarazzo, B., Słowiński, R.: Granular computing and data mining for ordered data: The dominance-based rough set approach. In: Encyclopedia of Complexity and Systems Science, pp. 4283–4305 (2009)
6. Grzymała-Busse, J.: A new version of the rule induction system LERS. Fundamenta Informaticae 31, 27–39 (1997)
7. Grzymała-Busse, J., Rząsa, W.: A local version of the MLEM2 algorithm for rule induction. Fundamenta Informaticae 100, 99–116 (2010)
8. Kryszkiewicz, M.: Rules in incomplete information systems. Information Sciences 113, 271–292 (1999)
9. Lipski, W.: On semantic issues connected with incomplete information data base. ACM Trans. DBS. 4, 269–296 (1979)
10. Lipski, W.: On databases with incomplete information. Journal of the ACM 28, 41–70 (1981)
11. Marek, W., Pawlak, Z.: Information storage and retrieval systems: Mathematical foundations. Theoretical Computer Science 1(4), 331–354 (1976)

12. Nakata, M., Sakai, H.: Twofold rough approximations under incomplete information. International Journal of General Systems 42(6), 546–571 (2013)
13. Orłowska, E., Pawlak, Z.: Representation of nondeterministic information. Theoretical Computer Science 29, 27–39 (1984)
14. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
15. Pawlak, Z.: Systettly Informacyjne. Podstawy Teoretyczne (Information Systems. Theoretical Foundations). WNT, Warsaw (1983)
16. Pawlak, Z.: Rough Sets. Kluwer Academic Publishers (1991)
17. Sakai, H., Ishibashi, R., Nakata, M.: On rules and apriori algorithm in non-deterministic information systems. Transactions on Rough Sets 9, 328–350 (2008)
18. Sakai, H.: RNIA software logs (2011), http://www.mns.kyutech.ac.jp/~sakai/RNIA
19. Sakai, H., Okuma, H., Nakata, M.: Rough non-deterministic information analysis: Foundations and its perspective in machine learning. In: Ramanna, S., Jain, L.C., Howlett, R.J. (eds.) Emerging Paradigms in ML. SIST, vol. 13, pp. 215–247. Springer, Heidelberg (2013)
20. 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)
21. Ślęzak, D., Eastwood, V.: Data warehouse technology by infobright, Proc. In: SIGMOD Conference, pp. 841–846 (2009)
22. Ślęzak, D., Sakai, H.: Automatic extraction of decision rules from non-deterministic data systems: Theoretical foundations and SQL-based implementation. In: Ślęzak, D., Kim, T.-H., Zhang, Y., Ma, J., Chung, K.-I. (eds.) DTA 2009. CCIS, vol. 64, pp. 151–162. Springer, Heidelberg (2009)
23. UCI Machine Learning Repository, http://mlearn.ics.uci.edu/MLRepository.html
24. Wu, M., Sakai, H.: getRNIA web software (2013), http://getrnia.appspot.com/
25. Yao, Y., Zhao, Y.: Attribute reduction in decision-theoretic rough set models. Information Sciences 178(17), 3356–3373 (2008)
26. Zhu, W.: Topological approaches to covering rough sets. Information Sciences 177(6), 1499–1508 (2007)

Granular Computing and Sequential Three-Way Decisions

Yiyu Yao*

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
yyao@cs.uregina.ca

Abstract. Real-world decision making typically involves the three options of acceptance, rejection and non-commitment. Three-way decisions can be motivated, interpreted and implemented based on the notion of information granularity. With coarse-grained granules, it may only be possible to make a definite decision of acceptance or rejection for some objects. A lack of detailed information may make a definite decision impossible for some other objects, and hence the third non-commitment option is used. Objects with a non-commitment decision may be further investigated by using fine-grained granules. In this way, multiple levels of granularity lead naturally to sequential three-way decisions.

1 Introduction

Two fundamental notions of rough set theory are knowledge granularity [15, 16] and the approximation of a concept by a pair of lower and upper approximations [3, 4] or three regions. In this paper, I argue that the two notions play an equally important role in a theory of three-way decisions [29]. Three-way decisions can be motivated, interpreted and implemented based on the notion of information and knowledge granularity. Three regions of rough sets [15], and in particular probabilistic rough sets [3, 4, 24, 25], lead naturally to three-way decisions [27, 28], which may produce better results in rule learning [5]. A theory of three-way decisions may be viewed an extension of rough set theory, based on the same philosophy but goes beyond. Three-way decisions focus on a more general class of problems where a set of objects are divided into three pair-wise disjoint regions [2, 29].

A two-way decision consists of either an acceptance or a rejection of an object for a specific purpose. However, a two-way decision may not always be possible in real life in the context of multiple levels of granularity and multiple levels of approximations. At a higher level of granularity, one may have a more abstract and compact representation of a decision problem by omitting details, leading to a faster decision process but a less accurate result. On the other hand, at a lower level of granularity, one may have a more concrete and elaborate representation, leading to a slower decision process but a more accurate result. Therefore,

* This work is partially supported by a discovery grant from NSERC Canada.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 16–27, 2013.

making the right decision at the right level is a crucial issue. Three-way decisions, consisting of acceptance, rejection, and non-commitment, are a practical solution. When the available information is insufficient or the evidence is not strong enough to support an acceptance or a rejection at a particular level of granularity, a third option of non-commitment allows us to defer a decision to the next level of granularity.

Three-way decisions may be related to a basic principle of granular computing. By utilizing granular structures, granular computing [1, 17, 21, 22, 31] focuses on a set of philosophy, methodology and paradigm for structured thinking, structured problem solving and structured information processing at multiple levels of granularity [26]. Granular structures consist of many hierarchies for multiview descriptions of a problem, with each hierarchy being composed of multiple levels of abstraction [26]. In an earlier paper [23], I stated that a basic principle of computing, guided by granular structures, is to

> *"··· examine the problem at a finer granulation level with more detailed information when there is a need or benefit for doing so."*

The objective of the present study is to introduce sequential three-way decisions based on this principle. We want to make a decision "at a finer granulation level with more detailed information when there is a need or benefit for doing so."

There are two contributions from the study. One is to provide a granular computing perspective on three-way decisions. I will demonstrate that three-way decisions are superior and necessary in the context of multiple levels of information granularity. That is, a decision problem is more appropriately formulated as a sequence of three-way decisions, leading finally to two-way decisions. The other is a demonstration of a basic principle of granular computing and, hence, makes it easily understandable and applicable to a wide range of applications.

2 An Overview of Three-Way Decisions

By extending the three-way classification of rough set theory and synthesizing results across many disciplines, I examined a theory of three-way decisions in an earlier paper [29]. The main results are briefly reviewed in this section.

Suppose U is a finite nonempty set of objects and $\mathbf{C}$ is a finite set of conditions. Depending on applications, a condition in $\mathbf{C}$ may be a criterion, an objective, or a constraint. A decision task is to divide U into regions according to the satisfiability of objects of the set of conditions $\mathbf{C}$. Formally, **the problem of three-way decisions** can be stated as follows:

> The problem of three-way decisions is to divide U, based on the set of conditions $\mathbf{C}$, into three pair-wise disjoint regions, POS, NEG, and BND, called the positive, negative, and boundary regions, respectively. The positive region POS consists of those objects that we *accept* as satisfying the conditions and the negative region NEG consists of those objects that we *reject* as satisfying the conditions. For objects in the boundary region BND, we neither accept nor reject, corresponding to a non-commitment.

The satisfiability reflects a nature of the objects. It may be either qualitative or quantitative; it may also be known, partially known, or unknown. For an object $x \in U$, let $s(x)$ denote the satisfiability of x of the set of conditions $\mathbf{C}$ and is called the state of x. Depending on the set of all possible values of $s(\cdot)$, we may have two-state and many-state decisions problems.

For the two-state case, if we know the true state $s(x)$ for every object, we do not really need three-way decisions, as we can simply classify objects into two regions based on $s(x)$. In many situations, we may not know the true state of an object and may only construct a function $v(x)$ to help us in probing the true state $s(x)$. The value $v(x)$ is called the decision status value of x and may be interpreted as the probability or possibility that x satisfies $\mathbf{C}$. In this context, three-way decisions seem to be appropriate. For the many-state case, even if we know $s(x)$, a three-way decision is still necessary. The results of three-way decisions may be viewed as a three-valued approximation.

In the rest of this paper, I only consider a two-state three-way decisions model that uses an evaluation $v : U \longrightarrow L$ to estimate the states of objects in U, where $(L, \preceq)$ is a totally ordered set. By introducing a pair of thresholds (α, β), $\beta \prec \alpha$ (i.e., $\beta \preceq \alpha$ and $\beta \neq \alpha$), on the evaluation v, we construct three regions as follows:

$$\begin{aligned}
\mathrm{POS}_{(\alpha,\beta)}(v) &= \{x \in U \mid v(x) \succeq \alpha\},\\
\mathrm{NEG}_{(\alpha,\beta)}(v) &= \{x \in U \mid v(x) \preceq \beta\},\\
\mathrm{BND}_{(\alpha,\beta)}(v) &= \{x \in U \mid \beta \prec v(x) \prec \alpha\},
\end{aligned} \tag{1}$$

where for $a, b \in L$, $a \succeq b \Longleftrightarrow b \preceq a$ and $a \prec b \Longleftrightarrow (a \preceq b, a \neq b)$. Condition $\beta \prec \alpha$ implies that the three regions are pair-wise disjoint. Since some of the regions may be empty, the three regions do not necessarily form a partition of the universe U.

From the formulation, we must consider at least the following issues:

- Construction and interpretation of the totally ordered set $(L, \preceq)$.
- Construction and interpretation of the evaluation $v(\cdot)$.
- Construction and interpretation of the pair of thresholds (α, β).

The value $v(x)$ may be interpreted as the probability, possibility or degree to which x satisfies $\mathbf{C}$. The pair of threshold (α, β) can be related to the cost or error of decisions. Those notions will be further discussed in the next section.

3 A Model of Sequential Three-Way Decisions

In this section, I propose a sequential three-way decision model for two-state decision model and show its advantages over two-way decisions.

3.1 Simple Two-Way Decisions

For a two-state decision problem, we assume that each object $x \in U$ is in one of the two states: either satisfies the set of conditions $\mathbf{C}$ or does not. The state of

an object is an inherent property of the object, independent of whether we have sufficient information to determine it. Let a mapping $s : U \longrightarrow \{0, 1\}$ denote the states of all objects as follows:

$$s(x) = \begin{cases} 1, & x \text{ satisfies } \mathbf{C}, \\ 0, & x \text{ does not satisfy } \mathbf{C}. \end{cases} \tag{2}$$

We make a decision regarding the true state of an object based on a representation, a description of, or some information about x. In many situations, the available information may be incomplete and uncertain, and the set of conditions may not be formally and precisely stated. It is impossible to determine the state of each object with certainty. We can construct an evaluation function to assist in a decision-making process.

Let $\text{Des}(x)$ denote a description of x and U_D denote the set of all possible descriptions. An evaluation, $v : U_D \longrightarrow L$, is now given by a mapping from U_D to a totally ordered set $(L, \preceq)$. The quantity $v(\text{Des}(x))$ is called the decision status value of x. Intuitively, a larger value $v(\text{Des}(x))$ suggests that the object x satisfies the conditions $\mathbf{C}$ to a higher degree. Based on the decision status values and a threshold $\gamma \in L$, we can divide U into a positive region and a negative region based on a strategy of two-way decisions:

$$\begin{aligned} \text{POS}_\gamma(v) &= \{x \in U \mid v(\text{Des}(x)) \succeq \gamma\}, \\ \text{NEG}_\gamma(v) &= \{x \in U \mid v(\text{Des}(x)) \prec \gamma\}. \end{aligned} \tag{3}$$

The positive region consists of those objects that we *accept* as satisfying the conditions in $\mathbf{C}$ and negative region consists of those objects that we *reject* as satisfying the conditions in $\mathbf{C}$.

3.2 Sequential Three-Way Decisions

In the simple two-way decisions, we use a single representation of an object. In real-world decision making, we may consider a sequence of three-way decisions that eventually leads to two-way decisions. At each stage, new and more information is acquired. For example, in clinical decision making, based on available information, a doctor may decide to treat or not to treat some patients; for some other patients, the doctor may prescribe further tests and defer a decision to the next stage [14]. The basic ideas of sequential three-way decisions appear in a model of sequential three-way hypothesis testing introduced by Wald [20] and a model of sequential three-way decisions with probabilistic rough sets [30]. Li et al. [7] consider a sequential strategy for making cost-sensitive three-way decisions. Sosnowski and Ślęzak [18] introduce a model of networks of comparators for solving problems of object identification, in which a sequence of comparators is used for decision-making. In this paper, I present another way to formally formulate sequential three-way decisions through the notion of multiple levels of granularity. The main components of the proposed model are discussed below.

Multiple Levels of Granularity. We assume that there are $n + 1$, $n \geq 1$, levels of granularity. For simplicity, we use the index set $\{0, 1, 2, \ldots, n\}$ to denote

the $n+1$ levels, with 0 representing the finest granularity (i.e., the ground level) and n the coarsest granularity. The simple two-way decisions can be viewed as decision-making at the ground level 0. For sequential three-way decisions, we assume that a three-way decision is made at levels $n, n-1, \ldots, 1$ and a two-way decision is made at the ground level 0. That is, the final result of sequential three-way decisions is a two-way decision. At each stage, only objects with a non-commitment decision will be further explored in the next level.

Multiple Descriptions of Objects. With $n+1$ levels, we have $n+1$ distinct representations and descriptions of the same object at different levels. Suppose

$$\text{Des}_0(x) \preceq \text{Des}_1(x) \preceq \ldots \preceq \text{Des}_n(x), \tag{4}$$

is a sequence of descriptions of object $x \in U$ with respect to $n+1$ levels of granularity. The relation $\preceq$ denotes a "finer than" relationship between different descriptions. A description at a coarser level is more abstract by removing some details of description in a finer level. It may be commented that the languages used to describe objects may be different at different levels. Consequently, the processing methods and costs may also be different.

Multiple Evaluations of Objects. Due to different representations at different levels, we need to consider different evaluations too. Let v_i, $0 \leq i \leq n$, denote an evaluation at level i whose values are from a totally ordered sets $(L_i, \preceq_i)$. In contrast to the strategy of simple two-way decision making, in a sequential three-way decision process the same object may be evaluated in several levels. Therefore, we must consider the extra costs of the decision process at different levels. The costs may include, for example, the cost needed for obtaining new information and the cost of computing the evaluation v_i.

Three-Way Decisions at a Particular Level. Except the ground level 0, we may make three-way decisions for objects with a non-commitment decision. Suppose U_{i+1} is the set of objects with a non-commitment decision from level $i+1$. For level n, we use the entire set U as the set of objects with a non-commitment decision, i.e., $U_{n+1} = U$. For level i, $1 \leq i \leq n$, we can choose a pair of thresholds $\alpha_i, \beta_i \in L_i$ with $\beta_i \prec_i \alpha_i$. Three-way decision making can be expressed as:

$$\begin{aligned}
\text{POS}_{(\alpha_i,\beta_i)}(v_i) &= \{x \in U_{i+1} \mid v_i(\text{Des}_i(x)) \succeq_i \alpha_i\}, \\
\text{NEG}_{(\alpha_i,\beta_i)}(v_i) &= \{x \in U_{i+1} \mid v_i(\text{Des}_i(x)) \preceq_i \beta_i\}, \\
\text{BND}_{(\alpha_i,\beta_i)}(v_i) &= \{x \in U_{i+1} \mid \beta_i \prec_i v_i(\text{Des}_i(x)) \prec_i \alpha_i\}.
\end{aligned} \tag{5}$$

The boundary region gives the set of objects with a non-commitment decision, namely, $U_i = \text{BND}_{(\alpha_i,\beta_i)}(v_i)$. For level 0, a two-way decision is made for the set of objects U_1 based on a single threshold $\gamma_0 \in L_0$.

Due to a lack of detailed information, one may prefer to a deferment decision to increase the chance of making a correct acceptance or rejection decision when more evidence and details are available at lower levels. This can be controlled by setting proper thresholds at different levels. Typically, one may use a larger threshold α and a smaller threshold β at a higher level of granularity [30].

Algorithm 1. S3D (Sequential three-way decisions)

Input: A set of objects U, a family of descriptions for each object $\{\text{Des}_i(x)\}$, a set of evaluations $\{v_i\}$, and a set of pairs of thresholds $\{(\alpha_i, \beta_i)\}$;
Output: Two regions POS and NEG;
begin
  $\text{POS} = \emptyset$;
  $\text{NEG} = \emptyset$;
  $i = n$;
  $U_{n+1} = U$;
  $U_1 = \emptyset$;
  while $U_{i+1} \neq \emptyset$ and $i > 0$ **do**
    $\text{POS}_{(\alpha_i,\beta_i)}(v_i) = \{x \in U_{i+1} \mid v_i(\text{Des}_i(x)) \succeq_i \alpha_i\}$;
    $\text{NEG}_{(\alpha_i,\beta_i)}(v_i) = \{x \in U_{i+1} \mid v_i(\text{Des}_i(x)) \preceq_i \beta_i\}$;
    $\text{BND}_{(\alpha_i,\beta_i)}(v_i) = \{x \in U_{i+1} \mid \beta_i \prec_i v_i(\text{Des}_i(x)) \prec_i \alpha_i\}$;
    $\text{POS} = \text{POS} \cup \text{POS}_{(\alpha_i,\beta_i)}(v_i)$;
    $\text{NEG} = \text{NEG} \cup \text{NEG}_{(\alpha_i,\beta_i)}(v_i)$;
    $U_i = \text{BND}_{(\alpha_i,\beta_i)}(v_i)$;
    $i = i - 1$;
  if $U_1 \neq \emptyset$ **then**
    $\text{POS}_{\gamma_0}(v_0) = \{x \in U \mid v_0(\text{Des}_0(x)) \succeq \gamma_0\}$;
    $\text{NEG}_{\gamma_0}(v_0) = \{x \in U \mid v_0(\text{Des}_0(x)) \prec \gamma_0\}$;
    $\text{POS} = \text{POS} \cup \text{POS}_{\gamma_0}(v_0)$;
    $\text{NEG} = \text{NEG} \cup \text{NEG}_{\gamma_0}(v_0)$;
  return POS, NEG;

Fig. 1. Algorithm of sequential three-way decisions

By summarizing the discussion, Figure 1 gives the algorithm S3D of sequential three-way decisions. In the algorithm, the set U_1 is initialized to the empty set. It will remind to be empty if an empty boundary region is obtained before reaching the ground level 0. In addition to the construction of the evaluation and thresholds at each level, for sequential three-way decisions, one must consider the construction and interpretation of a sequence of multiple levels of granularity.

4 Comparison of Simple Two-Way Decisions and Sequential Three-way Decisions

In this section, I provide an analysis of costs associated with two-way and sequential three-way decisions to demonstrate that there may be advantages to using a sequence of three-way decisions.

4.1 Total Cost of Decisions

Simple two-way decisions and sequential three-way decisions can be compared from two aspects. One is quality of the decision result in terms of errors or costs

caused by incorrect decisions and the other is cost of the decisions process for arriving at a decision. Both types of cost have been well studied and widely used in comparing different algorithms of two-way classification. In comparison, the latter has received less attention, except for the case of decision-tree based classification methods [10–13, 19]. When classifying an object with a decision tree, it is necessary to perform a sequence of tests of some internal nodes of the tree. The cost of the decision process can be viewed as the total cost of all required tests. The proposed sequential three-way decisions share some similarities with decision-tree based methods, but focus more on multiple levels of granularity and multiple representations of an objects. The cost of decision process becomes an important factor [6, 8, 9].

Suppose $COST_R$ and $COST_P$ denote, respectively, the cost of the decision result and the cost of the decision process. It is reasonable to assume that the total cost of decisions is a function for pooling together the two costs, that is,

$$COST = F(COST_R, COST_P). \tag{6}$$

There are many choices of the function F. Two special forms of the function are the simple linear combination and product:

$$\begin{aligned} COST' &= w_R * COST_R + w_P * COST_P, \\ COST'' &= (COST_R)^a * (COST_P)^b, \end{aligned} \tag{7}$$

where the weights $w_R \geq 0$, $w_P \geq 0$ and $w_R + w_P \neq 0$, and $a \geq 0$, $b \geq 0$ and $a + b \neq 0$, represent respectively the relative importance of the two types of costs. There seems to be an inverse relationship between the two types of costs. A decision-making method may produce a high quality result but tends to require a large processing cost. It may also happen that a decision-making method may require a small processing cost but produces a low quality result. In general, there is a trade-off between the two types of costs. Finding the right balance holds the key to making effective decisions.

4.2 Cost of the Decision Result

The result of simple two-way decisions and the final result of sequential three-way decisions are, respectively, a division of U into two regions POS and NEG. Some of the decisions of acceptance and rejection for constructing the two regions may, in fact, be incorrect. Let $S_1 = \{x \in U \mid s(x) = 1\}$ be the set of objects in state 1 and $S_0 = \{x \in U \mid s(x) = 0\}$ be the set of objects in state 0. Table 1 summarizes the errors and costs of various decisions, where $S = 1$ and $S = 0$ denote the two states of objects and $|\cdot|$ denotes the cardinality of a set.

The rates of two types of error, i.e., incorrect acceptance error (IAE) and incorrect rejection error (IRE), are given by:

$$\begin{aligned} IAE &= \frac{|\text{POS} \cap S_0|}{|\text{POS}|}, \\ IRE &= \frac{|\text{NEG} \cap S_1|}{|\text{NEG}|}, \end{aligned} \tag{8}$$

Table 1. Information of decision result

	$s(x) = 1$ (P)	$s(x) = 0$ (N)	total
a_A: accept	Correct acceptance	Incorrect acceptance	
	$\|\text{POS} \cap S_1\|$	$\|\text{POS} \cap S_0\|$	$\|\text{POS}\|$
a_R: reject	Incorrect rejection	Correct rejection	
	$\|\text{NEG} \cap S_1\|$	$\|\text{NEG} \cap S_0\|$	$\|\text{NEG}\|$
total	$\|S_1\|$	$\|S_0\|$	$\|U\|$

(a) Errors of decision result

	$s(x) = 1$ (P)	$s(x) = 0$ (N)
a_A: accept	$\lambda_{AP} = \lambda(a_A \| S = 1)$	$\lambda_{AN} = \lambda(a_A \| S = 0)$
a_R: reject	$\lambda_{RP} = \lambda(a_R \| S = 1)$	$\lambda_{RN} = \lambda(a_R \| S = 0)$

(b) Costs of decision result

where we assume that the positive and negative regions are nonempty, otherwise, the corresponding rate of error is defined as 0. Let $a(x)$ denote a decision made for object x. The total cost of decision results of all objects is computed as,

$$\begin{aligned} COST_R &= \sum_{x \in U} \lambda(a(x)|S = s(x)) \\ &= |\text{POS} \cap S_1| * \lambda(a_A|S = 1) + |\text{POS} \cap S_0| * \lambda(a_A|S = 0) + \\ &\quad |\text{NEG} \cap S_1| * \lambda(a_R|S = 1) + |\text{NEG} \cap S_0| * \lambda(a_R|S = 0) \\ &= |\text{POS}| * ((1 - IAE) * \lambda(a_A|S = 1) + IAE * \lambda(a_A|S = 0)) + \\ &\quad |\text{NEG}| * (IRE * \lambda(a_R|S = 1) + (1 - IRE) * \lambda(a_R|S = 0)). \end{aligned} \tag{9}$$

The total cost of decision result is related to the two types of decision error. The rates of errors and total cost may be used to design an objective function for finding an optimal threshold γ in simple two-way decisions.

Consider a special cost function defined by:

$$\begin{aligned} &\lambda_{AP} = 0, \qquad \lambda_{AN} = 1; \\ &\lambda_{RP} = 1, \qquad \lambda_{RN} = 0. \end{aligned} \tag{10}$$

There is a unit cost for an incorrect decision and zero cost for a correct decision. By inserting this cost function in to Equation (9), we have

$$\begin{aligned} COST_R &= |\text{POS}| * IAE + |\text{NEG}| * IRE \\ &= |\text{POS} \cap S_0| + |\text{NEG} \cap S_1|. \end{aligned} \tag{11}$$

The first expression suggests that the cost is a weighted sum of the two rates of incorrect decisions. The cost based measure is more informative than rates of incorrect decision, as the latter can be viewed as a special case of the former. The second expression suggests that the cost is the number of objects with an incorrect decision.

4.3 Costs of the Decision Process

For simple two-way decisions, we assume that all decisions are made at the ground level 0. The cost for processing each object is C_0 and the cost of the decision process is given by:

$$COST_{2P} = |U| * C_0. \tag{12}$$

When the cost C_0 is very large, the cost of the decision process $COST_{2P}$ may be very high. For many decision-making problem, we may not need to acquire all information of the ground level 0. This suggests a strategy of sequential decisions in which additional information is gradually acquired when it is necessary.

Let C_i denote the cost needed for evaluating an object at level i. It is reasonable to assume,

$$C_0 > C_i > 0, \quad i = n, n-1, \ldots, 1. \tag{13}$$

That is, the cost of the decision process at an abstract level is strictly less than at the ground levels; otherwise, we will not have any advantages of using the strategy of sequential three-way decision making. The magnitudes of C_i's depend on special applications. Consider a special case where C_i represents time needed for computing the evaluation at level i. We can assume that

$$C_n < C_{n-1} < \ldots < C_0. \tag{14}$$

This is equivalent to saying that we can make a faster decision at a higher level of granularity, as we do not have to consider minute details of the lower levels.

According to the condition $C_0 > C_i > 0, \quad i = n, n-1, \ldots, 1$, if we can make a definite decision of an acceptance or a rejection at higher levels of granularity, we may be able to avoid a higher cost at the ground level 0. Let $l(x)$ denote the level at which a decision of an acceptance or a rejection is made for x. The object x is considered in all levels from level n down to level $l(x)$. The processing cost of x can be computed as:

$$COST_{3P}(x) = \sum_{i=l(x)}^{n} C_i = C_{n \to l(x)}, \tag{15}$$

where $C_{n \to i}$ denote the cost incurred from level n down to level i. The total processing cost for all objects can be computed as follows:

$$\begin{aligned} COST_{3P} &= \sum_{x \in U} COST_{3P}(x) \\ &= \sum_{i=0}^{n} (|\mathrm{POS}_{(\alpha_i, \beta_i)}(v_i)| + |\mathrm{NEG}_{(\alpha_i, \beta_i)}(v_i)|) * C_{n \to i}. \end{aligned} \tag{16}$$

According to this equation, if the cost C_0 is very large and we can make an acceptance or a rejection decision for a majority of objects before reaching the ground level 0, the advantages of sequential three-way decisions will be more pronounced. On the other hand, if a definite decision of an acceptance or a rejection

is made for the majority of objects at lower levels of granularity, sequential three-way decisions would be inferior.

To gain more insights into sequential three-way decisions, let us consider a special composition of the cost C_i:

$$C_i = C_i^E + C_i^A, \tag{17}$$

where C_i^E denotes the cost for computing the evaluation v_i and C_i^A denotes the cost for acquiring additional information at level i. For this interpretation, we have the following assumption:

$$C_n^E \leq C_{n-1}^E \leq \ldots \leq C_0^E.$$

The assumption suggests that the cost for computing the evaluation function is lower at a higher level granularity due to the omission of detailed information. For simple two-way decisions at ground level 0, we must consider all information acquired from levels n down to 1. For an object x, the costs of decision processes of simple two-way decisions and sequential three-way decisions are given, respectively, by:

$$\begin{aligned} COST_{2P}(x) &= C_0^E + C_{n\to 0}^A, \\ COST_{3P}(x) &= C_{n\to l(x)}^E + C_{n\to l(x)}^A. \end{aligned} \tag{18}$$

It follows that

$$COST_{2P}(x) - COST_{3P}(x) = C_{(l(x)-1)\to 0}^A - (C_{n\to l(x)}^E - C_0^E). \tag{19}$$

The first term represents the extra cost of simple two-way decisions for acquiring extra information from level $l(x) - 1$ down to level 0, and the second term represents the extra cost of sequential three-way decisions in computing evaluations from level n down to level $l(x)$. That is, sequential three-way decisions reduce the cost of acquiring information at the expense of computing additional evaluations. If the difference in Equation (19) is greater than 0, then sequential three-way decisions have an advantage of a lower cost of the decision process. In situations where the cost of acquiring new information is more than the cost of computing evaluations, sequential three-way decisions are superior to simple two-way decisions at the ground level 0 with respect to the cost of decision process. In addition, when simple two-way decisions and sequential three-way decisions produce decision results of comparable quality, sequential three-way decisions are a better choice.

In general, we want to have sequential three-way decisions that produce the similar decision quality as simple two-way decisions but have a lower cost of decision process. To achieve this goal, one needs study carefully the cost structures of sequential three-way decisions in order to determine the best number of levels and best thresholds at each level. This implies that designing a sequential three-way decision procedure is more difficulty than designing a simple two-way decision procedure. There are many challenging problems to be solved for sequential three-way decisions.

5 Conclusion

In this paper, I present a granular computing perspective on sequential three-way decisions. Multiple levels of granularity lead to multiple representations of the same object, which in turn leads to sequential three-way decisions. Sequential decisions rely on a basic principle of granular computing, i.e., one only examines lower levels of granularity if there is a benefit. By considering the cost of the decision process, I show that a sequential three-way decision strategy may have a lower cost of the decision process than a simple two-way decision strategy, as the former may require less information and demand less time for computing evaluations at higher levels of granularity. Sequential three-way decisions are particularly useful for practical decision-making problems when information is unavailable and is acquired on demands with associated cost.

Sequential three-way decisions are much more complicated than simple two-way decisions. There are many challenging issues. One must construct multiple levels of granularity and multiple representations of the same object. One must consider more parameters, such as the number of levels, evaluations at different levels, and the thresholds at each level. One must also study cost structures that make sequential three-way decisions a better strategy.

References

1. Bargiela, A., Pedrycz, W. (eds.): Human-Centric Information Processing Through Granular Modelling. Springer, Berlin (2009)
2. Ciucci, D., Dubois, D., Prade, H.: Oppositions in rough set theory. 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. 504–513. Springer, Heidelberg (2012)
3. Clark, P.G., Grzymala-Busse, J.W., Rzasa, W.: Generalizations of approximations. In: Lingras, P., Wolski, M., Cornelis, C., Mitra, S., Wasilewski, P. (eds.) RSKT 2013. LNCS (LNAI), vol. 8171, pp. 41–52. Springer, Heidelberg (2013)
4. Grzymala-Busse, J.W.: Generalized probabilistic approximations. In: Peters, J.F., Skowron, A., Ramanna, S., Suraj, Z., Wang, X. (eds.) Transactions on Rough Sets XVI. LNCS, vol. 7736, pp. 1–16. Springer, Heidelberg (2013)
5. Grzymala-Busse, J.W., Yao, Y.Y.: Probabilistic rule induction with the LERS data mining system. International Journal of Intelligent Systems 26, 518–539 (2011)
6. Jia, X.Y., Liao, W.H., Tang, Z.M., Shang, L.: Minimum cost attribute reduction in decision-theoretic rough set models. Information Sciences 219, 151–167 (2013)
7. Li, H.X., Zhou, X.Z., Huang, B., Liu, D.: Cost-sensitive three-way decision: A sequential strategy. In: Lingras, P., Wolski, M., Cornelis, C., Mitra, S., Wasilewski, P. (eds.) RSKT 2013. LNCS (LNAI), vol. 8171, pp. 325–337. Springer, Heidelberg (2013)
8. Li, H.X., Zhou, X.Z., Zhao, J.B., Huang, B.: Cost-sensitive classification based on decision-theoretic rough set model. In: Li, T.R., Nguyen, H.S., Wang, G.Y., Grzymala-Busse, J.W., Janicki, R., Hassanien, A.E., Yu, H. (eds.) RSKT 2012. LNCS, vol. 7414, pp. 379–388. Springer, Heidelberg (2012)
9. Liu, D., Li, T.R., Liang, D.C.: Three-way government decision analysis with decision-theoretic rough sets. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 20, 119–132 (2012)

10. Min, F., He, H.P., Qian, Y.H., Zhu, W.: Test-cost-sensitive attribute reduction. Information Sciences 181, 4928–4942 (2011)
11. Min, F., Liu, Q.H.: A hierarchical model for test-cost-sensitive decision systems. Information Sciences 179, 2442–2452 (2009)
12. Moret, B.M.E.: Decision trees and diagrams. Computing Surveys 14, 593–623 (1982)
13. Murthy, S.K.: Automatic construction of decision trees from data: A multi-disciplinary survey. Data Mining and Knowledge Discovery 2, 345–389 (1998)
14. Pauker, S.G., Kassirer, J.P.: The threshold approach to clinical decision making. The New England Journal of Medicine 302, 1109–1117 (1980)
15. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Boston (1991)
16. Pawlak, Z.: Granularity of knowledge, indiscernibility and rough sets. In: Proceedings of the 1998 IEEE International Conference on Fuzzy Systems, pp. 106–110 (1998)
17. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. Wiley, New York (2008)
18. Sosnowski, L., Ślęzak, D.: How to design a network of comparators. In: BHI 2013 (2013)
19. Turney, P.D.: Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. Journal of Artificial Intelligence Research 2, 369–409 (1995)
20. Wald, A.: Sequential tests of statistical hypotheses. Annals of Mathematical Statistics 16, 117–186 (1945)
21. Yao, J.T. (ed.): Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation. Information Science Reference, Herskey (2010)
22. Yao, J.T., Vasilakos, A.V., Pedrycz, W.: Granular computing: Perspectives and challenges. IEEE Transactions on Cybernetics (2013), doi:10.1109/TSMCC, 2236648
23. Yao, Y.Y.: Granular computing: Basic issues and possible solutions. In: Proceedings of the 5th Joint Conference on Information Sciences, vol. 1, pp. 186–189 (2000)
24. Yao, Y.Y.: Probabilistic approaches to rough sets. Expert Systems 20, 287–297 (2003)
25. Yao, Y.Y.: Probabilistic rough set approximations. International Journal of Approximation Reasoning 49, 255–271 (2008)
26. Yao, Y.Y.: Granular computing: Past, present, and future. In: Proceedings of the 2008 IEEE International Conference on Granular Computing, pp. 80–85 (2008)
27. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180, 341–353 (2010)
28. Yao, Y.Y.: The superiority of three-way decisions in probabilistic rough set models. Information Sciences 181, 1080–1096 (2011)
29. Yao, Y.Y.: An outline of a theory of three-way decisions. In: Yao, J., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 1–17. Springer, Heidelberg (2012)
30. Yao, Y.Y., Deng, X.F.: Sequential three-way decisions with probabilistic rough sets. In: Proceedings of the 10th IEEE International Conference on Cognitive Informatics and Cognitive Computing, pp. 120–125 (2011)
31. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 19, 111–127 (1997)

A Scientometrics Study of Rough Sets in Three Decades

JingTao Yao and Yan Zhang

Department of Computer Science, University of Regina
{jtyao,zhang83y}@cs.uregina.ca

Abstract. Rough set theory has been attracting researchers and practitioners over three decades. The theory and its applications experienced unprecedented prosperity especially in the recent ten years. It is essential to explore and review the progress made in the field of rough sets. Mainly based on Web of Science database, we analyze the prolific authors, impact authors, impact groups, and the most impact papers in the past three decades. In addition, we also examine rough set development in the recent five years. One of the goals of this article is to use scientometrics approaches to study three decade research in rough sets. We review the historic growth of rough sets and elaborate on recent development status in this field.

1 Introduction

Rough set theory was proposed by Professor Zdzisław Pawlak in the early 1980s [30]. It is a mathematical approach to deal with inconsistent and uncertain data. The fundamental concept of rough sets is the approximation of a concept (or a crisp set) in terms of a pair of sets which give the lower and the upper approximation of the concept [30, 31].

Rough set theory has been being in a state of constant development over three decades. The related research on rough sets has attracted much attention of researchers and practitioners, who have contributed essentially to its development and applications.

One may need to study various aspects of a research domain in order to fully understand it, according to the basic principle of granular computing [59]. The research of rough sets should also be conducted in multi-aspects. There are at least three approaches in rough set research: content based approach which focuses on the content of rough set theory [61], method based approach which focuses on the constructive and algebraic (axiomatic) methods of rough sets [62], and scientometric approach which focuses on quantitatively analyzing the content and citation of rough set publications [47].

According to the result of scientometrics study, more than 80% rough sets related papers were published from 2004 to 2013. This shows that rough set research gained more popularity in the recent ten years. It is essential to explore and review the progress made in the field of rough sets.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 28–40, 2013.

We mainly use Web of Science database to conduct our research. We identify the prolific authors, impact authors, impact groups, and the most impact papers in the past three decades. We also examine the recent five years development of rough sets. The current status and the development trends of rough set theory could be identified based on the results. The research may help readers gain more understanding of developments in rough sets.

2 Scientometrics Study and Web of Science

Much research has been done in identifying research areas, trends, relationships, development and future direction [5, 46, 52]. One of representatives of such research is scientometrics which is the science that measures and analyzes science. Identification of research areas is a key theme of this area [46]. We gain more understanding of a research domain by examining its publications [57, 58].

Research impact may be measured by citations. It is suggested that a highly cited paper may have more impact than moderately cited papers [57]. This gives a simple but arguable way to measure the quality and impact of research. A study shows that of the 50 most-cited chemists, seven have been awarded the Nobel Prize [13]. In other words, the citation index may be used to predict Nobel Prize winners. According to a recent research, citation counts of the publications corresponded well with authors' own assessments of scientific contribution [2]. By analyzing citations, one may predict research influences [7]. Citation is also used as a bibliometric indicator to predict research development [46, 57]. It is suggested that the more recent or current highly cited papers in a research field, the more likely the field will grow rapidly in the near future.

We examine rough set research by exploring Thomson Reuters's Web of Science. Web of Science (http://thomson-reuters.com/web-of-science/) is one of Thomson Reuters's key products in the information age. It collects bibliographic information of research articles of high quality journals and selected international conferences. The database collects not only bibliographic information but also the information of citation relationship amongst research articles.

We start with Web of Science to locate rough set papers. The database is updated on a weekly base. The data were collected on the week of June 24-29. The data we examined were updated till June 21, 2013. Two basic measures, number of papers and number of citations, are used for popularity and influence of rough set research. A rough set paper is defined as a paper containing phrase "rough sets" or "rough set" or "rough computing" or "rough computation". We use the Topic field in Web of Science which is defined as the words or phrases within article titles, keywords, or abstracts. It should be noticed that not all rough set publications are included in the search. For instance, not all papers published in Transactions on Rough Sets are recorded.

3 Three Decades of Rough Sets

Result of querying rough set papers shows 7,088 papers are indexed by Web of Science. The total citation counts are 41,844 and h-index [4, 19] is 80. The numbers of rough set papers published in every year are shown in Fig. 1.

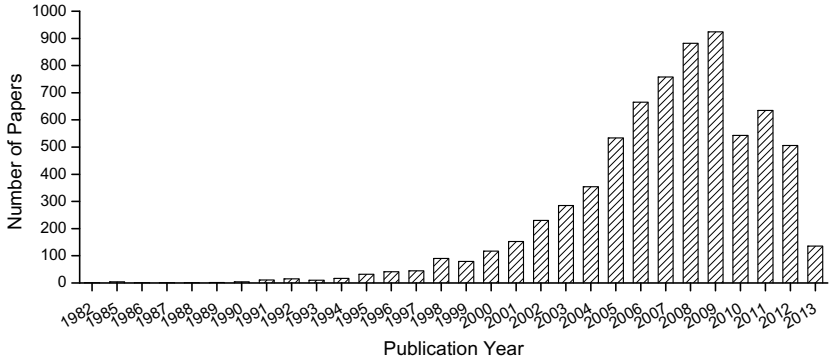


Fig. 1. Analyzing by publication years

The superficial declining of rough set publications may be affected by inclusion of rough set papers in Web of Science database. For instance, two major rough set conferences held in 2012, International Conference on Rough Sets and Knowledge Technology 2012 (RSKT'12) and International Conference on Rough Sets and Current Trends in Computing 2012 (RSCTC'12), were not included in the database. In addition, most papers in Transactions on Rough Sets after 2007 were not recorded in the database.

Only 0.71% of 7,088 papers were published in the first 12 years (from 1982 to 1993). 15.39% of 7,088 papers were published in the second 10 years (from 1994 to 2003), which is 21 times of that were published in the first 12 years. 83.9% of 7,088 papers were published in the recent 10 years (from 2004 to 2013). Among 7,088 papers, the number of international conference papers is more than 5,000, 80% of them were published in recent decade. This may show that rough set research has gained popularity and drawn attention of more researchers. Another evidence is that more international conferences were held in recent 10 years.

Web of Science provides a search feature for author and their affiliations. The results of most prolific authors, their affiliations and countries are presented in Tables 1 to 3.

Table 1 lists the most prolific authors in rough sets in term of the number of rough set papers published. The top 30 prolific authors published at least 37 rough set papers. There are more than 500 authors who published at least 5 rough set papers each. Please note that all coauthors are counted.

Table 1. Most prolific authors

Authors	Papers	Authors	Papers	Authors	Papers
Slowinski R	92	Hu QH	58	Grzymala-busse JW	43
Skowron A	91	Pal SK	57	Ramanna S	43
Yao YY	83	Miao DQ	55	Wang J	42
Wang GY	74	Chen DG	52	Polkowski L	41
Peters JF	72	Slezak D	52	Suraj Z	41
Wu WZ	70	Qian YH	48	Zhu W	40
Zhang WX	68	Li TR	47	Pawlak Z	39
Tsumoto S	67	Yu DR	46	Shi KQ	39
Greco S	66	Lin TY	44	Cheng CH	38
Liang JY	59	Ziarko W	44	Jensen R	37

Table 2. Top 15 countries or territories

Countries	Papers	Countries	Papers	Countries	Papers
P. R. China	3741	India	290	Wales	67
Poland	660	Taiwan	286	Iran	64
USA	440	Italy	120	South Korea	62
Canada	428	England	91	Germany	61
Japan	356	Malaysia	76	Spain	59

Table 2 shows top 15 countries or territories where authors are located. It is observed that People Republic of China and Poland are top 2 countries. As a matter of fact, 10 out of 20 prolific authors are from China, and top 2 are from Poland.

The top 15 institutions of the authors affiliated are shown in Table 3. The top institute is University of Regina and there were 177 papers published by authors affiliated with the University of Regina. 9 out of top 15 institutions are from China. This may explain why China is the top 1 country where authors are located. 3 out of top 15 institutions are from Poland.

Table 3. Top 15 institutions

Organizations	Papers	Organizations	Papers
Univ Regina	177	Zhejiang Univ	87
N China Elect Power Univ	154	Zhejiang Ocean Univ	80
Chinese Acad Sci	132	Shanghai Jiaotong Univ	74
Polish Acad Sci	117	Univ Manitoba	74
Harbin Inst Technol	116	Tongji Univ	73
Xian Jiaotong Univ	106	Indian Stat Inst	68
SW Jiaotong Univ	99	Polish Japanese Inst Infor Technol	68
Warsaw Univ	93		

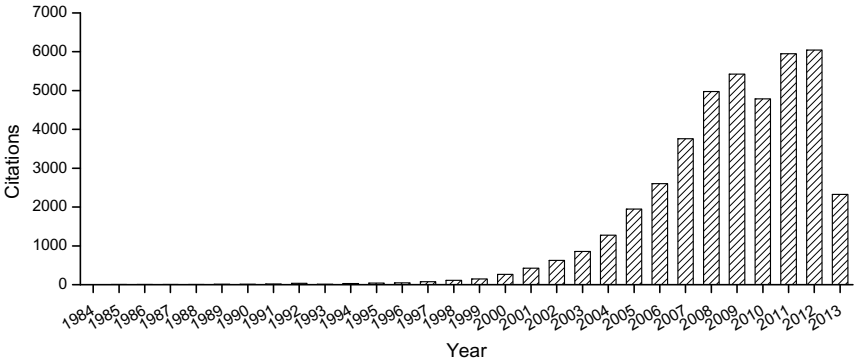


Fig. 2. Citation each year

The second analysis based on Web of Science database is citation analysis. There are 41,844 citations of 7,088 rough set papers. The average citation is 5.9 per paper. Fig. 2 shows the number of citations each year. We can see that the citations are growing every year even though the number of published papers decreased slightly since 2010. Please note that the figure showed here also includes citations from non-rough set papers. That means more papers from other research areas cite rough set papers. So we can make a prediction that applying rough set theory in other areas will be an important trend in the near future.

In order to understand more on rough set research, we identify top 20 cited papers shown in Table 4.

We may have two ranks, the total citation ranking and average citation per year. These two ranks are not always consistent. Top 5 average citations per year are ranked at 1, 4, 9, 11 and 2. We can see, 7 out of top 20 cited papers were published by Pawlak. 4 out of 7 are top 4 average citations per year. With a detailed examination of the top 20 cited papers, we can see:

- Twelve papers are about basic rough set theory, they are [29], [30], [36], [37], [38], [39], [40], [45], [61], [62], [63], [75];
- Two papers are about combining rough sets with other theories, they are [9], [54]; and
- Six papers are about applications of rough sets, they are [15], [24], [25], [33], [35], [48].

We further analyze the top 200 cited papers, and we find that applications of rough sets and combination with other theories account for relatively large proportions, which are 27% and 49.5%, respectively. The number of papers about applications of rough sets in different areas are constantly increasing. The application domains include data analysis, feature selection, decision making, incomplete information system, multi-criteria decision analysis, and three-way decision, to just name a few.

Table 4. Top cited 20 papers

	Paper	Total Citations	Average per Year	Main Results
1	Pawlak 1982 [30]	3694	115.44	Seminal paper, proposed RS
2	Ziarko 1993 [75]	659	31.38	Variable precision RS
3	Dubois+ 1990 [9]	565	23.54	Combining with fuzzy sets
4	Pawlak+ 2007 [37]	495	70.71	RS survey
5	Kryszkiewicz 1998 [24]	386	24.12	App - imcompete information table
6	Greco+ 2001 [15]	372	28.62	App - decision analysis
7	Pawlak+ 1995 [36]	352	18.53	Basic theory of RS
8	Slowinski+ 2000 [45]	323	23.07	Generalized RS
9	Pawlak+ 2007 [39]	315	45.00	RS survey
10	Mitra+ 2000 [29]	304	21.71	RS survey
11	Pawlak+ 2007 [38]	275	39.29	RS survey
12	Yao YY 1998 [62]	267	16.69	Research methods in RS
13	Kryszkiewicz 1999 [25]	249	16.60	App - imcompete information table
14	Yao YY 1998 [63]	236	14.75	Generalized RS using binary relation
15	Swiniarski+ 2003 [48]	231	21.00	App - feature selection
16	Pawlak 1998 [33]	231	14.40	App - data analysis
17	Yao YY 1996 [61]	219	12.17	Interpretation of RS
18	Wu+ 2003 [54]	205	18.64	Combining with fuzzy sets
19	Pawlak 2002 [35]	198	16.50	App - data analysis
20	Polkowski+ 1996 [40]	194	10.78	Basic theory of RS

We identified the most influential authors in the next step. We manually counted the authors of top 100 cited papers that have at least 70 citations. Table 5 lists the most impact authors whose rough set papers altogether received citations more than 300 times, the column *cts* shows the number of papers in top 100 cited papers.

Table 5. Impact authors

Authors	Cites	cts	Authors	Cites	cts	Authors	Cites	cts
Pawlak Z	5957	10	Kryszkiewicz M	749	3	Zhu W	474	4
Slowinski R	1689	8	Wu WZ	737	6	Grzymalabusse J	458	2
Skowron A	1510	5	Dubios D	641	2	Hu QH	407	3
Yao YY	1456	10	Prade H	641	2	Pal SK	392	4
Ziarko W	1088	3	Jensen R	548	5	Vanderpooten D	323	1
Zhang WX	798	6	Mi JS	545	4	Zopounidis C	312	2
Greco S	758	4	Shen Q	544	5	Yu DR	305	3
Matarazzo B	758	4	Mitra S	520	3	Hayashi Y	304	1

Each of the top five authors has more than 1,000 citations. Pawlak is the inventor and pioneer of rough set theory. Slowinski contributed mainly in rough set based decision making and dominance rough sets. The dominance-based rough set approach is the substitution of the indiscernibility relation by a dominance relation, which permits the formalism to deal with inconsistencies typical in consideration of criteria and preference-ordered decision classes [15]. Skowron's discernibility matrix led many research and algorithms on reduct construction. Yao YY contributed mainly to generalized rough sets in general and probabilistic rough sets and decision-theoretic rough sets in specific [61, 62, 65, 66]. The decision-theoretic rough set model introduces a pair of threshold on probabilities to define probabilistic regions and give a systematic method for interpreting and determining the thresholds based on Bayesian decision theory [65]. A recent proposal by Herbert and Yao [18] gives a new method for determining the threshold based on game theory. Ziarko proposed variable precision rough sets which improve on the traditional rough set approach and accept classification error by using user provided the lower boundary and the upper boundary [23, 75].

4 Recent Development of Rough Sets

We observe the development of rough sets in recent 5 years in order to get a deep understanding of rough set current status. Recent 5 years refer to 2008 to 2012 since most papers published in 2013 have not been recorded in databases. Result of querying rough set papers setting Timespan as 2008 to 2012 shows 3,496 papers are indexed by Web of Science. The total citation counts are 7,777. H-index is 33. The special h-index is called h5-index as defined by Google [14]. It is defined as the h-index for articles published in the last 5 complete years.

Table 6 shows the top 20 cited papers published during 2008 to 2012. With a detailed examination of the top 20 cited papers published between 2008 and 2012, we found that:

- Eight papers are about basic rough set theory, they are [28], [41], [42], [49], [56], [66], [73], [74];
- Five papers are about combining rough sets with other theories, they are [3], [10], [11], [12], [53]; and
- Seven papers are about applications of rough sets, they are [20], [21], [22], [26], [50], [55], [68].

It is noted that there are many new and young researchers, many of them from China, contributed to the highly cited papers in recent five year. We may notice that most of highly cited papers in last five years are extensions and applications of existing research, compared with top cited papers in Table 4. There is a need for new ideas and development. The theory of three-way decisions, motivated by rough set three-regions but goes beyond rough sets, is a promising research direction that may lead to new breakthrough.

Table 6. Top 20 cited papers in recent 5 years

	Paper	Total Citations	Average per Year	Main Results
1	Feng+ 2008 [10]	91	15.17	Soft sets
2	Yao YY+ 2008 [73]	81	13.50	Reduction in DTRS
3	Yao YY 2008 [66]	76	12.67	Probabilistic rough sets
4	Hu+ 2008 [20]	75	12.50	App - feature subset selection
5	Zhu 2009 [74]	69	13.80	Generalized RS
6	Hu+ 2008 [21]	65	10.83	App - neighborhood classifier
7	Jensen+ 2009 [22]	64	12.80	App - feature selection
8	Wu 2008 [53]	60	10.00	Attribute reduction
9	Qian+ 2010 [42]	55	13.75	Reduction accelerator
10	Wang+ 2008 [50]	52	8.67	App - rule induction
11	Thangavel+ 2009 [49]	48	9.60	Reduction (survey)
12	Liu 2008 [28]	48	8.00	Generalized RS
13	Qian+ 2008 [41]	48	8.00	Measures
14	Yang+ 2008 [56]	45	7.50	Dominance RS
15	Feng+ 2010 [11]	44	11.00	Soft sets
16	Yao YY 2010 [68]	41	10.25	Introduced three-way decision
17	Xiao+ 2009 [55]	41	8.20	App - forecasting
18	Bai+ 2010 [3]	38	9.50	Combining with grey system
19	Li+ 2008 [26]	38	6.33	App - prediction
20	Feng+ 2011 [12]	37	12.33	Soft sets

5 Concluding Remarks

We use scientometrics approach to examine the development of rough sets in this article. Prolific authors, impact authors, as well as most impact papers were identified based on Web of Science. It is observed that rough sets has been in a state of constant development. We can see that applying rough sets to different areas become more important. In order to broaden and deepen the study of rough sets, combining rough sets with other theories should be emphasized. It is hoped that readers may gain more understanding of the current status and development of rough sets.

The original rough set theory was defined by an equivalent relation, or equivalently a partition and Boolean algebra. Based on these definitions, rough set theory is generalized as: binary relation based rough set theory, covering based rough set theory, and subsystem based rough set theory [63, 70, 72]. These generalization increase our understanding of the theory.

Rough set theory can be considered as an independent discipline in its own right [39]. Based on the original theory, rough set theory has achieved substantial progress and applied to various application domains. Probabilistic rough sets apply probabilistic approaches to rough set theory, which weaken the strict limitations of Pawlak rough sets in order to increase the applicability of theory [66]. Probabilistic rough sets are considered as one of the important and prolific

research extensions. Probabilistic approaches to rough sets appear in many forms, such as decision-theoretic rough set model [65,71], Bayesian rough set model [44], game-theoretic rough set model [18]. The game-theoretic rough set provides an alternative way to determine effective probabilistic thresholds by formulating competition or cooperation among multiple criteria [18]. In addition, parameterized rough set model determines three regions through the concurrent use of two pairs of parameters [16, 17].

Moreover, rough set theory has been combined with other theories, such as fuzzy sets [9,54], granular computing [59,64], and neural network [1], etc. Three-way decision making is a new research proposed in 2009 [67, 69]. Three-way decision making can be benefited with many theories and methods including rough sets, shadowed sets, and approximation of fuzzy sets. Ciucci et al. [6] adopted the square of opposition, cube of opposition and hexagon of opposition to give a geometric view of relations between entities in rough sets. Yao argued that the same framework can be used to study relationships between regions of three-way decision [60].

The extent of rough set applications become much wider, including data analysis [31,33], feature selection [48,51], rules mining or decision making [27,32,34], incomplete information system [24,25], multicriteria decision analysis [15], business prediction [8], fault diagnosis [43], etc. It is expected that there will be more exploration on combining rough set theory with other theories. Applications will be remain as a trend in rough set research.

We have also conducted analyses with other databases, such as IEEE Digital Library, Google scholar, and Inspec. The results more or less confirm with findings reported here. Further detailed analyses and studies on classification, different school of thoughts, and other remaining research challenges will be reported in sequence articles.

Acknowledgements. This research was partially supported by a discovery grant from NSERC Canada.

References

1. Ahn, B.S., Cho, S.S., Kim, C.Y.: The integrated methodology of rough set theory and artificial neural network for business failure prediction. Expert Systems with Applications 18(2), 65–74 (2000)
2. Aksnes, D.W.: Citation rates and perceptions of scientific contribution. Journal of the American Society for Information Science and Technology 57(2), 169–185 (2006)
3. Bai, C.G., Sarkis, J.: Integrating sustainability into supplier selection with grey system and rough set methodologies. International Journal of Production Economics 124(1), 252–264 (2010)
4. Bar-Ilan, J.: Which h-index? A comparison of WoS, Scopus and Google Scholar. Scientometrics 74(2), 257–271 (2008)
5. Cardinal, B.J., Thomas, J.R.: The 75th anniversary of research quarterly for exercise and sport: An analysis of status and contributions. Research Quarterly for Exercise and Sport 76(suppl. 2), S122–S134 (2005)

6. Ciucci, D., Dubois, D., Prade, H.: Oppositions in rough set theory. 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. 504–513. Springer, Heidelberg (2012)
7. Dietz, L., Bickel, S., Scheffer, T.: Unsupervised prediction of citation influences. In: Proceedings of the 24th International Conference on Machine Learning, pp. 233–240. ACM (2007)
8. Dimitras, A., Slowinski, R., Susmaga, R., Zopounidis, C.: Business failure prediction using rough sets. European Journal of Operational Research 114(2), 263–280 (1999)
9. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. International Journal of General System 17(2-3), 191–209 (1990)
10. Feng, F., Jun, Y.B., Zhao, X.Z.: Soft semirings. Computers & Mathematics with Applications 56(10), 2621–2628 (2008)
11. Feng, F., Li, C.X., Davvaz, B., Ali, M.I.: Soft sets combined with fuzzy sets and rough sets: a tentative approach. Soft Computing 14(9), 899–911 (2010)
12. Feng, F., Liu, X.Y., Leoreanu-Fotea, V., Jun, Y.B.: Soft sets and soft rough sets. Information Sciences 181(6), 1125–1137 (2011)
13. Garfield, E., Welljams-Dorof, A.: Of nobel class: A citation perspective on high impact research authors. Theoretical Medicine 13, 117–135 (1992)
14. Google: Google Scholar Metrics (2013), `http://scholar.google.com/intl/en/scholar/metrics.html/` (accessed July 09, 2013)
15. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129(1), 1–47 (2001)
16. Greco, S., Matarazzo, B., Słowiński, R.: Rough membership and bayesian confirmation measures for parameterized rough sets. In: Ślęzak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 314–324. Springer, Heidelberg (2005)
17. Greco, S., Matarazzo, B., Słowiński, R.: Parameterized rough set model using rough membership and bayesian confirmation measures. International Journal of Approximate Reasoning 49(2), 285–300 (2008)
18. Herbert, J.P., Yao, J.T.: Game-theoretic rough sets. Fundamenta Informaticae 108(3-4), 267–286 (2011)
19. Hirsch, J.E.: An index to quantify an individual's scientific research output. Proceedings of the National Academy of Sciences of the United States of America 102(46), 16569–16572 (2005)
20. Hu, Q.H., Yu, D.R., Liu, J.F., Wu, C.X.: Neighborhood rough set based heterogeneous feature subset selection. Information Sciences 178(18), 3577–3594 (2008)
21. Hu, Q.H., Yu, D.R., Xie, Z.X.: Neighborhood classifiers. Expert Systems with Applications 34(2), 866–876 (2008)
22. Jensen, R., Shen, Q.: New approaches to fuzzy-rough feature selection. IEEE Transactions on Fuzzy Systems 17(4), 824–838 (2009)
23. Katzberg, J.D., Ziarko, W.: Variable precision rough sets with asymmetric bounds. In: Rough Sets, Fuzzy Sets and Knowledge Discovery, pp. 167–177. Springer (1994)
24. Kryszkiewicz, M.: Rough set approach to incomplete information systems. Information Sciences 112(1), 39–49 (1998)
25. Kryszkiewicz, M.: Rules in incomplete information systems. Information Sciences 113(3), 271–292 (1999)
26. Li, H., Sun, J.: Ranking-order case-based reasoning for financial distress prediction. Knowledge-Based Systems 21(8), 868–878 (2008)

27. Li, R.P., Wang, Z.O.: Mining classification rules using rough sets and neural networks. European Journal of Operational Research 157(2), 439–448 (2004)
28. Liu, G.L.: Generalized rough sets over fuzzy lattices. Information Sciences 178(6), 1651–1662 (2008)
29. Mitra, S., Hayashi, Y.: Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Transactions on Neural Networks 11(3), 748–768 (2000)
30. Pawlak, Z.: Rough sets. International Journal of Parallel Programming 11(5), 341–356 (1982)
31. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academatic Publishers, Boston (1991)
32. Pawlak, Z.: Rough set approach to knowledge-based decision support. European Journal of Operational Research 99(1), 48–57 (1997)
33. Pawlak, Z.: Rough set theory and its applications to data analysis. Cybernetics & Systems 29(7), 661–688 (1998)
34. Pawlak, Z.: Decision rules, Bayes' rule and rough sets. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 1–9. Springer, Heidelberg (1999)
35. Pawlak, Z.: Rough sets, decision algorithms and Bayes' theorem. European Journal of Operational Research 136(1), 181–189 (2002)
36. Pawlak, Z., Grzymala-Busse, J., Slowinski, R., Ziarko, W.: Rough sets. Communications of the ACM 38(11), 88–95 (1995)
37. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
38. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. Information Sciences 177(1), 41–73 (2007)
39. Pawlak, Z., Skowron, A.: Rough sets: some extensions. Information Sciences 177(1), 28–40 (2007)
40. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. International Journal of Approximate Reasoning 15(4), 333–365 (1996)
41. Qian, Y.H., Liang, J.Y., Li, D.Y., Zhang, H.Y., Dang, C.Y.: Measures for evaluating the decision performance of a decision table in rough set theory. Information Sciences 178(1), 181–202 (2008)
42. Qian, Y.H., Liang, J.Y., Pedrycz, W., Dang, C.Y.: Positive approximation: An accelerator for attribute reduction in rough set theory. Artificial Intelligence 174(9), 597–618 (2010)
43. Shen, L.X., Tay, F.E., Qu, L.S., Shen, Y.D.: Fault diagnosis using rough sets theory. Computers in Industry 43(1), 61–72 (2000)
44. Ślęzak, D., Ziarko, W.: The investigation of the bayesian rough set model. International Journal of Approximate Reasoning 40(1), 81–91 (2005)
45. 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)
46. Small, H.: Tracking and predicting growth areas in science. Scientometrics 68(3), 595–610 (2006)
47. Suraj, Z., Grochowalski, P., Lew, Ł.: Discovering patterns of collaboration in rough set research: Statistical and graph-theoretical approach. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 238–247. Springer, Heidelberg (2011)
48. Swiniarski, R.W., Skowron, A.: Rough set methods in feature selection and recognition. Pattern Recognition Letters 24(6), 833–849 (2003)

49. Thangavel, K., Pethalakshmi, A.: Dimensionality reduction based on rough set theory: A review. Applied Soft Computing 9(1), 1–12 (2009)
50. Wang, X.Z., Zhai, J.H., Lu, S.X.: Induction of multiple fuzzy decision trees based on rough set technique. Information Sciences 178(16), 3188–3202 (2008)
51. Wang, X.Y., Yang, J., Teng, X.L., Xia, W.J., Jensen, R.: Feature selection based on rough sets and particle swarm optimization. Pattern Recognition Letters 28(4), 459–471 (2007)
52. White, H.D., McCain, K.W.: Visualizing a discipline: An author co-citation analysis of information science, 1972-1995. Journal of the American Society for Information Science 49(4), 327–355 (1998)
53. Wu, W.Z.: Attribute reduction based on evidence theory in incomplete decision systems. Information Sciences 178(5), 1355–1371 (2008)
54. Wu, W.Z., Mi, J.S., Zhang, W.X.: Generalized fuzzy rough sets. Information Sciences 151, 263–282 (2003)
55. Xiao, Z., Gong, K., Zou, Y.: A combined forecasting approach based on fuzzy soft sets. Journal of Computational and Applied Mathematics 228(1), 326–333 (2009)
56. Yang, X.B., Yang, J.Y., Wu, C., Yu, D.J.: Dominance-based rough set approach and knowledge reductions in incomplete ordered information system. Information Sciences 178(4), 1219–1234 (2008)
57. Yao, J.T.: A ten-year review of granular computing. In: IEEE International Conference on Granular Computing, pp. 734–734. IEEE (2007)
58. Yao, J.T.: Recent developments in granular computing: a bibliometrics study. In: IEEE International Conference on Granular Computing, pp. 74–79. IEEE (2008)
59. Yao, J.T., Vasilakos, A.V., Pedrycz, W.: Granular computing: Perspectives and challenges. IEEE Transactions on Cybernetics PP(99), 1–13 (2013)
60. Yao, Y.Y.: Duality in rough set theory on square if opposition, doi:10.3233/FI-2013-881
61. Yao, Y.Y.: Two views of the theory of rough sets in finite universes. International Journal of Approximate Reasoning 15(4), 291–317 (1996)
62. Yao, Y.Y.: Constructive and algebraic methods of the theory of rough sets. Information Sciences 109(1), 21–47 (1998)
63. Yao, Y.Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111(1), 239–259 (1998)
64. Yao, Y.Y.: Information granulation and rough set approximation. International Journal of Intelligent Systems 16(1), 87–104 (2001)
65. Yao, Y.Y.: Decision-theoretic rough set models. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 1–12. Springer, Heidelberg (2007)
66. Yao, Y.Y.: Probabilistic rough set approximations. International Journal of Approximate Reasoning 49(2), 255–271 (2008)
67. Yao, Y.Y.: Three-way decision: An interpretation of rules in rough set theory. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 642–649. Springer, Heidelberg (2009)
68. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180(3), 341–353 (2010)
69. Yao, Y.Y.: An outline of a theory of three-way decisions. In: Yao, J., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 1–17. Springer, Heidelberg (2012)
70. Yao, Y.Y., Chen, Y.H.: Subsystem based generalizations of rough set approximations. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 210–218. Springer, Heidelberg (2005)

71. 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)
72. Yao, Y.Y., Yao, B.X.: Covering based rough set approximations. Information Sciences 200 (2012)
73. Yao, Y.Y., Zhao, Y.: Attribute reduction in decision-theoretic rough set models. Information Sciences 178(17), 3356–3373 (2008)
74. Zhu, W.: Relationship between generalized rough sets based on binary relation and covering. Information Sciences 179(3), 210–225 (2009)
75. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46(1), 39–59 (1993)

Generalizations of Approximations

Patrick G. Clark[1], Jerzy W. Grzymała-Busse[1,2], and Wojciech Rząsa[3]

[1] Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA
{pclark,jerzy}@ku.edu, wrzasa@univ.rzeszow.pl
[2] Institute of Computer Science, Polish Academy of Sciences, 01-237 Warsaw, Poland
[3] Department of Computer Science, Rzeszow University, 35-310 Rzeszow, Poland

Abstract. In this paper we consider a generalization of the indiscernibility relation, i.e., a relation R that is not necessarily reflexive, symmetric, or transitive. There exist 36 basic definitions of lower and upper approximations based on such relation R. Additionally, there are six probabilistic approximations, generalizations of 12 corresponding lower and upper approximations. How to convert remaining 24 lower and upper approximations to 12 respective probabilistic approximations is an open problem.

1 Introduction

Rough set theory is based on ideas of lower and upper approximations. For completely defined data sets such approximations are defined using an *indiscerniblity* relation R [25, 26], an equivalence relation. A probabilistic approximation, a generalization of lower and upper approximations, was introduced in [36] and then studied in many papers, e.g., [19, 27–29, 34, 40, 42–45]. Probabilistic approximations are defined using an additional parameter, interpreted as probability, and denoted by α. Lower and upper approximations are special cases of the probability approximation, if $\alpha = 1$, the probabilistic approximation becomes the lower approximation; if α is quite small, the probabilistic approximation is equal to the upper approximation.

Some data sets, e.g., incomplete data sets, are described by relations that are not equivalence relations [8, 9]. Lower and upper approximations for such a relation R that does not need to be reflexive, symmetric or transitive were studied in many papers as well. Corresponding definitions were summarized in [14, 16], where also basic properties were studied. There exist 36 basic definitions of lower and upper approximations based on such general relation R. These lower and upper approximations were generalized to probabilistic approximations in [11]. There are six such probabilistic approximations, generalizations of 12 corresponding lower and upper approximations, since a probabilistic approximation, with α between 0 and 1, represents the entire spectrum of approximations, including lower and upper approximations. How to convert remaining 24 lower and upper approximations to 12 respective probabilistic approximations is an open problem.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 41–52, 2013.

2 Equivalence Relations

First we will quote some definitions for complete data sets that are characterized by an equivalence relation, namely, by the indiscernibility relation [25, 26].

2.1 Lower and Upper Approximations

The set of all cases of a data set is denoted by U. 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. For a case x, the value of an attribute a will be denoted by $a(x)$. If for any $a \in A$ and $x \in U$ the value $a(x)$ is specified, the data set is called *completely specified*, or *complete*.

Rough set theory, see, e.g., [25] and [26], is based on the idea of an indiscernibility relation, defined for complete data sets. Let B be a nonempty subset of the set A of all attributes. The indiscernibility relation $IND(B)$ is a relation on U defined for $x, y \in U$ by

$$(x, y) \in IND(B) \text{ if and only if } a(x) = a(y) \text{ for all } a \in B.$$

A complete data set may be described by an (U, R) called an *approximation space*, where R is an indiscernibility relation $IND(B)$ on U.

The indiscernibility relation $IND(B)$ is an equivalence relation. Equivalence classes of $IND(B)$ are called *elementary sets* of B and are denoted by $[x]_B$. For completely specified data sets lower and upper approximations are defined on the basis of the indiscernibility relation. Any finite union of elementary sets, associated with B, will be called a *B-definable set*. Let X be any subset of the set U of all cases. The set X is called a *concept* and is usually defined as the set of all cases defined by a specific value of the decision. In general, X is not a B-definable set. However, set X may be approximated by two B-definable sets, the first one is called a *B-lower approximation* of X, denoted by $\underline{B}X$ and defined by

$$\{x \in U | [x]_B \subseteq X\}.$$

The second set is called a B-upper approximation of X, denoted by $\overline{B}X$ and defined by

$$\{x \in U | [x]_B \cap X \neq \emptyset\}.$$

The above shown way of computing lower and upper approximations, by constructing these approximations from singletons x, will be called the *first method*. The B-lower approximation of X is the greatest B-definable set, contained in X. The B-upper approximation of X is the smallest B-definable set containing X.

As it was observed in [26], for complete data sets we may use a *second method* to define the B-lower approximation of X, by the following formula

$$\cup\{[x]_B | x \in U, [x]_B \subseteq X\},$$

and the B-upper approximation of x may be defined, using the second method, by

$$\cup\{[x]_B | x \in U, [x]_B \cap X \neq \emptyset).$$

Note that for a binary relation R that is not an equivalence relation these two methods lead, in general, to different results.

2.2 Probabilistic Approximations

Let (U, R) be an approximation space, where R is an equivalence relation on U. A probabilistic approximation of the set X with the threshold α, $0 < \alpha \leq 1$, is denoted by $appr_\alpha(X)$ and defined by

$$\cup\{[x] \mid x \in U,\ Pr(X|[x]) \geq \alpha\},$$

where $[x]$ is an elementary set of R and $Pr(X|[x]) = \frac{|X \cap [x]|}{|[x]|}$ is the conditional probability of X given $[x]$.

Obviously, for the set X, the probabilistic approximation of X computed for the threshold equal to the smallest positive conditional probability $Pr(X \mid [x])$ is equal to the standard upper approximation of X. Additionally, the probabilistic approximation of X computed for the threshold equal to 1 is equal to the standard lower approximation of X.

3 Arbitrary Binary Relations

In this section we will discuss first lower and upper approximations and then probabilistic approximations based on an arbitrary binary relation R.

3.1 Lower and Upper Approximations

First we will quote some definitions from [14, 16]. Let U be a finite nonempty set, called the *universe*, let R be a binary relation on U, and let x be a member of U. The relation R is a generalization of the indiscernibility relation. In general, R does not need to be reflexive, symmetric, or transitive. Basic granules defined by a relation R are called R-*successor* and R-*predecessor* sets.

An R-*successor* set of x, denoted by $R_s(x)$, is defined by

$$R_s(x) = \{y \mid xRy\}.$$

An R-*predecessor* set of x, denoted by $R_p(x)$, is defined by

$$R_p(x) = \{y \mid yRx\}.$$

Let X be a subset of U. A set X is R-*successor definable* if and only if $X = \emptyset$ or X is a union of some R-successor sets.

A set X is R-*predecessor definable* if and only if $X = \emptyset$ or X is a union of some R-predecessor sets.

Singleton, Subset and Concept Approximations. An R-*singleton successor lower approximation* of X, denoted by $\underline{appr}_s^{singleton}(X)$, is defined by

$$\{x \in U \mid R_s(x) \subseteq X\}.$$

The singleton successor lower approximations were studied in many papers, see, e.g., [8, 9, 20–23, 30–33, 35, 37–39, 41].

An R-*singleton predecessor lower approximation* of X, denoted by $\underline{appr}_p^{singleton}(X)$, is defined as follows

$$\{x \in U \mid R_p(x) \subseteq X\}.$$

The singleton predecessor lower approximations were studied in [30].

An R-*singleton successor upper approximation* of X, denoted by $\overline{appr}_s^{singleton}(X)$, is defined as follows

$$\{x \in U \mid R_s(x) \cap X \neq \emptyset\}.$$

The singleton successor upper approximations, like singleton successor lower approximations, were also studied in many papers, e.g., [8, 9, 20, 21, 30–33, 35, 37–39, 41].

An R-*singleton predecessor upper approximation* of X, denoted by $\overline{appr}_p^{singleton}(X)$, is defined as follows

$$\{x \in U \mid R_p(x) \cap X \neq \emptyset\}.$$

The singleton predecessor upper approximations were introduced in [30].

An R-*subset successor lower approximation* of X, denoted by $\underline{appr}_s^{subset}(X)$, is defined by

$$\cup \{R_s(x) \mid x \in U \text{ and } R_s(x) \subseteq X\}.$$

The subset successor lower approximations were introduced in [8, 9].

An R-*subset predecessor lower approximation* of X, denoted by $\underline{appr}_p^{subset}(X)$, is defined by

$$\cup \{R_p(x) \mid x \in U \text{ and } R_p(x) \subseteq X\}.$$

The subset predecessor lower approximations were studied in [30].

An R-*subset successor upper approximation* of X, denoted by $\overline{appr}_s^{subset}(X)$, is defined by

$$\cup \{R_s(x) \mid x \in U \text{ and } R_s(x) \cap X \neq \emptyset\}.$$

The subset successor upper approximations were introduced in [8, 9].

An R-*subset predecessor upper approximation* of X, denoted by $\overline{appr}_p^{subset}(X)$, is defined by

$$\cup \{R_p(x) \mid x \in U \text{ and } R_p(x) \cap X \neq \emptyset\}.$$

The subset predecessor upper approximations were studied in [30].

An R-*concept successor lower approximation* of X, denoted by $\underline{appr}_s^{concept}(X)$, is defined by

$$\cup \{R_s(x) \mid x \in X \text{ and } R_s(x) \subseteq X\}.$$

The concept successor lower approximations were introduced in [8, 9].

An R-*concept predecessor lower approximation* of X, denoted by $\underline{appr}_p^{concept}(X)$, is defined by

$$\cup \{R_p(x) \mid x \in X \text{ and } R_p(x) \subseteq X\}.$$

The concept predecessor lower approximations were introduced, for the first time, in [13].

An R-*concept successor upper approximation* of X, denoted by $\overline{appr}_s^{concept}(X)$, is defined by

$$\cup \{R_s(x) \mid x \in X \text{ and } R_s(x) \cap X \neq \emptyset\}$$

The concept successor upper approximations were studied in [8, 9, 23].

An R-*concept predecessor upper approximation* of X, denoted by $\overline{appr}_p^{concept}(X)$, is defined by

$$\cup \{R_p(x) \mid x \in X \text{ and } R_p(x) \cap X \neq \emptyset\}$$

The concept predecessor upper approximations were studied in [30].

Sets $\underline{appr}_s^{subset}(X)$, $\underline{appr}_s^{concept}(X)$, $\overline{appr}_s^{subset}(X)$, $\overline{appr}_s^{concept}(X)$ and $\overline{appr}_p^{singleton}(X)$ are R-successor definable, while sets $\underline{appr}_p^{subset}(X)$, $\underline{appr}_p^{concept}(X)$, $\overline{appr}_p^{subset}(X)$, $\overline{appr}_p^{concept}(X)$ and $\overline{appr}_s^{singleton}(X)$ are R-predecessor definable for any approximation space (U, R), see. e.g., [8, 10, 24].

Modified Singleton Approximations. Definability and duality of lower and upper approximations of a subset X of the universe U are basic properties of rough approximations defined for the standard lower and upper approximations [25, 26].

To avoid problems with inclusion for singleton approximations, the following modification of the corresponding definitions were introduced in [14]:

An R-*modified singleton successor lower approximation* of X, denoted by $\underline{appr}_s^{modsingleton}(X)$, is defined by

$$\{x \in U \mid R_s(x) \subseteq X \text{ and } R_s(x) \neq \emptyset\}.$$

An R-*modified singleton predecessor lower approximation* of X, denoted by $\underline{appr}_p^{modsingleton}(X)$, is defined by

$$\{x \in U \mid R_p(x) \subseteq X \text{ and } R_p(x) \neq \emptyset\}.$$

An R-*modified singleton successor upper approximation* of X, denoted by $\overline{appr}_s^{modsingleton}(X)$, is defined by

$$\{x \in U \mid R_s(x) \cap X \neq \emptyset \text{ or } R_s(x) = \emptyset\}.$$

An R-*modified singleton predecessor upper approximation* of X, denoted by $\overline{appr}_p^{modsingleton}(X)$, is defined by

$$\{x \in U \mid R_p(x) \cap X \neq \emptyset \text{ or } R_p(x) = \emptyset\}.$$

Largest Lower and Smallest Upper Approximations. For any relation R, the R-subset successor (predecessor) lower approximation of X is the largest R-successor (predecessor) definable set contained in X. It follows directly from the definition.

On the other hand, the smallest R-successor definable set containing X does not need to be unique. It was observed, for the first time, in [13].

Any R-smallest successor upper approximation, denoted by $\overline{appr}_s^{smallest}(X)$, is defined as a R-successor definable set with the smallest cardinality containing X. An R-smallest successor upper approximation does not need to be unique.

An R-smallest predecessor upper approximation, denoted by $\overline{appr}_p^{smallest}(X)$, is defined as an R-predecessor definable set with the smallest cardinality containing X. Likewise, an R-smallest predecessor upper approximation does not need to be unique.

Dual Approximations. As it was shown in [38], singleton approximations are *dual* for any relation R. In [16] it was proved that modified singleton approximations are also dual. On the other hand it was shown in [38] that if R is not an equivalence relation then subset approximations are not dual. Moreover, concept approximations are not dual as well, unless R is reflexive and transitive [14].

Two additional approximations were defined in [38]. The first approximation, denoted by $\underline{appr}_s^{dualsubset}(X)$, was defined by

$$\neg(\overline{appr}_s^{subset}(\neg X))$$

while the second one, denoted by $\overline{\underline{appr}}_s^{dualsubset}(X)$ was defined by

$$\neg(\underline{appr}_s^{subset}(\neg X)),$$

where $\neg X$ denotes the complement of X.

These approximations are called an *R-dual subset successor lower* and *R-dual subset successor upper approximations*, respectively. Obviously, we may define as well an *R-dual subset predecessor lower approximation*

$$\neg(\overline{appr}_p^{subset}(\neg X))$$

and an *R-dual subset predecessor upper approximation*

$$\neg(\underline{appr}_p^{subset}(\neg X)).$$

By analogy we may define dual concept approximations. Namely, an *R-dual concept successor lower approximation* of X, denoted by $\underline{appr}_s^{dualconcept}(X)$ is defined by

$$\neg(\overline{appr}_s^{concept}(\neg X)).$$

An *R-dual concept successor upper approximation* of X, denoted by $\overline{appr}_s^{dualconcept}(X)$ is defined by

$$\neg(\underline{appr}_s^{concept}(\neg X)).$$

The set denoted by $\underline{appr}_p^{dualconcept}(X)$ and defined by the following formula

$$\neg(\overline{appr}_p^{concept}(\neg X))$$

will be called an *R-dual concept predecessor lower approximation*, while the set $\overline{appr}_p^{dualconcept}(X)$ defined by the following formula

$$\neg(\underline{appr}_p^{concept}(\neg X))$$

will be called an *R-dual concept predecessor upper approximation*.

These four R-dual concept approximations were introduced in [14].

Again, by analogy we may define dual approximations for the smallest upper approximations. The set, denoted by $\underline{appr}_s^{dualsmallest}(X)$ and defined by

$$\neg(\overline{appr}_s^{smallest}(\neg X)),$$

will be called an *R-dual smallest successor lower approximation* of X while the set denoted by $\underline{appr}_p^{dualsmallest}(X)$ and defined by

$$\neg(\overline{appr}_p^{smallest}(\neg X)).$$

will be called an *R-dual smallest predecessor lower approximation* of X.

These two approximations were introduced in [16].

Approximations with Mixed Idempotency. Smallest upper approximations, introduced in Section 3.1, and subset lower approximations are the only approximations discussed so far that satisfy the Mixed Idempotency Property, so

$$\underline{appr}_s(X) = \overline{appr}_s(\underline{appr}_s(X))(\underline{appr}_p(X) = \overline{appr}_p(\underline{appr}_p(X))), \tag{1}$$

and

$$\overline{appr}_s(X) = \underline{appr}_s(\overline{appr}_s(X))(\overline{appr}_p(X) = \underline{appr}_p(\overline{appr}_p(X))). \tag{2}$$

For the following approximations, defined sets satisfy the above two conditions. The upper approximation, denoted by $\overline{appr}_s^{subset\text{-}concept}(X)$ and defined by

$$\underline{appr}_s^{subset}(X) \cup \bigcup \{R_s(x) \mid x \in X - \underline{appr}_s^{subset}(X) \text{ and } R_s(x) \cap X \neq \emptyset\}$$

will be called an *R-subset-concept successor upper approximation* of X.

The upper approximation, denoted by $\overline{appr}_p^{subset\text{-}concept}(X)$ and defined by

$$\underline{appr}_p^{subset}(X) \cup \bigcup \{R_p(x) \mid x \in X - \underline{appr}_p^{subset}(X) \text{ and } R_p(x) \cap X \neq \emptyset\}$$

will be called an *R-subset-concept predecessor upper approximation* of X. The upper approximation, denoted by $\overline{appr}_s^{subset\text{-}subset}(X)$ and defined by

$$\underline{appr}_s^{subset}(X) \cup \bigcup \{R_s(x) \mid x \in U - \underline{appr}_s^{subset}(X) \text{ and } R_s(x) \cap X \neq \emptyset\}$$

will be called an *R-subset-subset successor upper approximation* of X.

The upper approximation, denoted by $\overline{appr}_p^{subset\text{-}subset}(X)$ and defined by

$$\underline{appr}_p^{subset}(X) \cup \bigcup \{R_p(x) \mid x \in U - \underline{appr}_p^{subset}(X) \text{ and } R_p(x) \cap X \neq \emptyset\}$$

will be called an *R-subset-subset predecessor upper approximation* of X.

These four upper approximations, together with $\underline{appr}_s^{subset}$ (or $\underline{appr}_p^{subset}$, respectively), satisfy Mixed Idempotency Property.

Note that for these four upper approximations corresponding dual lower approximations may be defined as well. These definitions are skipped since they are straightforward.

3.2 Probabilistic Approximations

By analogy with standard approximations defined for arbitrary binary relations, we will introduce three kinds of probabilistic approximations for such relations: singleton, subset and concept. For simplicity, we restrict our attention only to R-successor sets as the basic granules. Obviously, analogous three definitions based on R-predecessor sets may be easily introduced as well.

A *singleton probabilistic approximation* of X with the threshold α, $0 < \alpha \leq 1$, denoted by $appr_{\alpha}^{singleton}(X)$, is defined by

$$\{x \mid x \in U,\ Pr(X|R_s(x)) \geq \alpha\},$$

where $Pr(X|R_s(x)) = \frac{|X \cap R_s(x)|}{|R_s(x)|}$ is the conditional probability of X given $R_s(x)$.

A *subset probabilistic approximation* of the set X with the threshold α, $0 < \alpha \leq 1$, denoted by $appr_{\alpha}^{subset}(X)$, is defined by

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

A *concept probabilistic approximation* of the set X with the threshold α, $0 < \alpha \leq 1$, denoted by $appr_{\alpha}^{concept}(X)$, is defined by

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

Obviously, for the concept X, the probabilistic approximation of a given type (singleton, subset or concept) of X computed for the threshold equal to the smallest positive conditional probability $Pr(X \mid R_s(x))$ is equal to the standard upper approximation of X of the same type. Additionally, the probabilistic approximation of a given type of X computed for the threshold equal to 1 is equal to the standard lower approximation of X of the same type.

Results of many experiments on probabilistic approximations were published in [1–7, 12, 17, 18].

4 Conclusions

We discussed 36 basic definitions of lower and upper approximations based on a relation R that is not an equivalence relation. For such a relation R, there are six probabilistic approximations, generalizations of 12 corresponding lower and upper approximations. How to convert remaining 24 lower and upper approximations to 12 respective probabilistic approximations is an open problem.

Note that other definitions of approximations, called *local*, were discussed in [6, 13, 15]. First, local lower and upper approximations were introduced in [13, 15], then these approximations were generalized to probabilistic in a few different ways in [6].

References

1. 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)
2. Clark, P.G., Grzymala-Busse, J.W.: Experiments on rule induction from incomplete data using three probabilistic approximations. In: Proceedings of the 2012 IEEE International Conference on Granular Computing, pp. 90–95 (2012)
3. Clark, P.G., Grzymala-Busse, J.W.: Experiments using three probabilistic approximations for rule induction from incomplete data sets. In: Proceeedings of the MCCSIS 2012, IADIS European Conference on Data Mining, ECDM 2012, pp. 72–78 (2012)
4. 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)
5. Clark, P.G., Grzymala-Busse, J.W., Hippe, Z.S.: How good are probabilistic approximations for rule induction from data with missing attribute values? In: Yao, J., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 46–55. Springer, Heidelberg (2012)
6. Clark, P.G., Grzymala-Busse, J.W., Kuehnhausen, M.: Local probabilistic approximations for incomplete data. In: Chen, L., Felfernig, A., Liu, J., Raś, Z.W. (eds.) ISMIS 2012. LNCS, vol. 7661, pp. 93–98. Springer, Heidelberg (2012)
7. Clark, P.G., Grzymala-Busse, J.W., Kuehnhausen, M.: Mining incomplete data with many missing attribute values. a comparison of probabilistic and roug set approaches. In: Proceedings of the INTELLI 2013, the Second International Conference on Intelligent Systems and Applications, pp. 12–17 (2013)
8. Grzymala-Busse, J.W.: Rough set strategies to data with missing attribute values. In: Workshop Notes, Foundations and New Directions of Data Mining, in Conjunction with the 3rd International Conference on Data Mining, pp. 56–63 (2003)
9. Grzymala-Busse, J.W.: Data with missing attribute values: Generalization of indiscernibility relation and rule induction. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B., Swiniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 78–95. Springer, Heidelberg (2004)
10. 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)
11. Grzymala-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)
12. Grzymala-Busse, J.W., Marepally, S.R., Yao, Y.: An empirical comparison of rule sets induced by LERS and probabilistic rough classification. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 590–599. Springer, Heidelberg (2010)
13. Grzymala-Busse, J.W., Rząsa, W.: Local and global approximations for incomplete data. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 244–253. Springer, Heidelberg (2006)

14. Grzymala-Busse, J.W., Rząsa, W.: Definability of approximations for a generalization of the indiscernibility relation. In: Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence (IEEE FOCI 2007), pp. 65–72 (2007)
15. Grzymala-Busse, J.W., Rząsa, W.: Local and global approximations for incomplete data. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets VIII. LNCS, vol. 5084, pp. 21–34. Springer, Heidelberg (2008)
16. Grzymala-Busse, J.W., Rząsa, W.: Definability and other properties of approximations for generalized indiscernibility relations. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets XI. LNCS, vol. 5946, pp. 14–39. Springer, Heidelberg (2010)
17. Grzymala-Busse, J.W., Yao, Y.: A comparison of the LERS classification system and rule management in PRSM. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 202–210. Springer, Heidelberg (2008)
18. Grzymala-Busse, J.W., Yao, Y.: Probabilistic rule induction with the LERS data mining system. International Journal of Intelligent Systems 26, 518–539 (2011)
19. 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)
20. Kryszkiewicz, M.: Rough set approach to incomplete information systems. In: Proceedings of the Second Annual Joint Conference on Information Sciences, pp. 194–197 (1995)
21. Kryszkiewicz, M.: Rules in incomplete information systems. Information Sciences 113(3-4), 271–292 (1999)
22. Lin, T.Y.: Neighborhood systems and approximation in database and knowledge base systems. In: Proceedings of the ISMIS 1989, the Fourth International Symposium on Methodologies of Intelligent Systems, pp. 75–86 (1989)
23. Lin, T.Y.: Topological and fuzzy rough sets. In: Slowinski, R. (ed.) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory, pp. 287–304. Kluwer Academic Publishers, Dordrecht (1992)
24. Liu, G., Zhu, W.: Approximations in rough sets vs granluar computing for coverings. International Journal of Cognitive Informatics and Natural Intelligence 4, 63–76 (2010)
25. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
26. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
27. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177, 28–40 (2007)
28. Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough sets: probabilistic versus deterministic approach. International Journal of Man-Machine Studies 29, 81–95 (1988)
29. Ślęzak, D., Ziarko, W.: The investigation of the bayesian rough set model. International Journal of Approximate Reasoning 40, 81–91 (2005)
30. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Transactions on Knowledge and Data Engineering 12, 331–336 (2000)
31. Stefanowski, J.: Algorithms of Decision Rule Induction in Data Mining. Poznan University of Technology Press, Poznan (2001)
32. Stefanowski, J., Tsoukiàs, A.: On the extension of rough sets under incomplete information. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 73–82. Springer, Heidelberg (1999)

33. Stefanowski, J., Tsoukias, A.: Incomplete information tables and rough classification. Computational Intelligence 17(3), 545–566 (2001)
34. Tsumoto, S., Tanaka, H.: PRIMEROSE: probabilistic rule induction method based on rough sets and resampling methods. Computational Intelligence 11, 389–405 (1995)
35. Wang, G.: Extension of rough set under incomplete information systems. In: Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 1098–1103 (2002)
36. 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)
37. Yao, Y.Y.: Two views of the theory of rough sets in finite universes. International Journal of Approximate Reasoning 15, 291–317 (1996)
38. Yao, Y.Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111, 239–259 (1998)
39. Yao, Y.Y.: Probabilistic approaches to rough sets. Expert Systems 20, 287–297 (2003)
40. Yao, Y.Y.: Probabilistic rough set approximations. International Journal of Approximate Reasoning 49, 255–271 (2008)
41. Yao, Y.Y., Lin, T.Y.: Generalization of rough sets using modal logics. Intelligent Automation and Soft Computing 2, 103–120 (1996)
42. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximate concepts. International Journal of Man-Machine Studies 37, 793–809 (1992)
43. Yao, Y.Y., Wong, S.K.M., Lingras, P.: A decision-theoretic rough set model. In: Proceedings of the 5th International Symposium on Methodologies for Intelligent Systems, pp. 388–395 (1990)
44. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46(1), 39–59 (1993)
45. Ziarko, W.: Probabilistic approach to rough sets. International Journal of Approximate Reasoning 49, 272–284 (2008)

Expression and Processing of Uncertain Information

Guoyin Wang[1,2,3], Changlin Xu[1,2], and Hong Yu[2]

[1] School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031, China
[2] Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[3] Institute of Electronic Information Technology, Chongqing Institute of Green & Intelligent Technology, CAS, Chongqing 401122, China
wanggy@ieee.org

Abstract. Uncertainty is one basic feature in the information processing, and the expressing and processing of uncertain information have attracted more attentions. There are many theories introduced to process the uncertain information, such as probability theory, random set, evidence theory, fuzzy set theory, rough set theory, cloud model theory and so on. They depict the uncertain information from different aspects. This paper mainly discusses their differences and relations in expressing and processing for uncertain information. The future development trend is also discussed.

Keywords: uncertain information, probability theory, evidence theory, random set, fuzzy set, rough set, cloud model.

1 Introduction

In the era of increasing popularity of computer and network, the manifestations of information are more diversified with the development of Internet and multimedia technology, such as text, image, video, audio, etc. Human-computer intersection is more frequent and closer. The expression and reasoning of uncertainty as a fundamental feature of information have always been the important issues of knowledge representation and reasoning [13].

There are many kinds of uncertainties, such as randomness, fuzziness, imprecision, incompleteness, inconsistency, etc.. Correspondingly, there are many theoretical models to study uncertain information. For example, the probability theory and the random set theory mainly study the random uncertainty [17][30]; the evidence theory mainly expresses and processes the uncertainties of unascertained information [7][24]; the fuzzy set theory [39] and their derivations, such as the type-2 fuzzy set, the intuitionistic fuzzy set and the interval-valued fuzzy set, study the fuzzy uncertainty of cognition; the rough set theory [19] and its

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 53–65, 2013.

corresponding expansion models discuss the ambiguity indiscernibility and imprecision of information; the cloud model studies the randomness and fuzziness and their relationships [13][14].

Generally speaking, when talking about uncertainty of information, the uncertainty doesn't mean only one kind of uncertainty, but is the coexistence of multi kinds of uncertainty. In this paper, we will discuss the relations among the probability theory, the evidence theory, the random set theory, the fuzzy set theory and its derivations, the rough set theory and its extended models and the cloud model theory.

2 Uncertainty Expression in Probability Theory

Probability, as a measurement of random event, has been already applied widely. Probability and random variable are two important tools during the research of random phenomena. The axiomatic definition of probability is as follows.

Definition 1. [30] *Given a sample space Ω and an associated sigma algebra Σ, For $\forall A \in \Sigma$, the real-valued set function $P(A)$ defined on Σ is called a probability of the event A, when it satisfies:* (1) $0 \leq P(A) \leq 1$*;* (2) $P(\Omega)=1$*;* (3) *If the countable infinite events $A_1, A_2, \cdots \in \Sigma$, $A_i \cap A_j = \emptyset$, $i \neq j$, then* $P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$.

For a given probability space (Ω, Σ, P), random variable X is a real-valued function on sample space Ω. Random variables and their probability distributions are two important concepts of studying stochastic system.

From Definition 1, we know that if the countable infinite events $A_1, A_2, \cdots \in \Sigma$, $A_i \cap A_j = \emptyset$, $i \neq j$, and $\bigcup_{i=1}^{\infty} A_i = \Omega$, then $P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i) = 1$. However, in actual applications, the random events A_i and A_j $(i \neq j)$ may not satisfy strictly $A_i \cap A_j = \emptyset$ due to the uncertainty of random events. So, the countable additivity of probability could not be satisfied. In 1967, Dempster gave a probability which does not satisfy countable additivity, and he tried to use a range of probabilities (upper and lower probabilities) rather than a single probability value to depict the uncertainty so as to establish evidence theory, which is further expansion of probability theory. Random set theory is also another expansion of probability theory, in which the value of a random variable is a closed set rather than a real number. Specific contents will be introduced in section 2.1 and section 2.2 respectively.

2.1 Evidence Theory

In evidence theory, belief function and plausibility function are two most fundamental and important notions. Let Ω be the frame of discernment representing all possible states of a system under consideration. Evidence theory assigns a belief mass to each element of the power set. Formally, the definition of a belief mass function is as follows.

Definition 2. [3] *Let Ω be a frame of discernment, a function $m(A)$: $2^{\Omega} \to [0,1]$, is called a function of basic probability assignment, when it satisfies two properties: $m(\emptyset)=0$ and $\sum_{A \subseteq \Omega} m(A)=1$.*

From Definition 2, we know that the function of basic probability assignment does not satisfy countable additivity due to $\sum_{A \subseteq \Omega} m(A)=1$, so it is different from probability function.

Based on the function of basic probability assignment, the belief function Bel and the plausibility function Pl are defined as:

Definition 3. [3] *Let Ω be a frame of discernment, $\forall A \subseteq \Omega$, a function $Bel : 2^{\Omega} \to [0,1]$, is called a function of belief , when it satisfies: $Bel(X)= \sum_{A \subseteq X} m(A)$. A function $Pl : 2^{\Omega} \to [0,1]$, is called a function of plausibility, when it satisfies: $Pl(X)= \sum_{A \cap X \neq \emptyset} m(A)$.*

From Definition 3, $Bel(A)$ expresses the confident degree of the evidence supporting the event A being true, while $Pl(A)$ expresses the confident degree of the event A being non-false, and $Bel(A) \leq Pl(A)(\forall A \subseteq \Omega)$. $Bel(A)$ and $Pl(A)$ are called the lower limit and the upper limit of confidence degree for A, respectively.

Thus, another difference from the probability theory is that the evidence theory uses a range $[Bel(A), Pl(A)]$ to depict the uncertainty. The interval-span $Pl(A)$-$Bel(A)$ describes the "unknown part" with respect to the event A. Different belief intervals represent different meanings, see Figure 1.

Obviously, the three intervals are relative to the three-way decisions [38]. That is, the support intervals and reject intervals mean the two-way immediate decisions, and the uncertain interval means the third-way decision which also called the deferred decision.

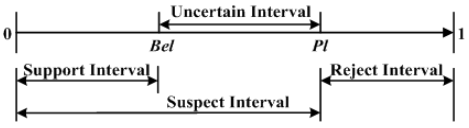


Fig. 1. Uncertainty expression of information

2.2 Random Set Theory

Random set is a set-valued function on sample space Ω, which is a generalization of random variable concept. The strict mathematical definition is as follows.

Definition 4. [17] *Let (Ω, Σ, P) be a probability space, and $(\Psi, \sigma(\beta))$ be a measurable space, where, $\beta \subseteq 2^{\Psi}$, if mapping $F: \Omega \to 2^{\Psi}$, is called random set, when it satisfies: $\forall \Lambda \in \sigma(\beta)$, $\{u \in \Omega | F(u) \in \Lambda\} \in \Sigma$.*

From Definition 4, the difference between random variable and random set is that the former is a random point function, while the latter is a random set-valued function. Thus, random set theory is a generalization from point variable statistics to set variable statistics.

3 Uncertainty Expression in Fuzzy Set Theory

Fuzzy set, which is proposed by Prof. Zadeh as an extension of Cantor set [39], is used to describe the uncertainty of cognition, that is, the extension of concept is not clear and we can not give definitive assessment standard. In Cantor set theory, an element either belongs or does not belong to the set. By contrast, fuzzy set permits the gradual assessment of the membership of elements in a set.

Definition 5. [39] *Let U be a universe of discourse, and A be a fuzzy subset on U, a map μ_A: $U \to [0,1]$, $x \mapsto \mu_A(x)$, is called membership function of A, and $\mu_A(x)$ is called membership degree respect to A.*

From Definition 5, $\mu_A(x)$ expresses the membership degree of an element x belonging to a fuzzy subset A. Once $\mu_A(x)$ is determined, it will be a fixed value. Thus, the operations between fuzzy sets based on membership degree become certainty calculation. Considering the uncertainty of membership degree, Zadeh proposed interval-valued fuzzy set (IVFS) and type-2 fuzzy set (T2FS) as extension of fuzzy set (FS) [40].

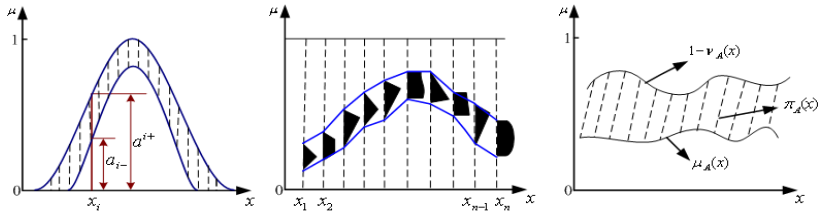
Definition 6. [40] *Let U be a universe of discourse, an interval-valued fuzzy set, denoted A_{IV}, is a map $\mu_{A_{IV}}: U \to Int[0,1]$, where, $Int[0,1]$ expresses a collection of all closed subintervals on $[0,1]$; A type-2 fuzzy set, denoted $\tilde{A}$, is characterized by a type-2 membership function $\mu_{\tilde{A}}(x,u)$, where $\forall x \in U$ and $u \in J_x \subseteq [0,1]$, i.e.: $\tilde{A}=\{((x,u), \mu_{\tilde{A}}(x,u))\}$, or $\tilde{A} = \int_{x \in U} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u)$, where $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$, $\iint$ denotes union over all admissible x and u.*

From Figure 2(a), the membership degree of IVFS A_{IV} is $\mu_{A_{IV}}(x_i)=[a_{i-}, a^{i+}]$. For T2FS, each membership degree $\mu_{\tilde{A}}(x,u)$ is a type-1 membership function $u \in J_x$. Therefore, different x may have different membership function u, see Figure 2(b).

On the other hand, in FS, the membership degree $\mu_A(x)$ is a degree of an element x belonging to a fuzzy subset A, which implies that the non-membership degree of x belonging to A is equal to 1-$\mu_A(x)$. Considering the hesitation degree of an element x belonging to A, Atanassov proposed intuitionistic fuzzy set (IFS) [1], and Gau and Buehrer proposed vague set [9] through membership degree and non-membership degree respectively. Afterward, Bustince and Burillo proved that intuitionistic fuzzy set and vague set are equivalent [4]. The definition of IFS is as follows.

Definition 7. [1] *Let U be a universe of discourse, an intuitionistic fuzzy set A is an object of the form: $A=\{ \langle x, \mu_A(x), \nu_A(x) \rangle | \, x \in U \}$, where μ_A: $U \to [0,1]$ and ν_A: $U \to [0,1]$ are such that $0 \leq \mu_A + \nu_A \leq 1$, and $\mu_A, \nu_A \in [0,1]$ denote degrees of membership and non-membership of x belonging to A, respectively.*

Comparing the FS and IFS, we find that $\mu_A(x)+\nu_A(x)=1$ in FS, while in IFS, $\mu_A(x)+\nu_A(x)\leq 1$. The IFS is shown in Figure 2(c).



(a) Interval-valued fuzzy set (b) Type-2 fuzzy set (c) Intuitionistic fuzzy set

Fig. 2. Uncertainty expression of information

4 Uncertainty Expression in Rough Set Theory

Rough set (RS), proposed by Prof. Pawlak, uses the certain knowledge to depict the uncertain or imprecise knowledge from the perspective of knowledge classification [19], that is, it uses two certain sets (lower approximation set and upper approximation set) to define an uncertain set based on an equivalence relation. The definition of rough set is as follows.

Definition 8. [29] *Let* $K=(U,\mathbf{R})$ *be a knowledge base, the subset* $X\subseteq U$ *and the equivalence relation* $R\in\mathbf{R}$ ($\mathbf{R}$ *is a family of equivalence relation on* U)*, then* $\underline{R}X=\{x\in U|[x]_R\subseteq X\}$, $\overline{R}X=\{x\in U|[x]_R\cap X\neq\emptyset\}$, *are called the* $R-$*lower approximation set and* $R-$*upper approximation set of* X *respectively.* $\mathrm{BN}_R(X)=\overline{R}X-\underline{R}X$ *is called the* $R-$*boundary region of* X*;* $\mathrm{Pos}_R(X)=\underline{R}X$ *is called the positive region of* X*, and* $\mathrm{Neg}_R(X)=U-\overline{R}X$ *is called the negative region of* X*. If* $\mathrm{BN}_R(X)=\emptyset$*, then* X *is definable, otherwise* X *is a rough set.*

A limitation of Pawlak rough set model is that the classification which it deals with must be totally correct or definite. Because the classification is based on the equivalence classes, its results are accurate, that is, "include" or "not include" certainly. To combat the question, some probabilistic rough set (PRS) models are introduced such as the 0.5 probabilistic rough set (0.5-PRS) model [20], the decision-theoretic rough set (DTRS) model [35], the variable precision rough set (VPRS) model [45], the Bayesian rough set (BRS) model [26], the Game-theoretic rough set (GTRS) model [10], and so on.

RS model is based on equivalence relations, and for each object, there is one and only one equivalence class containing it, then this equivalence class can be regarded as the neighborhood of this object, which constitutes the neighborhood system of this object. In general neighborhood system, the object may have two or more neighborhoods. Lin constructed rough set model based on neighborhood system by means of interior point and closure in topology [32]. It is a more generalized approximation set manifestation and also an extension of RS.

In many cases, the information systems are not complete, such as default attribute values. Thus, the rough set theory and method based on incomplete information systems has been extensively studied and developed [12][33].

In short, we can describe the relations among the above models in Figure 3.

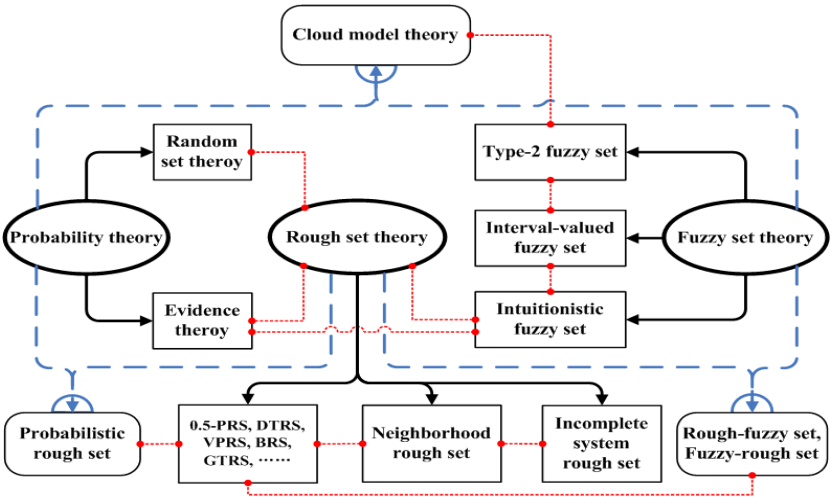


Fig. 3. The relationships between several uncertainty theories

In the foregoing discussion, probability theory, rough set theory and fuzzy set theory are three main uncertainty theories represented with elliptical shape in Figure 3. The random set theory and the evidence theory, IVFS, T2FS, IFS, 0.5-PRS, DTRS, VPRS, BRS, GTRS, the neighborhood rough set and incomplete system rough set are the extended models of probability theory, fuzzy set theory and rough set theory respectively, and they are expressed by rectangle shape. The probabilistic rough set, the rough-fuzzy set, the fuzzy-rough set and the cloud model are obtained by the combination of different theories, and they are expressed by rounded rectangle. The red dotted line expresses the connections among different extended models. The associations and differences between these models will be introduced and analyzed in detail in section 5.

5 Combination, Association and Difference between Different Extended Models

5.1 Probabilistic Rough Set Model

Pawlak RS is based on completion of available information, but the incompletion and statistical information of available information are ignored, so Pawlak RS is often powerless when processing the rule acquisition of inconsistent decision table. Some probabilistic rough set models were introduced to solve problems.

The DTRS was proposed by Yao et al. [35][36], which provides a novel rough set model for studying uncertain information system.

Definition 9. [35] *Let U be a universe of discourse, and R be an equivalence relation on U. A triple $A_p=(U,R,P)$ is a probabilistic approximation space, where a probability measure P defined on sigma algebra of subsets of U. In terms of conditional probability, $\forall X\subseteq U$, the lower and upper probabilistic approximations of X on parameters $\alpha,\beta(0\leq\beta<\alpha\leq 1)$ are: $\underline{\mathrm{P}}_\alpha(X)=\{x\in U|P(X|[x]_R)\ \geq\alpha\}$, $\overline{\mathrm{P}}_\beta(X)=\{x\in U|P(X|[x]_R)>\beta\}$. The corresponding positive region, boundary region and negative region are respectively:* $\mathrm{Pos}(X,\alpha,\beta)=\underline{\mathrm{P}}_\alpha(X)$; $\mathrm{BN}(\mathrm{X},\alpha,\ \beta)=\{x\in\ U|\beta<P(X|[x]_R)<\alpha\}$; $\mathrm{Neg}(X,\alpha,\beta)=\{x\in U|P(X|[x]_R)\leq\beta\}$. *If $\mathrm{BN}(\mathrm{X},\alpha,\beta)\neq\emptyset$, then X is called probabilistic rough set on parameters α,β.*

In this context, each subset of U representing a random event is called a "concept". The conditional probability $P(X|[x]_R)$ can be interpreted as the probability that a randomly selected object with the description of concept $[x]_R$ belongs to X.

5.2 Fuzzy-Rough Set and Rough-Fuzzy Set Models

In the above mentioned various rough set models, the concepts and knowledge are all clear, that is, all sets are classical. However, it is mostly fuzzy concept and fuzzy knowledge that involve in people's actual life. There are two types reflected in rough set model, one is that knowledge of knowledge base is clear while the approximated concept is fuzzy, another is that knowledge of knowledge base and the approximated concept are all fuzzy. Based on this point, Dubois and Prade proposed rough fuzzy sets (RFS) model and fuzzy rough sets (FRS) model based on fuzzy set and rough set [8].

Definition 10. [8] *Let U be a universe of discourse, and R be an equivalence relation on U. If A is a fuzzy set on U, then $\forall x\in U$, $\mu_{\underline{A}_R}(x)=\inf\{\mu_A(y)|y\in[x]_R\}$ and $\mu_{\overline{A}_R}(x)=\sup\{\mu_A(y)|y\in[x]_R\}$ are called the membership functions of lower approximation fuzzy set $\underline{A}_R$ and upper approximation fuzzy set $\overline{A}_R$ respectively. If $\underline{A}_R=\overline{A}_R$, then A is definable, otherwise A is a rough fuzzy set.*

Definition 11. [8] *Let U be a universe of discourse, and $\mathscr{R}$ be a fuzzy equivalence relation on U. If A is a fuzzy set on U, then $\forall y\in U$, $\mu_{\underline{A}_{\mathscr{R}}}(x)=\inf\max\{1-\mu_{[x]_{\mathscr{R}}}(y),\mu_A(y)\}$, $\mu_{\overline{A}_{\mathscr{R}}}(x)=\sup\min\{\mu_{[x]_{\mathscr{R}}}(y),\mu_A(y)\}$ are called the membership functions of lower approximation fuzzy set $\underline{A}_{\mathscr{R}}$ and upper approximation fuzzy set $\overline{A}_{\mathscr{R}}$ respectively. If $\underline{A}_{\mathscr{R}}=\overline{A}_{\mathscr{R}}$, then A is definable, else A is a fuzzy rough set.*

According to Definition 10, if A is a classical set, then $\underline{A}_R$ and $\overline{A}_R$ are two classical sets. The difference between rough set and rough fuzzy set is whether the approximated concept is a classical set or a fuzzy set. Thereupon, the rough fuzzy set is natural generalization of rough set. From Definition 11, we can see that fuzzy rough set is a further expansion of rough fuzzy set due to the equivalence relation R transformed into fuzzy equivalence relation $\mathscr{R}$. In addition, the reference [23] also studied the fuzzy rough set.

5.3 Cloud Model

Cloud model, proposed by Prof. Li, studies the randomness of sample data and membership degree based on probability theory and fuzzy set theory [13], see Figure 3. A formalized definition is as follows.

Definition 12. [13] *Let U be a universal set described by precise numbers, and C be a qualitative concept related to U. If there is a number $x \in U$, which randomly realizes the concept C, and the membership degree μ of x for C is a random number with a stabilization tendency, i.e., $\mu : U \to [0,1], \forall x \in U, x \to \mu(x)$, then the distribution of x on U is defined as a cloud, and each x is a cloud drop.*

From Definition 12, the membership degree $\mu(x)$ of each cloud drop x is a random number, and all the cloud drops satisfy a certain distribution. The density of cloud drops expresses uncertainty degree of a concept C. Generally, a qualitative concept C is expressed by numerical characteristics (Ex, En, He), wherein, Ex is the most expected value of concept; En is used to figure its granularity scale; He is used to depict the uncertainty of concept's granularity. If the distribution of cloud drops is a normal distribution, then the corresponding cloud model is called a normal cloud.

Definition 13. [13] *Let U be a universal set described by precise numbers, and C be a qualitative concept containing three numerical characters (Ex, En, He) related to U. If there is a number $x \in U$, which is a random realization of the concept C and satisfies $x = R_N(Ex, y)$, where $y = R_N(En, He)$, and the certainty degree of x on U is $\mu(x) = \exp\{-\frac{(x-Ex)^2}{2y^2}\}$, then the distribution of x on U is a normal cloud. Where $y = R_N(En, He)$ denoted a normally distributed random number with expectation En and variance He^2.*

From Definition 13, we can depict an uncertain concept concretely. For example, let (Ex=25, En=3, He=0.3) express "Young", where, Ex=25 represents the expected age of "Young", and the corresponding normal cloud map is shown in Figure 4. The generated cloud drops have randomness (horizontal axis), at the same time, for each cloud drop x, the membership degree $\mu(x)$ also has randomness (vertical axis). That is, different people give different ages for "Young", such as 18, 18.5, 19, 20, 22, 28, 30, $\cdots$, namely these ages have stochastic to a certain extent, and each age may have different membership degree of belonging to "Young", take for 22 years example, $\mu(22)$ may equal to 0.3, 0.35, 0.4, 0.47, 0.51, $\cdots$. Thus, cloud model not only considers the randomness of concept, but also involves the randomness of membership degree of object or sample belonging to the concept.

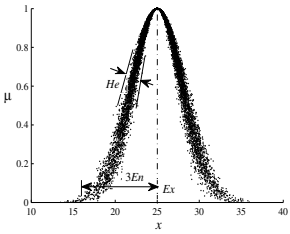


Fig. 4. Normal cloud

5.4 Association and Difference between Different Extended Models

From the above discussion, we know that probability theory, FS theory and RS theory have some extended models respectively, see Figure 3. The associations and differences of these extended models will be discussed.

(1) In FS theory, T2FS, IVFS and IFS are all the generalization of FS based on membership function. The integration of T2FS, IVFS and IFS can obtain some new models, such as interval-valued intuitionistic fuzzy set [2], interval-valued type-2 fuzzy sets [18], type-2 intuitionistic fuzzy set [44], etc. In rough set theory, VPRS and PRS loose the strict definition of approximate boundary. Compared with RS, the positive region and negative region will become larger, while the boundary region will be smaller in VPRS and PRS due to allowing error classification rate to some extent. In this sense, VPRS and PRS have some similar aspects [27]. In addition, the references [15][16][28] studied the variable precision fuzzy rough set and variable precision rough fuzzy set on the basic of VPRS, FRS and RFS, respectively. For the faults of FRS and VPRS, the reference [43] set up a model named fuzzy VPRS by combing FRS and VPRS with the goal of making FRS a special case. The reference [5] studied the vaguely quantified rough set model which is closely related to VPRS. The references [6] and [11] studied the ordered weighted average based FRS and robust FRS model respectively because the classical model of FRS is sensitive to noisy information. The reference [33] studied the rough set model and attribute reduction based on neighborhood system in incomplete system, and the reference [34] proved the VPRS and multi-granulation rough set model [21][22] are the special cases of neighborhood system rough set model and the neighborhood system rough set is a more generalized rough approach. According to the meanings of belief function and plausibility function, they are similarities with the lower and upper approximation of rough set. The references [25][37] discussed the relationship between them. In incomplete information systems, considering all possible values of the object attributes with incomplete information, then the values of some attribute are no longer a single point value but a set value. Based on this, the references [41][42] made the random set introduce into rough set theory and studied the rough set models based on random sets. The reference [41] discussed the relationships between random set, rough set and belief function. The above relations are shown in Figure 3.

(2) The similarities between evident theory, RS and IFS on the representation of uncertain information: The evidence theory depicts the uncertainty of information based on the belief function Bel and plausibility function Pl. IFS uses the membership degree and non-membership degree to study the fuzziness of information which is caused by the extension unclear. RS gives a characterization of uncertain information through lower approximation set $\underline{R}X$ and upper approximation set $\overline{R}X$ based on a equivalence relation R, and uses the roughness $\rho_R(X)=1-|\underline{R}X|/|\overline{R}X|$ to measure the uncertainty. From the aspects of decision-making, people usually perform three kinds of decision-making in our daily life according to the given information [31][38]: determine decision-making including acceptance decision and refusal decision, and delay decision (we can not make the

acceptance or refusal decision based on the current information, and additional information is required to make a decision). The evidence theory, RS and IFS are all able to describe the three decisions. In evidence theory, $Bel(A)$ expresses the degree of acceptance decision, $1-Pl(A)$ expresses the degree of refusal decision, and $Pl(A)-Bel(A)$ describes the degree of delay decision. In RS, the positive region $\mathrm{Pos}_R(X)$ and the negative region $\mathrm{Neg}_R(X)$ can be used to depict the acceptance decision and the refusal decision respectively, and the boundary region $\mathrm{BN}_R(X)$ depicts the delay decision. In IFS, membership function $\mu_A(x)$ and non-membership function $\nu_A(x)$ describe the degrees of acceptance decision and refusal decision respectively, and the hesitation degree $\pi_A(x)=1-(\mu_A(x)+\nu_A(x))$ describes the degree of delay decision. From this point of view, the three theories have common place in expressing of uncertainty information.

(3) The difference between cloud model and T2FS: T2FS discusses the fuzziness of membership degree using the type-1 fuzzy set. Once the membership function is determined, then it will be fixed. While the membership degree of a object belonging to uncertain concept is not a fixed value, but a random number with a stabilization tendency in cloud model. Thus, they have difference. In addition, cloud model considers the randomness of research object. In this sense, cloud model can well integrate the randomness and fuzziness of information.

6 Conclusions and Prospects

The paper summarizes the research on some uncertainty theory models and the corresponding extended models and discusses the associations and differences between them. But there are still some deficiencies, such as the countable additivity of probability may not be satisfied perfectly in practical applications due to uncertainty; how to determine the values of mass function and membership function objectively; the independence of evidence restricts the application range of evidence theory; RS theory does not take into account the randomness of sample data, which makes the generalization ability of acquired knowledge and rules be relatively low and so on. Thus, these problems will be further studied. In addition, because cloud model can deal with randomness and fuzziness, it will be a good issue, worth to study the combination of rough set and cloud model, and the reasoning mechanism, the combination rule of many uncertain concept, the automatically transformed method among multiple granularities based on cloud model are also urgent problems in the future research.

In recent years, computer and network technology advance rapidly. Along with the development of computer network and the widespread application of the Web technology, the data in database is becoming increasingly complicated. Incomplete information, inconsistent information, etc. are also getting more and more general. However, computer can only perform logic and four arithmetic operations essentially. If there are no good models and algorithms, it is still difficult to get the desired results even if there exists highly efficient large-scale computer. Thus, for the problem solving of large-scale complex systems, it needs more methodological innovations. Granular computing, deep learning, quantum coding and so on may be used to reduce system complexity.

Acknowledgments. This work is supported by National Natural Science Foundation of China under grant 61272060, Key Natural Science Foundation of Chongqing under grant CSTC2013jjB40003, and Chongqing Key Laboratory of Computational Intelligence(CQ-LCI-2013-08).

References

1. Atanassov, K.T.: Intuitionistic Fuzzy Sets. Fuzzy Sets and Systems 20, 87–96 (1986)
2. Atanassove, K.T.: Interval Valued Intuitionistic Fuzzy Sets. Fuzzy Sets and Systems 31(3), 343–349 (1989)
3. Baraldi, P., Compare, M., Zio, E.: Maintenance Policy Performance Assessment in Presence of Imprecision Based on Dempster-Shafer Theory of Evidence. Information Sciences 245, 112–131 (2013)
4. Bustince, H., Burillo, P.: Vague Sets are Intuitionistic Fuzzy Sets. Fuzzy Sets and Systems 79, 403–405 (1996)
5. 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)
6. 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)
7. Dempster, A.P.: Upper and Lower Probabilities Induced by a Multivalued Mapping. The Annals of Mathematical Statistics 38(2), 325–339 (1967)
8. Dubois, D., Prade, H.: Rough Fuzzy Sets and Fuzzy Rough Sets. International Journal of General Systems 17, 191–209 (1990)
9. Gau, W.L., Buehrer, D.J.: Vague Sets. IEEE Transaction on Systems Man Cybernetics 23(2), 610–614 (1993)
10. Herbert, J.P., Yao, J.T.: Game-theoretic Risk Analysis in Decision-theoretic Rough Sets. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 132–139. Springer, Heidelberg (2008)
11. 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)
12. Leung, Y., Wu, W.Z., Zhang, W.X.: Knowledge Acquisition in Incomplete Information Systems: A Rough Set Approach. European Journal of Operational Research 168(1), 164–180 (2006)
13. Li, D.Y., Du, Y.: Artificial Intelligence with Uncertainty. Chapman and Hall/CRC, London (2007)
14. Li, D.Y., Liu, C.Y., Gan, W.Y.: A New Cognitive Model: Cloud Model. International Journal of Intelligent Systems 24, 357–375 (2009)
15. Mieszkowicz-Rolka, A., Rolka, L.: Variable Precision Fuzzy Rough Sets. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B., Swiniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 144–160. Springer, Heidelberg (2004)
16. Mieszkowicz-Rolka, A., Rolka, L.: Fuzzy Rough Approximations of Process Data. Internation Journal of Approximate Reasoning 49, 301–315 (2008)
17. Molchanov, I.S.: Theory of Random Sets. Springer, Berlin (2005)

18. Niewiadomski, A.: Interval-valued and Interval Type-2 Fuzzy Sets: A Subjective Comparision. In: 2007 IEEE Int. Conf. on Fuzzy Systems, London, pp. 1–6 (2007)
19. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
20. Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough Sets: Probabilistic Versus Deterministic Approach. Information Sciences 29, 81–95 (1988)
21. Qian, Y.H., Liang, J.Y., Dang, C.Y.: Incomplete Multi-granulations Rough Set. IEEE Transactions on Systems, Man and Cybernetics-Part A 40(2), 420–431 (2010)
22. Qian, Y.H., Liang, J.Y., Yao, Y.Y., et al.: MGRS: A Multi-granulation Rough Set. Information Sciences 180(6), 949–970 (2010)
23. Radzikowska, A.M., Kerre, E.E.: A Comparative Study of Fuzzy Rough Sets. Fuzzy Sets and Systems 126(2), 13–156 (2002)
24. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
25. Skowron, A.: The Relationship Betweem the Rough Set Theory and Evidence Theory. Bulletiin of the Polish Academy of Sciences Mathematics 37, 87–90 (1989)
26. Ślęzak, D.: Rough Sets and Bayes Factor. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 202–229. Springer, Heidelberg (2005)
27. Sun, B.Z., Gong, Z.T.: Variable Precision Probabilistic Rough Set Model. Journal of Northwest Normal University (Natural Science) 41(4), 23–26 (2005)
28. Tsang, E.C.C., Ma, W.M., Sun, B.Z.: Variable Precision Rough Fuzzy Set Model Based on General Relations. In: 2012 IEEE Int. Conf. on Machine Learning and Cybernetics, Xi'an, pp. 195–199 (2012)
29. Wang, G.Y.: Rough Set Theory and Knowledge Acquisition. Xi'an Jiao Tong University Press, Xi'an (2001)
30. Wei, L.L., Ma, J.H., Yan, R.F.: An Introduction to Probability and Statistics. Science Press, Beijing (2012)
31. Wu, W.Z., Leung, Y., Zhang, W.X.: Connections Between Rough Set Theory and Dempster-Shafer Theory of Evidence. International Journal of General Systems 31(4), 405–430 (2002)
32. Yang, X.B., Li, X.Z., Lin, T.Y.: First GrC Model–Neighborhood Systems the Most General Rough Set Models. In: 2009 IEEE Int. Conf. on Granular Computer, Beijing, pp. 691–695 (2009)
33. Yang, X.B., Yang, J.Y.: Incomplete Information System and Rough Set Theory and Attraibute Reduction. Science Press, Beijing (2011)
34. Yang, X.B., Yang, J.Y.: Rough Set Model Based on Neighborhood System. Journal of Nanjing University of Science and Technology 36(2), 291–295 (2012)
35. Yao, Y.Y., Wong, S.K.M., Lingras, P.: A Decision-theoretic Rough Set Model. In: Proceedings of the 5th International Symposium on Methodologies for Intelligent Systems, pp. 17–25. North–Holland, New York (1990)
36. 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)
37. Yao, Y.Y., Lingras, P.J.: Inerpretations of Belief Functions in the Theory of Rough Sets. Information Sciences 104(1-2), 81–106 (1998)
38. Yao, Y.Y.: The Superiority of Three-way Decisions in Probabilistic Rough Set Models. Information Sciences 181(6), 1080–1096 (2011)
39. Zadeh, L.A.: Fuzzy Sets. Information and Control 8(3), 338–353 (1965)
40. Zadeh, L.A.: The Concept of a Linguistic Variable and Its Application to Approximate Reasoning-I. Information Sciences 8, 199–249 (1975)

41. Zhang, W.X., Wu, W.Z.: Rough Set Models Based on Random Sets(I). Journal of Xi'an Jiaotong University 34(2), 75–79 (2000)
42. Zhang, W.X., Wu, W.Z.: Rough Set Models Based on Random Sets(II). Journal of Xi'an Jiaotong University 35(4), 425–429 (2001)
43. 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)
44. Zhao, T., Xiao, J.: Type-2 Intuitionistic Fuzzy Set. Control Theory & Applications 29(9), 1215–1222 (2012)
45. Ziarko, W.: Variable Precision Rough Set Model. Journal of Computer and System Sciences 46(1), 39–59 (1993)

Early Development of Rough Sets - From a Personal Perspective

S.K.M. Wong

Department of Computer Science
University of Regina
Regina Saskatchewan
Canada S4S 0A2

First I would like to thank Dr. Lingras and Dr. Yao for giving me the opportunity to talk to you this morning and get acquainted again with many friends whom I have not seen for quite a while.

I would like to share with you my own personal involvement in the early development of Rough Sets proposed by Professor Pawlak [1,2]. My talk this morning is definitely not meant to be a review of all the important work done in Rough Sets since then. Another thing I want to emphasize is that I am not an expert in this field at all, but it will become clear to you as the story unfolds that somehow my connection with Rough Sets is not broken during these years.

Time goes by very quickly. I still remember quite vividly the day I first met Professor Pawlak almost 30 years ago in Regina. I attended his seminar in which for the first time I heard the term, Rough Sets. Of course, we did not realize then that this term would have had so much impact on us. I am indeed fortunate to be one of the early students learning the concepts of Rough Sets directly from Prof. Pawlak himself. He visited Regina many times. Dr. Ziarko and I also joined him on many occasions when he visited the University of North Carolina at Charlotte. Not only that I learned a lot from Prof. Pawlak, but more importantly we became friends. This period was indeed the heydays of my involvement with the research of Rough Sets.

Dr. Ziarko and I wrote some trivial notes on Rough-Sets at that time. Naturally, we had difficulty to publish them anywhere as one would expect. Who would have heard of such a concept in those days? In order to lessen our frustration, Prof. Pawlak himself recommended those short notes to be published in the Bulletin of the Polish Academy of Sciences [3,4,5,6]. Now looking back at these papers makes me feel quite embarrassed. Nevertheless, I think that they do have some historical value.

We were quite delighted a short time later as we finally succeeded in publishing our first Rough-Sets paper [7] in a real journal. In this paper, we demonstrated that constructing decision rules by inductive learning based on Rough Sets is comparable to (perhaps better than) a popular method of the time proposed by Quinlan [8].

In subsequent years, after we had gained a better understanding of Rough Sets, we began to search for appropriate applications of this new concept. Our main objective was to demonstrate the usefulness of Rough Sets in decision making, learning and classification problems. Soon we realized that it might be

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 66–74, 2013.

necessary to incorporate some sort of probabilistic or numeric measure into the algebraic structure of the Rough-Sets model.

Before we do that, let us re-visit the original ideas suggested by Pawlak [1]. Each rectangle in Figure 1 represents an equivalence class of the equivalence relation (partition) induced by a set of attributes in the knowledge representation system [1,2]. The objective is to approximate a concept set by these equivalence classes. (Interestingly, such a procedure is very much analogous to the one used for estimating the area of a two-dimensional figure drawn on a graph paper as shown in Figure 2. In this case, we assume that the area of each square on the graph paper is known. Similar to Rough Sets, we can define the lower and upper bounds of the actual area of the figure of interest. Obviously, the accuracy of such an approximation depends on the size of the individual squares (i.e., the level of our knowledge). This is perhaps a good example in showing the usefulness of granular computations.

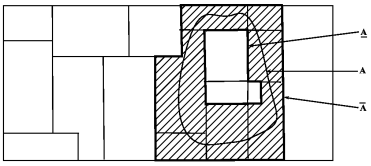


Fig. 1. Approximate classification of set A in the Rough-Sets structure (S, P)

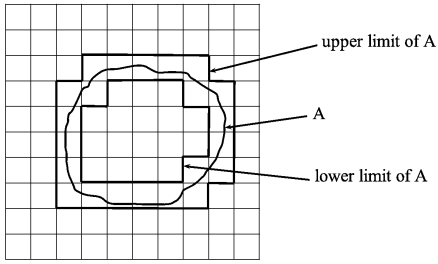


Fig. 2. Approximate computation of the area of an irregular figure A - an illustration of the concept of integration in elementary Calculus

Although we are dealing with similar approximations in the two examples shown in Figures 1 and 2, there are, however, major differences in their physical interpretations. In the Rough-Sets model, a partition represents one's knowledge (incomplete knowledge). The concept represented by the irregular figure in Figure 1 is what we want to learn (or to classify). On the other hand, Figure 2 demonstrates the concept of integration as a limit of summation in Calculus. That is, the area of the figure will approach the true area (the limit) when the size of the individual square approaches zero.

In practical applications, to treat each equivalence class in the boundary region equally may not be too satisfactory. For example, if certain equivalence classes in the boundary region (in Figure 1) are ignored, then one obvious drawback is that some relevant (useful) information may be lost. Conversely, extraneous information (noise) may be retained inadvertently. To alleviate these problems, we suggested a simple way [9] to incorporate a probabilistic measure

into the standard Rough-Sets model. The probabilistic lower and upper approximations of a concept are defined as:

$$A_p = \bigcup_{p(A/X_i)>1/2} X_i, \tag{1}$$

$$\bar{A}_p = \bigcup_{p(A/X_i)\geq 1/2} X_i, \tag{2}$$

where A (a subset of objects) represents a concept, $[X_i]$ denotes the partition of objects in a knowledge system, and $p(A/X_i)$ is the probability of A conditioned on $[X_i]$

We also introduced the notions of statistical Reduct and Core [9]. (Note that we obtained some new results on attribute reduction [11].)

It is perhaps worth mentioning here that we suggested a decision theoretic framework for approximating concepts [10]. In this paper, we explored the implications of approximating a concept based on the Bayesian decision procedure. We showed that if a given concept is approximated by two sets, we can derive both the algebraic and probabilistic Rough-Sets approximations from our approach [10].

In this period, a lot of activities happened as people became more aware of the potential applications of Rough Sets. On the theoretical front, there were two interesting and related developments, namely, Belief Functions and Modal Logic. Actually, at the time, the study of Belief Functions proposed by Shafer [12] was a hot but controversial research topic. Actually, some of the controversies remain today.

As probability theory has long been accepted as the standard numerical measure of uncertainty, many questions were raised about Belief Functions. What is a Belief Function for? How is it related to the Probabilistic Function? From our vantage point, since rough approximation is also a measure of uncertainty resulting from insufficient knowledge, a natural question was: is the notion of Rough Sets in any way related to the Probability or Belief Functions? It turned out that they all share a common algebraic structure. It is perhaps worthwhile to give you (especially to those younger students) some insight of this intriguing relationship among Rough Sets, Belief and Probability Functions.

For this purpose, let us first introduce Belief Functions in a form from which one will see right away that Belief Functions and Rough-Sets actually share the same algebraic structure. It will also become clear that the underlying structure of Probability Functions is a special case of Belief-Functions.

Let S denote a set of *possibilities* (states, possible worlds, objects). A Belief Function, $Bel : 2^S \rightarrow [0, 1]$, over S is defined as follows, for any $A \subseteq S$,

$$Bel(A) = \sum_{f \subseteq A} m(f), \tag{3}$$

where $F = \{f\}$ is a set of *focal* elements satisfying the conditions $f \subseteq S$ and $\bigcup_{f \epsilon F} f = S$, and m is a *probability assignment* defined by:

a) $m(\emptyset) = 0$,
b) $\sum_{f \epsilon F} m(f) = 1, \ \ m(f) > 0$.

Note that in general the focal elements in F are not necessarily *disjoint*.

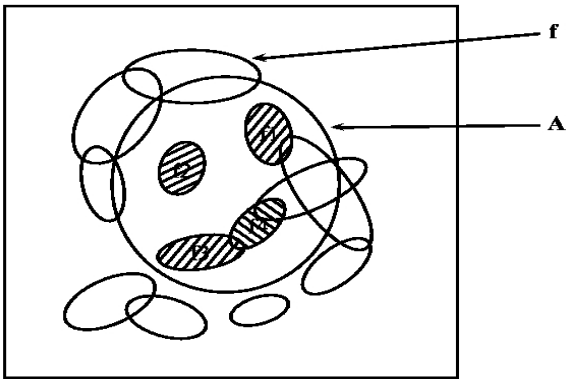


Fig. 3. Belief-Functions Structure (S, F) defined by a *Focal Set* $F = \{f\}$. f_1, f_2, f_3, $f_4 \subseteq A$, and $Bel(A) = m(f_1) + m(f_2) + m(f_3) + m(f_4)$.

From Figure 3, one can conclude immediately that if all the focal elements of the focal set are pairwise disjoint, the Rough-Sets and the Belief-Functions Structures are indeed the same as we mentioned above (compare Figure 1 and Figure 3).

Furthermore, the Belief-Functions Structure reduces to the Probability-Functions Structure if every focal element is a singleton set. Note that in the standard probability theory, no uncertainty is involved in characterizing (describing) the concepts themselves as it is based on the Proposition Logic. (In contrast, the theory of Rough Sets is based on the Modal Logic. We will discuss this important difference in greater details later.) In Rough-Sets terms, we say that in the probabilistic case, both the lower and upper approximations of any concept are the same. It is assumed in this theory that one has sufficient knowledge to represent (define) any concept unambiguously by a set of states. This means that given a state, there are only two possibilities, namely, either the state (object) belongs to a given concept set or it does not. Uncertainty is reflected only in the numerical ordering of the concepts. (The numeric ordering is based on probabilities assigned to the individual concepts).

I want to point out that the pairwise disjoint assumption in the Rough-Sets model and in many other approaches is inherent in the knowledge system (see the example in Table 1 and Figure 4) in which knowledge is represented by an equivalence relation inferred from the values of the attributes [1,2] or from the truth values of the primitive propositions [19].

I want to mention one more aspect about Belief Functions before introducing a logical knowledge system below.

Table 1. A Rough-Sets Knowledge System

Objects	Color	Classification
s_1	red	+
s_2	red	+
s_3	green	+
s_4	green	-
s_5	blue	-
s_6	yellow	-
s_7	yellow	-
s_8	yellow	-

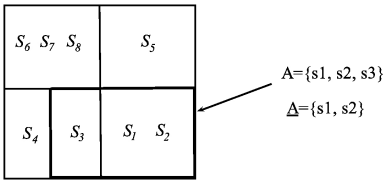


Fig. 4. Partition $P = [\{s_1, s_2\},\ \{s_3, s_4\}, \{s_5\},\ \{s_6, s_7, s_8\}]$ induced by the attribute Colour

Belief Functions provide numeric orderings of subsets of possibilities (possible worlds). The numeric ordering of a Belief Function induces a specific type of (qualitative) ordering relation (referred to as the belief relation) on subsets of the possible worlds. As in the standard probability theory [13], an important question arose was: whether there is a finite set of axioms that must be satisfied by the ordering relation such that there exists a Belief Function consistent with such a qualitative ordering? The answer to this question will become crucial if either qualitative or numeric ordering of concepts is required for making decisions (choices) in some applications [14]. Our main result is summarized by the following theorem [15].

Theorem 1. *Let S be a set of possibilities (possible worlds) and $\succ$ a preference relation defined on 2^S. There exists a Belief Function, $Bel : 2^S \to [0, 1]$, satisfying for $A, B \in 2^S$,*

$$A \succ B \Longleftrightarrow Bel(A) > Bel(B) \tag{4}$$

if and only if the preference relation $\succ$ satisfies the following axioms:

- *(B1) (asymmetry) $A \succ B \Rightarrow \neg(B \succ A)$,*
- *(B2) (negative transitivity) $(\neg(A \succ B),\ \neg(B \succ C)) \Rightarrow \neg(A \succ C)$,*
- *(B3) (dominance) $A \subseteq B \Rightarrow \neg\ (B \succ A)$,*
- *(B4) (partial monotonicity) $(A \supset B, A \cap C = \emptyset) \Rightarrow (A \succ B \Rightarrow A \cup C \succ B \cup C)$,*
- *(B5) (nontriviality) $S \succ \emptyset$.*

Another important development at this time was the study [16,17,18,19] of the relationship between Rough Sets and Logic as I mentioned above. Indeed, there exists a fundamental relationship among the three knowledge systems, Belief Functions, Rough Sets and Modal Logic.

More recently, we suggested a unified *structure* (referred to as the Belief Structure [21]) for both Belief Functions and Rough Sets within the framework of Modal Logic [20]. An example of a Belief Structure (for n agents) is shown in Figure 5. Note that the Kripke Structure in modal logic is in fact a special case of the Belief Structure when knowledge is represented by equivalence relations. The basic ideas are outlined as follows.

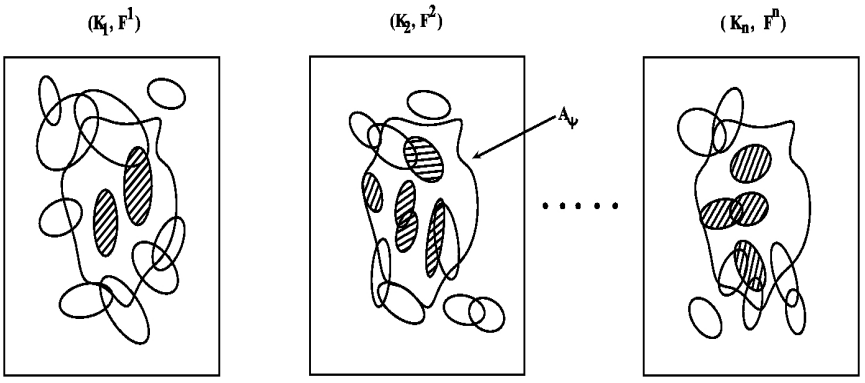


Fig. 5. Belief Structure for n agents. K_i is the knowledge operator, F^i is the focal set of agent i and $A^i_\varphi = \{s \mid (B, s) \models K_i\varphi\}$ is represented by the shaded area.

We first describe a language. Let Φ denote a set of primitive propositions. These propositions stand for basic facts about the individual *possible worlds* (states, objects, possibilities) in a universe S. Suppose there are n agents. We augment the language by modal operators $K_1, K_2, ..., K_n$(one for each agent). We start with the primitive propositions in Φ and form more complicated formulas by closing off under negation($\neg$), conjunction($\wedge$) and the modal operators $K_1, K_2, ..., K_n$.

We have just described the syntax of the language. Next we need to define the semantics which will enable us to determine if a formula is *true* or *false* in a given *structure*. Here we assume that the knowledge of the agent $i \leq n$ is represented by a *focal set* F^i(a family of subsets of S),

$$F^i = \{f^i_1, f^i_2, ..., f^i_l\}, \text{ where } f^i \subseteq S, \bigcup_{f \epsilon F^i} f = S. \tag{5}$$

Let us now define the *Belief Structure* B for n agents over the set of primitive propositions $\Phi = \{p\}$ as a tuple:

$$B = (S, \pi, F^1, F^2, ..., F^n), \tag{6}$$

where $\pi(s)$ is a truth assignment to each $s \epsilon S$,

$$\pi(s) : \Phi \rightarrow \{true, false\}. \tag{7}$$

To denote that a primitive proposition $p \epsilon \Phi$ is true in the possible world $s \epsilon S$ with respect to the Belief Structure B, we write:

$$(B, s) \models p \text{ iff } \pi(s)(p) = true. \tag{8}$$

This is the base case. By induction, the truth value of a general formula is defined by:

$$(B, s) \models \varphi \wedge \phi \text{ iff } (B, s) \models \varphi \text{ and } (B, s) \models \phi, \tag{9}$$

$$(B, s) \models \neg\varphi \text{ iff } (B, s) \not\models \varphi. \tag{10}$$

Finally, we define the meaning of the formula $K_i\varphi$. We say that the formula $K_i\varphi$ is true in s if φ is true at all the possible worlds in *some* focal element $f \epsilon F^i$ containing s. Formally,

$$(B, s) \models K_i\varphi \text{ iff } \exists f \epsilon F^i \text{ such that } s \epsilon f \text{ and } f \subseteq \{s' \mid (B, s') \models \varphi\}. \tag{11}$$

In the above analysis one may interpret the formula φ as a subset of possible worlds (objects), namely,

$$\boldsymbol{\varphi} = \{s \mid (B, s) \models \varphi\}. \tag{12}$$

Similarly we have

$$\boldsymbol{K_i\varphi} = \{s \mid (B, s) \models K_i\varphi\}. \tag{13}$$

One can immediately conclude that $\boldsymbol{K_i\varphi}$ is the lower approximation of the concept $\boldsymbol{\varphi}$ in the Rough-Sets model.

Thus we have established a unified framework for the Rough Sets, Belief Functions and Kripke Structures. Our approach augmented by the power of modal logic will broaden the applications of the Rough-Sets model to include reasoning about knowledge[18].

Let me use a simple example to demonstrate reasoning in the logical approach. Table 2 depicts a knowledge system of agents a, b and c, in which each state (possible world) s in $S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$ is described by the truth values of the primitive propositions p_a, p_b and p_c:

p_a (agent a has mud on forehead),
p_b (agent b has mud on forehead),
p_c (agent c has mud on forehead).

Table 2. A "Logical" Knowledge System

S	p_a	p_b	p_c
s_1	0	0	0
s_2	0	0	1
s_3	0	1	0
s_4	0	1	1
s_5	1	0	0
s_6	1	0	1
s_7	1	1	0
s_8	1	1	1

The knowledge relations of agents a, b and c are:

$$K_a = \{(s_1, s_5), (s_2, s_6), (s_3, s_7), (s_4, s_8)\},$$
$$K_b = \{(s_1, s_3), (s_2, s_4), (s_5, s_7), (s_6, s_8)\},$$
$$K_c = \{(s_1, s_2), (s_3, s_4), (s_5, s_6), (s_7, s_8)\}.$$

We can compute

$A_{p_a} = \{s \mid (B, s) \models p_a\} = \{s_5, s_6, s_7, s_8\}$ (agent a has mud on forehead),
$A^a_{p_a} = \{s \mid (B, s) \models K_a p_a\} = \emptyset$ (agent a does not know if he has mud on forehead),
$A^b_{p_a} = \{s \mid (B, s) \models K_b p_a\} = \{s_5, s_6, s_7, s_8\}$ (agent b knows agent a has mud on forehead),
$A^c_{p_a} = \{s \mid (B, s) \models K_c p_a\} = \{s_5, s_6, s_7, s_8\}$ (agent c knows agent a has mud on forehead).

Note that $A^a_{p_a}, A^b_{p_a}$ and $A^c_{p_a}$ are in fact the lower approximations of the concept A_{P_a} with respect to different partitions K_a, K_b and K_c.

I would like to conclude our discussion here. It has given me a lot of satisfaction to see so many young and bright researchers in this meeting. I am sure that Rough Sets are in good hands in the years to come. Thank you.

Acknowledgement. I would like to give special thanks to Dr. Dan Wu for his generous assistance and suggestions in preparing this paper.

References

[1] Pawlak, Z.: Rough Sets. International Journal of Information and Computer Sciences 11, 341–356 (1982)
[2] Pawlak, Z.: On Superfluous Attributes in Knowledge Representation. Bulletin of Polish Academy of Sciences, Technical Sciences 32, 211–213 (1984)
[3] Wong, S.K.M., Ziarko, W.: On Reducing the Complexity of the Selection Problem. Bulletin of the Polish Academy of Sciences, Mathematics 33(11-12), 697–700 (1985)
[4] Wong, S.K.M., Ziarko, W.: On Optimal Decision Rules in Decision Tables. Bulletin of Polish Academy of Sciences, Mathematics 33(11-12), 693–696 (1985)
[5] Wong, S.K.M., Ziarko, W.: Algorithm for Inductive Learning. Bulletin of Polish Academy of Sciences, Technical Sciences 34(5-6), 272–276 (1986)

[6] Wong, S.K.M., Ziarko, W., Ye, R.L.: Remarks on Attribute Selection Criterion in Inductive Learning Based on Rough Sets. Bulletin of Polish Academy of Sciences, Technical Sciences 34(5-6), 277–283 (1986)
[7] Wong, S.K.M., Ziarko, W., Ye, R.L.: Comparison of Rough-Set and Statistical Methods in Inductive Learning. Int. J. Man - Machine Studies 24, 53–72 (1986)
[8] Quinlan, J.R.: Learning Efficient Classification Procedures and their Application to Chess and Games. In: Michalski, R.S., Carbonnell, J.G., Mitchell, T.M. (eds.) Machine Learning: the Artificial Intelligence Approach. Tioga Press, Palo Alto (1983)
[9] Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough Sets: Probabilistic versus Deterministic Approach. Int. J. Man - Machine Studies 29, 81–89 (1986)
[10] Yao, Y.Y., Wong, S.K.M.: A Decision Theoretic Framework for Approximating Concepts. Int. J. Man - Machine Studies 37, 793–809 (1992)
[11] Zhao, Y., Wong, S.K.M., Yao, Y.: A Note on Attribute Reduction in the Decision-Theoretic Rough Set Model. In: Peters, J.F., Skowron, A., Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) Transactions on Rough Sets XIII. LNCS, vol. 6499, pp. 260–275. Springer, Heidelberg (2011)
[12] Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
[13] Scott, D.: Measurement Models and Linear Inequalities. Journal of Mathematical Psychology 1, 233–247 (1964)
[14] Wong, S.K.M., Lingras, P.: Representation of Qualitative User Preference by Quantitative Belief Functions. IEEE Transactions on Knowledge and Engineering 6(1), 72–78 (1994)
[15] Wong, S.K.M., Yao, Y.Y., Bollmann, P., Burger, H.C.: Axiomatization of Qualitative Belief Structure. IEEE Transactions on Systems, Man and Cybernetics 21(4), 726–734 (1991)
[16] Orlowska, E.: Logic of Indiscernibility Relations. In: Skowron, A. (ed.) SCT 1984. LNCS, vol. 208, pp. 177–186. Springer, Heidelberg (1985)
[17] Yao, Y.Y., Lin, T.Y.: Generalization of Rough Sets Using Model Logic. Intelligent Automation and Soft Computing 2, 103–120 (1996)
[18] Yao, Y.Y., Wong, S.K.M., Lin, T.L.: A Review of Rough Set Model. In: Lin, T.Y., Cercone, N. (eds.) Rough Sets and Data Mining Analysis of Imprecise Data, pp. 47–75. Kluwer Academic Publishers (1997)
[19] Wong, S.K.M.: A Rough-Set Model for Reasoning about Knowledge. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 276–285. Physica-Verlag, Springer (1998)
[20] Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge. MIT Press, Cambridge (1996)
[21] Wong, S.K.M., Noroozi, N.: A Belief Structure for Reasoning about Knowledge. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS, vol. 6401, pp. 288–297. Springer, Heidelberg (2010)

Contraction to Matroidal Structure of Rough Sets

Jingqian Wang and William Zhu

Lab of Granular Computing,
Minnan Normal University, Zhangzhou 363000, China
williamfengzhu@gmail.com

Abstract. As an important technique for granular computing, rough sets deal with vagueness and granularity in information systems. Rough sets are usually used in attribute reduction, however, the corresponding algorithms are often greedy ones. Matroids generalize the linear independence in vector spaces and provide well-established platforms for greedy algorithms. In this paper, we apply contraction to a matroidal structure of rough sets. Firstly, for an equivalence relation on a universe, a matroid is established through the lower approximation operator. Secondly, three characteristics of the dual of the matroid, which are useful for applying a new operation to the dual matroid, are investigated. Finally, the operation named contraction is applied to the dual matroid. We study some relationships between the contractions of the dual matroid to two subsets, which are the complement of a single point set and the complement of the equivalence class of this point. Moreover, these relationships are extended to general cases. In a word, these results show an interesting view to investigate the combination between rough sets and matroids.

Keywords: Approximation operator, Contraction, Matroid, Rough set.

1 Introduction

Rough set theory was proposed by Pawlak [16,17] in 1981 as a tool to conceptualize, organize and analyze various types of data in data mining. Rough set theory has been widely used to deal with many practical problems, such as attribute reduction [7,15], feature selection [1,6], rule extraction [2,4], and knowledge discovery [11,24]. Moreover, through extending equivalence relations or partitions, rough set theory has been extended to generalized rough sets based on relations [8,18] and covering-based rough sets [21,25].

Matroid theory [9,12] is a generalization of linear algebra and graph theory. It has been used in diverse fields, such as combinatorial optimization [10], algorithm design [5], information coding [19], cryptology [3], and so on. Recently, matroid theory has been connected with other theories, such as rough set theory [20,26,27] and lattice theory [13,14].

In this paper, we apply contraction to a matroidal structure of rough sets. Firstly, for an equivalence relation on a universe, a matroid is induced by the lower approximation operator through independent set axiom of matroids. Secondly, three characteristics of the dual of the matroid, which are independent sets, bases and the rank function, are investigated. These characteristics are useful for the following study. Finally, an

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 75–86, 2013.

operation named contraction is introduced, and the contraction of the dual matroid to a subset is a new matroid. The contractions of the dual matroid to two subsets, which are the complement of a single point set and the complement of the equivalence class of this point, are obtained. That is to say, two matroids are obtained by applying contraction to the dual matroid. We study some relationships between characteristics of these two matroids, such as they have the same independent sets. Moreover, these relationships are extended to general cases. The relationships between contractions of the dual matroid to another two subsets, which are the complement of a subset and the complement of the upper approximation of this subset, are investigated.

The rest of this paper is organized as follows. Section 2 reviews some fundamental definitions about rough sets and matroids. In Section 3, a matroid is induced by the lower approximation operator in rough sets, and the dual of the matroid is investigated. In Section 4, we study some relationships between the contractions of the dual matroid to two subsets, which are the complement of a single point set and the complement of the equivalence class of this point. Then these relationships are extended to general cases. Finally, Section 5 concludes this paper and indicates further works.

2 Basic Definitions

This section recalls some fundamental definitions related to Pawlak's rough sets and matroids.

2.1 Pawlak's Rough Sets

The following definition shows that a universe together with an equivalence relation on it forms an approximation space.

Definition 1. *(Approximation space [22,23]) Let U be a nonempty and finite set called universe and R an equivalence relation on U. The ordered pair (U, R) is called a Pawlak's approximation space.*

In rough sets, a pair of approximation operators are used to describe an object. In the following definition, we introduce the pair of approximation operators.

Definition 2. *(Approximation operator [22,23]) Let R be an equivalence relation on U. A pair of approximation operators R_*, $R^* : 2^U \to 2^U$, are defined as follows: for all $X \subseteq U$,*

$$R_*(X) = \{x \in U : RN(x) \subseteq X\}, \text{ and}$$
$$R^*(X) = \{x \in U : RN(x) \bigcap X \neq \emptyset\},$$

where $RN(x) = \{y \in U : xRy\}$. They are called the lower and upper approximation operators with respect to R, respectively.

In the above definition, we call $RN(x)$ the equivalence class of x. Let $\emptyset$ be the empty set and $-X$ the complement of X in U. According to the definition of approximation operators, we have the following conclusions.

Proposition 1. *([22,23]) The properties of the Pawlak's rough sets are:*

(1L) $R_*(U) = U$	*(1H)* $R^*(U) = U$
(2L) $R_*(\phi) = \phi$	*(2H)* $R^*(\phi) = \phi$
(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) $X \subseteq Y \Rightarrow R_*(X) \subseteq R_*(Y)$	*(6H)* $X \subseteq Y \Rightarrow R^*(X) \subseteq R^*(Y)$
(7L) $R_*(-R_*(X)) = -R_*(X)$	*(7H)* $R^*(-R^*(X)) = -R^*(X)$

(8LH) $R_*(-X) = -R^*(X)$
(9LH) $R_*(X) \subseteq R^*(X)$

2.2 Matroids

Matroids generalize the linear independency in linear algebra and the cycle in graph theory. In the following definition, one of the most valuable definitions of matroids is presented from the viewpoint of independent sets.

Definition 3. *(Matroid [9]) A matroid is an ordered pair* $M = (U, \boldsymbol{I})$ *where* U *(the ground set) is a finite set, and* $\boldsymbol{I}$ *(the independent sets) is a family of subsets of* U *with the following properties:*
(I1) $\emptyset \in \boldsymbol{I}$*;*
(I2) If $I \in \boldsymbol{I}$*, and* $I' \subseteq I$*, then* $I' \in \boldsymbol{I}$*;*
(I3) If $I_1, I_2 \in \boldsymbol{I}$*, and* $|I_1| < |I_2|$*, then there exists* $e \in I_2 - I_1$ *such that* $I_1 \bigcup \{e\} \in \boldsymbol{I}$*, where* $|I|$ *denotes the cardinality of* I*.*

In order to show that linear algebra is an original source of matroid theory, we present an example from the viewpoint of the linear independence in vector spaces.

Example 1. Let $U = \{a_1, a_2, a_3\}$ where $a_1 = [1 \ \ 0]^T$, $a_2 = [0 \ \ 1]^T$, $a_3 = [1 \ \ 1]^T$, where a^T is the transpose of a. Denote $\mathbf{I} = \{X \subseteq U : X$ are linearly independent $\}$, i.e., $\mathbf{I} = \{\emptyset, \{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_3\}\}$. Then $M = (U, \mathbf{I})$ is a matroid.

In order to illustrate that graph theory is another original source of matroid theory, an example is presented from the viewpoint of the cycle of a graph.

Example 2. Let $G = (V, E)$ be the graph as shown in Fig. 1. Denote $\mathbf{I} = \{X \subseteq E : X$ does not contain a cycle of $G\}$, i.e., $\mathbf{I} = \{\emptyset, \{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_1, e_2\}, \{e_1, e_3\}, \{e_1, e_4\}, \{e_2, e_3\}, \{e_3, e_4\}, \{e_1, e_2, e_3\}, \{e_1, e_3, e_4\}\}$. Then $M = (E, \mathbf{I})$ is a matroid, where $E = \{e_1, e_2, e_3, e_4\}$.

If a subset of the ground set is not an independent set of a matroid, then it is called a dependent set of the matroid. Based on the dependent set, we introduce the circuit of a matroid. For this purpose, two denotations are presented.

Definition 4. *([9]) Let* $\boldsymbol{A} \subseteq 2^U$ *be a family of subsets of* U*. One can denote:*
$Max(\boldsymbol{A}) = \{X \in \boldsymbol{A} : \forall Y \in \boldsymbol{A}, X \subseteq Y \Rightarrow X = Y\}$*;*
$Min(\boldsymbol{A}) = \{X \in \boldsymbol{A} : \forall Y \in \boldsymbol{A}, Y \subseteq X \Rightarrow X = Y\}$.

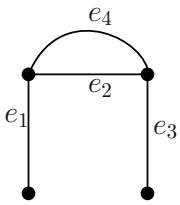


Fig. 1. A graph

The dependent set of a matroid generalizes the linear dependence in vector spaces and the cycle in graphs. Any circuit of a matroid is a minimal dependent set.

Definition 5. *(Circuit [9]) Let $M = (U, \boldsymbol{I})$ be a matroid. A minimal dependent set in M is called a circuit of M, and we denote the family of all circuits of M by $\boldsymbol{C}(M)$, i.e., $\boldsymbol{C}(M) = Min(-\boldsymbol{I})$, where $-\boldsymbol{I} = 2^U - \boldsymbol{I}$.*

In fact, e_2 and e_4 form a cycle of the graph as shown in Fig. 1. Therefore, $\mathbf{C}(M) = \{\{e_2, e_4\}\}$ in Eaxmple 2. A base of a matroid is a maximal independent set.

Definition 6. *(Base [9]) Let $M = (U, \boldsymbol{I})$ be a matroid. A maximal independent set in M is called a base of M, and we denote the family of all bases of M by $\boldsymbol{B}(M)$, i.e., $\boldsymbol{B}(M) = Max(\boldsymbol{I})$.*

The dimension of a vector space and the rank of a matrix are useful concepts in linear algebra. The rank function of a matroid is a generalization of these two concepts.

Definition 7. *(Rank function [9]) Let $M = (U, \boldsymbol{I})$ be a matroid. The rank function r_M of M is defined as $r_M(X) = max\{|I| : I \subseteq X, I \in \boldsymbol{I}\}$ for all $X \subseteq U$. $r_M(X)$ is called the rank of X in M.*

Given a matroid, we can generate a new matroid through the following proposition.

Proposition 2. *([9]) Let $M = (U, \boldsymbol{I})$ be a matroid and $\boldsymbol{B}^* = \{U - B : B \in \boldsymbol{B}(M)\}$. Then $\boldsymbol{B}^*$ is the family of bases of a matroid on U.*

The new matroid in the above proposition, whose ground set is U and whose family of bases is $\mathbf{B}^*$, is called the dual of M and denoted by M^*.

3 Matroidal Structure of Rough Sets

In this section, we establish a matroidal structure of rough sets through approximation operators of rough sets.

3.1 Matroid Induced by Lower Approximation Operator

This subsection induces a matroid through the lower approximation operator for any Pawlak's approximation space. In the following proposition, for an equivalence relation on a universe, a family of subsets of the universe induced by the lower approximation operator satisfies the independent set axiom of matroids.

Proposition 3. *Let R be an equivalence relation on U. Then $\boldsymbol{I}(R) = \{X \subseteq U : R_*(X) = \emptyset\}$ satisfies (I1), (I2) and (I3) of Definition 3.*

Proof. (I1): According to Proposition 1, $R_*(\emptyset) = \emptyset$. Then $\emptyset \in \mathbf{I}(R)$.
(I2): Let $I \in \mathbf{I}(R)$, $I' \subseteq I$. Since $I \in \mathbf{I}(R)$, so $R_*(I) = \emptyset$. According to Proposition 1, $R_*(I') \subseteq R_*(I) = \emptyset$. Therefore, $R_*(I') = \emptyset$, i.e., $I' \in \mathbf{I}(R)$.
(I3): Let the partition generated by R on U be $U/R = \{P_1, P_2, \cdots, P_m\}$. Let $I_1, I_2 \in \mathbf{I}(R)$ and $|I_1| < |I_2|$. Since $I_1 = I_1 \bigcap U$ and $I_2 = I_2 \bigcap U$, so $I_1 = I_1 \bigcap (\bigcup_{i=1}^m P_i) = \bigcup_{i=1}^m (I_1 \bigcap P_i)$ and $I_2 = I_2 \bigcap (\bigcup_{i=1}^m P_i) = \bigcup_{i=1}^m (I_2 \bigcap P_i)$. Since $I_1, I_2 \in \mathbf{I}(R)$, so $R_*(I_1) = \emptyset$ and $R_*(I_2) = \emptyset$. Thus, for all $1 \leq i \leq m$, $(I_1 \bigcap P_i) \subset P_i$ and $(I_2 \bigcap P_i) \subset P_i$. Since $|I_1| < |I_2|$, so $|\bigcup_{i=1}^m (I_1 \bigcap P_i)| < |\bigcup_{i=1}^m (I_2 \bigcap P_i)|$, i.e., $\sum_{i=1}^m |I_1 \bigcap P_i| < \sum_{i=1}^m |I_2 \bigcap P_i|$. Therefore, there exists $1 \leq i \leq m$ such that $|I_1 \bigcap P_i| < |I_2 \bigcap P_i|$ (In fact, if for all $1 \leq i \leq m$ such that $|I_1 \bigcap P_i| \geq |I_2 \bigcap P_i|$, then $\sum_{i=1}^m |I_1 \bigcap P_i| \geq \sum_{i=1}^m |I_2 \bigcap P_i|$, i.e., $|I_1| \geq |I_2|$. It is contradictory with $|I_1| < |I_2|$). Thus, $|I_1 \bigcap P_i| < |I_2 \bigcap P_i| < |P_i|$, and there exists $e \in (I_2 \bigcap P_i) - (I_1 \bigcap P_i) \subseteq I_2 - I_1$ such that $(I_1 \bigcap P_i) \bigcup \{e\} \subset P_i$, i.e., $R_*(I_1 \bigcup \{e\}) = \emptyset$. Hence $I_1 \bigcup \{e\} \in \mathbf{I}(R)$. This completes the proof.

Therefore, there exists a matroid on U such that $\mathbf{I}(R)$ is the family of its independent sets, and the matroid is denoted by $M(R) = (U, \mathbf{I}(R))$.

Example 3. Let $U = \{a, b, c, d, e\}$, R be an equivalence relation on U and the partition generated by R on U be $U/R = \{\{a, b\}, \{c, d, e\}\}$. According to Proposition 3, $\mathbf{I}(R) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, d\}, \{c, e\}, \{d, e\}, \{a, c, d\}, \{a, c, e\}, \{a, d, e\}, \{b, c, d\}, \{b, c, e\}, \{b, d, e\}\}$. Therefore, the matroid induced by the lower approximation operator is $M(R) = (U, \mathbf{I}(R))$.

The following proposition presents an equivalent formulation of the independent sets of the matroid.

Proposition 4. *Let R be an equivalence relation on U. Then $\boldsymbol{I}(R) = \{X \subseteq U : \forall x \in U, RN(x) \nsubseteq X\}$.*

Proof. We need to prove only $\{X \subseteq U : R_*(X) = \emptyset\} = \{X \subseteq U : \forall x \in U, RN(x) \nsubseteq X\}$. For all $Y \in \{X \subseteq U : R_*(X) = \emptyset\}$, $R_*(Y) = \{x \in U : RN(x) \subseteq Y\} = \emptyset$. Therefore, for all $x \in U$, $RN(x) \nsubseteq Y$. Hence $Y \in \{X \subseteq U : \forall x \in U, RN(x) \nsubseteq X\}$. Conversely, for all $Y \in \{X \subseteq U : \forall x \in U, RN(x) \nsubseteq X\}$, $R_*(Y) = \{x \in U : RN(x) \subseteq Y\} = \emptyset$. Hence $Y \in \{X \subseteq U : R_*(X) = \emptyset\}$. This completes the proof.

The following corollary presents the family of all bases of the matroid, which is denoted by $\mathbf{B}(R)$.

Corollary 1. *Let R be an equivalence relation on U. Then* $\boldsymbol{B}(R) = \{X \subseteq U : \forall x \in U, |RN(x) \bigcap X| = |RN(x)| - 1\}$.

Example 4. (Continued from Example 3) $\mathbf{B}(R) = \{\{a, c, d\}, \{a, c, e\}, \{a, d, e\}, \{b, c, d\}, \{b, c, e\}, \{b, d, e\}\}$.

3.2 The Dual of the Matroid

This subsection investigates three characteristics of the dual of the matroid, which are useful for the next section. The dual of the matroid can be obtained through the family of all bases of the matroid. First of all, the dual of the matroid $M(R)$ is denoted by $M^*(R)$, and the family of all bases of $M^*(R)$ is denoted by $\mathbf{B}^*(R)$.

Proposition 5. *Let R be an equivalence relation on U. Then* $\boldsymbol{B}^*(R) = \{X \subseteq U : \forall x \in U, |RN(x) \bigcap X| = 1\}$.

Proof. According to Corollary 1, $\mathbf{B}(R) = \{X \subseteq U : \forall x \in U, |RN(x) \bigcap X| = |RN(x)| - 1\}$. According to Proposition 2, $\mathbf{B}^*(R) = \{U - X : X \in \mathbf{B}(R)\} = \{X \subseteq U : \forall x \in U, |RN(x) \bigcap X| = 1\}$. This completes the proof.

Example 5. (Continued from Example 3) $\mathbf{B}^*(R) = \{\{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}\}$.

Another two characteristics, which are independent sets and the rank function, are investigated in Corollary 2 and Proposition 6. We denote the family of all independent sets and the rank function of $M^*(R)$ by $\mathbf{I}^*(R)$ and $r_{M^*(R)}$, respectively.

Corollary 2. *Let R be an equivalence relation on U. Then* $\boldsymbol{I}^*(R) = \{X \subseteq U : \forall x \in U, |RN(x) \bigcap X| \leq 1\}$.

Example 6. (Continued from Example 3) $\mathbf{I}^*(R) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}\}$. Hence, the dual of $M(R)$ is $M^*(R) = (U, \mathbf{I}^*(R))$.

Proposition 6. *Let R be an equivalence relation on U. For all $X \subseteq U$,* $r_{M^*(R)}(X) = |\{RN(x) : x \in U, RN(x) \bigcap X \neq \emptyset\}|$.

Proof. According to Definition 7, $r_{M^*(R)}(X) = max\{|I| : I \subseteq X, I \in \mathbf{I}^*(R)\}$. We need to prove only $|\{RN(x) : x \in U, RN(x) \bigcap X \neq \emptyset\}| = max\{|I| : I \subseteq X, I \in \mathbf{I}^*(R)\}$ for all $X \subseteq U$. For all $I \in \mathbf{I}^*(R) = \{X \subseteq U : \forall x \in U, |RN(x) \bigcap X| \leq 1\}$, $|I| = |\{RN(x) : x \in U, RN(x) \bigcap I \neq \emptyset\}|$. Therefore, $max\{|I| : I \subseteq X, I \in \mathbf{I}^*(R)\} = |\{RN(x) : x \in U, RN(x) \bigcap X \neq \emptyset\}|$. This completes the proof.

4 Contraction to the Dual of the Matroid

In this section, we apply contraction to the dual of the matroid. Different contractions produce different matroids, but some characteristics of them may be same.

4.1 Single Contraction to the Dual of the Matroid

This subsection mainly studies the relationships between the contractions of the dual matroid to two subsets, which are the complement of a single point set and the complement of the equivalence class of this point. In other words, two matroids are obtained by applying contraction to the dual matroid, and the relationships between these two matroids are investigated. The following two definitions show that two new matroids are obtained from a matroid by restriction and contraction, respectively.

Definition 8. *(Restriction and deletion [9]) Let $M = (U, \boldsymbol{I})$ be a matroid and $X \subseteq U$. Then $M|X = (X, \boldsymbol{I}_X)$ is a matroid called the restriction of M to X, where $\boldsymbol{I}_X = \{I \subseteq X : I \in \boldsymbol{I}\}$. $M\backslash X = (U - X, \boldsymbol{I}_{U-X})$ is called the deletion of $U - X$ from M.*

Definition 9. *(Contraction [9]) Let $M = (U, \boldsymbol{I})$ be a matroid, $X \subseteq U$ and B_X be a base of $M|X$ (i.e., $B_X \in \boldsymbol{B}(M|X)$). Then $M/X = (U - X, \boldsymbol{I}')$ is a matroid called the contraction of M to $U - X$, where $\boldsymbol{I}' = \{I \subseteq U - X : I \bigcup B_X \in \boldsymbol{I}\}$ (The definition of M/X has no relationship with the selection of $B_X \in \boldsymbol{B}(M|X)$).*

The following proposition investigates the relationship between independent sets of these two matroids obtained by applying contraction to the dual matroid.

Proposition 7. *Let R be an equivalence relation on U. For all $x \in U$, $\boldsymbol{I}(M^*(R)/\{x\}) = \boldsymbol{I}(M^*(R)/RN(x))$.*

Proof. According to Corollary 2 and Definition 8, $\mathbf{I}(M^*(R)|\{x\}) = \{\emptyset, \{x\}\}$ and $\mathbf{I}(M^*(R)|RN(x)) = \{\emptyset\} \bigcup \{\{y\} : y \in RN(x)\}$. According to Definition 6, $\{x\} \in \mathbf{B}(M^*(R)|\{x\})$ and $\{x\} \in \mathbf{B}(M^*(R)|RN(x))$. Thus $\mathbf{I}(M^*(R)/\{x\}) = \{I \subseteq U - \{x\} : I \bigcup \{x\} \in \mathbf{I}^*(R)\}$, $\mathbf{I}(M^*(R)/RN(x)) = \{I \subseteq U - RN(x) : I \bigcup \{x\} \in \mathbf{I}^*(R)\}$. For all $Y \subseteq RN(x) - \{x\}$ and $Y \neq \emptyset$, $Y \bigcup \{x\} \notin \mathbf{I}^*(R)$. Thus $\mathbf{I}(M^*(R)/\{x\}) = \{I \subseteq U - \{x\} : I \bigcup \{x\} \in \mathbf{I}^*(R)\} = \{I \subseteq U - RN(x) : I \bigcup \{x\} \in \mathbf{I}^*(R)\}$. Hence $\mathbf{I}(M^*(R)/\{x\}) = \mathbf{I}(M^*(R)/RN(x))$. This completes the proof.

The following proposition shows the above relationship from the viewpoint of deletion.

Proposition 8. *Let R be an equivalence relation on U. For all $x \in U$, $\boldsymbol{I}(M^*(R)/\{x\}) = \boldsymbol{I}(M^*(R)\backslash RN(x))$.*

Proof. According to Proposition 7, we need to prove only $\mathbf{I}(M^*(R)\backslash RN(x)) = \mathbf{I}(M^*(R)/RN(x))$. According to Definition 8, $\mathbf{I}(M^*(R)\backslash RN(x)) = \{I \subseteq U - RN(x) : I \in \mathbf{I}^*(R)\}$. According to Definition 6, $\{x\} \in \mathbf{B}(M^*(R)|RN(x))$. According to Definition 9, $\mathbf{I}(M^*(R)/RN(x)) = \{I \subseteq U - RN(x) : I \bigcup \{x\} \in \mathbf{I}^*(R)\}$. For any $I \in \mathbf{I}(M^*(R)\backslash RN(x))$, $I \bigcup \{x\} \in \mathbf{I}^*(R)$. In fact, if there exists $I \in \mathbf{I}(M^*(R)\backslash RN(x))$ such that $I \bigcup \{x\} \notin \mathbf{I}^*(R)$, then there exists $y \in I$ such that $y \in RN(x)$, i.e., $I \bigcap RN(x) \neq \emptyset$, which is contradictory with $I \subseteq U - RN(x)$. Therefore, $\mathbf{I}(M^*(R)\backslash RN(x)) \subseteq \mathbf{I}(M^*(R)/RN(x))$. Conversely, according to (I2) of Definition 3, $\mathbf{I}(M^*(R)/RN(x)) \subseteq \mathbf{I}(M^*(R)\backslash RN(x))$. Hence $\mathbf{I}(M^*(R)\backslash RN(x)) = \mathbf{I}(M^*(R)/RN(x))$. This completes the proof.

Note that $M^*(R)/RN(x) = M^*(R)\backslash RN(x)$ for all $x \in U$, because they have the same ground and the same family of independent sets. Therefore, $M^*(R)/RN(x)$ can be replaced by $M^*(R)\backslash RN(x)$ in this subsection.

Example 7. (Continued from Example 6) Since $RN(a) = \{a, b\}$, so $\mathbf{I}(M^*(R)\backslash RN(a)) = \{\emptyset, \{c\}, \{d\}, \{e\}\}$. $\mathbf{I}(M^*(R)|\{a\}) = \{\emptyset, \{a\}\}$ and $\mathbf{I}(M^*(R)|RN(a)) = \{\emptyset, \{a\}, \{b\}\}$. Hence $\{a\} \in \mathbf{B}(M^*(R)|\{a\})$ and $\{a\} \in \mathbf{B}(M^*(R)|RN(a))$. $\mathbf{I}(M^*(R)/\{a\}) = \{I \subseteq U - \{a\} : I \bigcup \{a\} \in \mathbf{I}^*(R)\} = \{\emptyset, \{c\}, \{d\}, \{e\}\}$ and $\mathbf{I}(M^*(R)/RN(a)) = \{\emptyset, \{c\}, \{d\}, \{e\}\}$,i.e., $\mathbf{I}(M^*(R)/\{a\}) = \mathbf{I}(M^*(R)/RN(a)) = \mathbf{I}(M^*(R)\backslash RN(a))$.

The following corollary presents an equivalent formulation of Proposition 7.

Corollary 3. *Let R be an equivalence relation on U. For all $x \in U$, $\boldsymbol{I}(M^*(R)/\{x\}) = \boldsymbol{I}(M^*(R)/R^*(\{x\}))$.*

Not that for any equivalence relation R on U, $R^*(\{x\}) = RN(x)$ for all $x \in U$. Therefore, $RN(x)$ can be replaced by $R^*(\{x\})$ in this subsection. The following proposition investigates the relationship between bases of these two matroids.

Proposition 9. *Let R be an equivalence relation on U. For all $x \in U$, $\boldsymbol{B}(M^*(R)/\{x\}) = \boldsymbol{B}(M^*(R)/RN(x))$.*

Proof. According to Definition 6, $\mathbf{B}(M^*(R)/\{x\}) = Max(\mathbf{I}(M^*(R)/\{x\}))$, and $\mathbf{B}(M^*(R)/RN(x)) = Max(\mathbf{I}(M^*(R)/RN(x)))$. According to Proposition 7, $\mathbf{I}(M^*(R)/\{x\}) = \mathbf{I}(M^*(R)/RN(x))$. Thus $\mathbf{B}(M^*(R)/\{x\}) = \mathbf{B}(M^*(R)/RN(x))$. This completes the proof.

The following lemma shows a relationship between ranks of two subsets of a universe.

Lemma 1. *([9]) Let $M = (U, \boldsymbol{I})$ be a matroid. If $X, Y \subseteq U$ such that for all $y \in Y - X$, $r_M(X) = r_M(X \bigcup \{y\})$, then $r_M(X) = r_M(X \bigcup Y)$.*

The following lemma shows a relationship between the rank functions of a matroid and the contraction of the matroid.

Lemma 2. *([9]) Let $M = (U, \boldsymbol{I})$ be a matroid and $X \subseteq U$. For all $Y \subseteq U - X$, $r_{M/X}(Y) = r_M(X \bigcup Y) - r_M(X)$.*

The following two propositions investigate the relationships between rank functions of these two matroids, and between circuits of these two matroids, respectively.

Proposition 10. *Let R be an equivalence relation on U. For all $x \in U$ and $X \subseteq U - RN(x)$, $r_{M^*(R)/\{x\}}(X) = r_{M^*(R)/RN(x)}(X)$.*

Proof. For all $X \subseteq U - RN(x)$, $X \subseteq U - \{x\}$. According to Lemma 2, $r_{M^*(R)/\{x\}}(X) = r_{M^*(R)}(X \bigcup \{x\}) - r_{M^*(R)}(\{x\})$ and $r_{M^*(R)/RN(x)}(X) = r_{M^*(R)}(X \bigcup RN(x)) - r_{M^*(R)}(RN(x))$. According to Proposition 6, $r_{M^*(R)}(\{x\}) = r_{M^*(R)}(RN(x)) = 1$. So, we need to prove only $r_{M^*(R)}(X \bigcup \{x\}) = r_{M^*(R)}(X \bigcup RN(x))$. For all $y \in (X \bigcup RN(x)) - (X \bigcup \{x\}) = RN(x) - (X \bigcup \{x\}) = RN(x) - \{x\}$, $r_{M^*(R)}(X \bigcup \{x\}) = r_{M^*(R)}(X \bigcup \{x, y\})$. According to Lemma 1, $r_{M^*(R)}(X \bigcup \{x\}) = r_{M^*(R)}((X \bigcup \{x\}) \bigcup (X \bigcup RN(x))) = r_{M^*(R)}(X \bigcup RN(x))$. This completes the proof.

Proposition 11. *Let R be an equivalence relation on U. For all $x \in U$, $\boldsymbol{C}(M^*(R)/RN(x)) \subseteq \boldsymbol{C}(M^*(R)/\{x\})$.*

Proof. According to Proposition 7, $\mathbf{I}(M^*(R)/\{x\}) = \mathbf{I}(M^*(R)/RN(x))$. According to Definition 5, $\mathbf{C}(M^*(R)/\{x\}) = Min(-\mathbf{I}(M^*(R)/\{x\}))$, where $-\mathbf{I}(M^*(R)/\{x\}) = 2^{U-\{x\}} - \mathbf{I}(M^*(R)/\{x\})$, and $\mathbf{C}(M^*(R)/RN(x)) = Min(-\mathbf{I}(M^*(R)/RN(x)))$, where $-\mathbf{I}(M^*(R)/RN(x)) = 2^{U-RN(x)} - \mathbf{I}(M^*(R)/RN(x))$. Therefore, $Min(-\mathbf{I}(M^*(R)/RN(x))) \subseteq Min(-\mathbf{I}(M^*(R)/\{x\}))$, i.e., $\mathbf{C}(M^*(R)/RN(x)) \subseteq \mathbf{C}(M^*(R)/\{x\})$. This completes the proof.

In the following proposition, we study a condition under which these two matroids have the same circuits.

Proposition 12. *Let R be an equivalence relation on U. For all $x \in U$, $\boldsymbol{C}(M^*(R)/\{x\}\backslash RN(x)) = \boldsymbol{C}(M^*(R)/RN(x))$.*

Proof. According to Proposition 7, $\mathbf{I}(M^*(R)/\{x\}) = \mathbf{I}(M^*(R)/RN(x))$. Therefore, for all $I \in \mathbf{I}(M^*(R)/\{x\})$, $I \subseteq U - RN(x)$. Hence $\mathbf{I}(M^*(R)/\{x\}) = \mathbf{I}(M^*(R)/\{x\}\backslash RN(x))$. According to Definition 5, $\mathbf{C}(M^*(R)/\{x\}\backslash RN(x)) = Min(-\mathbf{I}(M^*(R)/\{x\}\backslash RN(x)))$, where $-\mathbf{I}(M^*(R)/\{x\}\backslash RN(x)) = 2^{U-RN(x)} - \mathbf{I}(M^*(R)/\{x\}\backslash RN(x)) = 2^{U-RN(x)} - \mathbf{I}(M^*(R)/\{x\})$, and $\mathbf{C}(M^*(R)/RN(x)) = Min(-\mathbf{I}(M^*(R)/RN(x)))$, where $-\mathbf{I}(M^*(R)/RN(x)) = 2^{U-RN(x)} - \mathbf{I}(M^*(R)/RN(x))$. Hence, $\mathbf{C}(M^*(R)/\{x\}\backslash RN(x)) = \mathbf{C}(M^*(R)/RN(x))$. This completes the proof.

4.2 Complicated Contraction to the Dual of the Matroid

This subsection considers an issue when we apply a sequence of contractions to the dual matroid like the above subsection, can we get the same relationships? In order to solve this problem, the following lemma is presented.

Lemma 3. *([9]) Let $M = (U, \boldsymbol{I})$ be a matroid, $X_1, X_2 \subseteq U$ and $X_1 \bigcap X_2 = \emptyset$. Then $(M/X_1)/X_2 = M/(X_1 \bigcup X_2) = (M/X_2)/X_1$.*

Therefore, the above problem is the relationships between some characteristics of $M^*(R)/X$ and $M^*(R)/R^*(X)$ for all $X \subseteq U$. First of all, $M^*(R)/X$ and $M^*(R)/R^*(X)$ have the same family of independent sets.

Theorem 1. *Let R be an equivalence relation on U. For all $X \subseteq U$, $\boldsymbol{I}(M^*(R)/X) = \boldsymbol{I}(M^*(R)/R^*(X))$.*

Proof. According to Corollary 2 and Definition 8, $\mathbf{I}(M^*(R)|X) = \{I \subseteq X : \forall x \in X, |RN(x) \bigcap I| \leq 1\}$. According to Definition 6, there exists $B_X \in \mathbf{B}(M^*(R)|X)$ such that for any $x \in X$, $|B_X \bigcap RN(x)| = 1$. Therefore, $B_X \in \mathbf{B}(M^*(R)|R^*(X))$. Thus $\mathbf{I}(M^*(R)/X) = \{I \subseteq U - X : I \bigcup B_X \in \mathbf{I}^*(R)\}$, $\mathbf{I}(M^*(R)/R^*(X)) = \{I \subseteq U - R^*(X) : I \bigcup B_X \in \mathbf{I}^*(R)\}$. For all $Y \subseteq R^*(X) - X$ and $Y \neq \emptyset$, $Y \bigcup B_X \notin \mathbf{I}^*(R)$. Thus $\mathbf{I}(M^*(R)/X) = \{I \subseteq U - X : I \bigcup B_X \in \mathbf{I}^*(R)\} = \{I \subseteq U - R^*(X) : I \bigcup B_X \in \mathbf{I}^*(R)\}$. Hence $\mathbf{I}(M^*(R)/X) = \mathbf{I}(M^*(R)/R^*(X))$. This completes the proof.

The following theorem shows the above theorem from the viewpoint of deletion.

Theorem 2. *Let R be an equivalence relation on U. For all $X \subseteq U$, $\boldsymbol{I}(M^*(R)/X) = \boldsymbol{I}(M^*(R)\backslash R^*(X))$.*

Proof. According to Proposition 7, we need to prove only $\mathbf{I}(M^*(R)\backslash R^*(X)) = \mathbf{I}(M^*(R)/R^*(X))$. According to Definition 8, $\mathbf{I}(M^*(R)\backslash R^*(X)) = \{I \subseteq U - R^*(X) : I \in \mathbf{I}^*(R)\}$. Suppose $B_X \in \mathbf{B}(M^*(R)|R^*(X))$, $\mathbf{I}(M^*(R)/R^*(X)) = \{I \subseteq U - R^*(X) : I \bigcup B_X \in \mathbf{I}^*(R)\}$. For any $I \in \mathbf{I}(M^*(R)\backslash R^*(X))$, $I \bigcup B_X \in \mathbf{I}^*(R)$. In fact, if there exists $I \in \mathbf{I}(M^*(R)\backslash R^*(X))$ such that $I \bigcup B_X \notin \mathbf{I}^*(R)$, then there exist $x \in I$, $y \in B_X$ and $x \neq y$ such that $x \in RN(y)$, i.e., $I \bigcap R^*(X) \neq \emptyset$, which is contradictory with $I \subseteq U - R^*(X)$. Therefore, $\mathbf{I}(M^*(R)\backslash R^*(X)) \subseteq \mathbf{I}(M^*(R)/R^*(X))$. Conversely, according to (I2) of Definition 3, $\mathbf{I}(M^*(R)/R^*(X)) \subseteq \mathbf{I}(M^*(R)\backslash R^*(X))$. Hence $\mathbf{I}(M^*(R)\backslash R^*(X)) = \mathbf{I}(M^*(R)/R^*(X))$. This completes the proof.

Note that $M^*(R)/R^*(X) = M^*(R)\backslash R^*(X)$ for all $X \subseteq U$. Therefore, $M^*(R)/R^*(X)$ can be replaced by $M^*(R)\backslash R^*(X)$ in this subsection. The relationships between another characteristics of $M^*(R)/X$ and $M^*(R)/R^*(X)$, which are bases, rank functions and circuits, are investigated in Proposition 13, Theorem 3 and Theorem 4.

Proposition 13. *Let R be an equivalence relation on U. For all $X \subseteq U$, $\boldsymbol{B}(M^*(R)/X) = \boldsymbol{B}(M^*(R)/R^*(X))$.*

Proof. According to Definition 6, $\mathbf{B}(M^*(R)/X) = Max(\mathbf{I}(M^*(R)/X))$, and $\mathbf{B}(M^*(R)/R^*(X)) = Max(\mathbf{I}(M^*(R)/R^*(X)))$. According to Theorem 1, $\mathbf{I}(M^*(R)/X) = \mathbf{I}(M^*(R)/R^*(X))$. Thus $\mathbf{B}(M^*(R)/X) = \mathbf{B}(M^*(R)/R^*(X))$. This completes the proof.

Theorem 3. *Let R be an equivalence relation on U. For all $X \subseteq U$ and $Y \subseteq U - R^*(X)$, $r_{M^*(R)/X}(Y) = r_{M^*(R)/R^*(X)}(Y)$.*

Proof. For all $Y \subseteq U - R^*(X)$, $Y \subseteq U - X$. According to Lemma 2, $r_{M^*(R)/X}(Y) = r_{M^*(R)}(X \bigcup Y) - r_{M^*(R)}(X)$ and $r_{M^*(R)/R^*(X)}(Y) = r_{M^*(R)}(R^*(X) \bigcup Y) - r_{M^*(R)}(R^*(X))$. According to Proposition 6, $r_{M^*(R)}(X) = r_{M^*(R)}(R^*(X))$. So, we need to prove only $r_{M^*(R)}(X \bigcup Y) = r_{M^*(R)}(R^*(X) \bigcup Y)$. For all $y \in (R^*(X) \bigcup Y) - (X \bigcup Y) = R^*(X) - (X \bigcup Y) = R^*(X) - X$, $r_{M^*(R)}(X \bigcup Y) = r_{M^*(R)}(X \bigcup Y \bigcup \{y\})$. According to Lemma 1, $r_{M^*(R)}(X \bigcup Y) = r_{M^*(R)}((X \bigcup Y) \bigcup (R^*(X) \bigcup Y)) = r_{M^*(R)}(R^*(X) \bigcup Y)$. This completes the proof.

Theorem 4. *Let R be an equivalence relation on U. For all $X \in U$, $\boldsymbol{C}(M^*(R)/R^*(X)) \subseteq \boldsymbol{C}(M^*(R)/X)$.*

Proof. According to Theorem 1, $\mathbf{I}(M^*(R)/X) = \mathbf{I}(M^*(R)/R^*(X))$. According to Definition 5, $\mathbf{C}(M^*(R)/X) = Min(-\mathbf{I}(M^*(R)/X))$, where $-\mathbf{I}(M^*(R)/X) = 2^{U-X} - \mathbf{I}(M^*(R)/X)$, and $\mathbf{C}(M^*(R)/R^*(X)) = Min(-\mathbf{I}(M^*(R)/R^*(X)))$, where $-\mathbf{I}(M^*(R)/RN(x)) = 2^{U-(R^*(X))} - \mathbf{I}(R^*(X))$. Therefore, $Min(-\mathbf{I}(M^*(R)/R^*(X))) \subseteq Min(-\mathbf{I}(M^*(R)/X))$, i.e., $\mathbf{C}(M^*(R)/R^*(X)) \subseteq \mathbf{C}(M^*(R)/X)$. This completes the proof.

The following proposition presents a condition under which $M^*(R)/R^*(X)$ and $M^*(R)/X$ have the same circuits.

Proposition 14. *Let R be an equivalence relation on U. For all $X \subseteq U$, $\boldsymbol{C}(M^*(R)/X \setminus R^*(X)) = \boldsymbol{C}(M^*(R)/R^*(X))$.*

Proof. According to Theorem 1, $\mathbf{I}(M^*(R)/X) = \mathbf{I}(M^*(R)/R^*(X))$. Therefore, for all $I \in \mathbf{I}(M^*(R)/X)$, $I \subseteq U - R^*(X)$. Hence $\mathbf{I}(M^*(R)/X) = \mathbf{I}(M^*(R)/X \setminus R^*(X))$. According to Definition 5, $\mathbf{C}(M^*(R)/X \setminus R^*(X)) = Min(-\mathbf{I}(M^*(R)/X \setminus R^*(X)))$, where $-\mathbf{I}(M^*(R)/X \setminus R^*(X)) = 2^{U-R^*(X)} - \mathbf{I}(M^*(R)/X \setminus R^*(X)) = 2^{U-R^*(X)} - \mathbf{I}(M^*(R)/X)$, and $\mathbf{C}(M^*(R)/R^*(X)) = Min(-\mathbf{I}(M^*(R)/R^*(X)))$, where $-\mathbf{I}(M^*(R)/R^*(X)) = 2^{U-R^*(X)} - \mathbf{I}(M^*(R)/R^*(X))$. Hence $\mathbf{C}(M^*(R)/X \setminus R^*(X)) = \mathbf{C}(M^*(R)/R^*(X))$. This completes the proof.

5 Conclusions

In this paper, we establish a matroid through the lower approximation operator for any Pawlak's approximation space. We investigate the dual of the matroid and its characteristics which are independent sets, bases and the rank function. We study the relationships between the contractions of the dual matroid to two subsets, which are the complement of a single point set and the complement of the equivalence class of this point. Moreover, these relationships are extended to general cases. We will do more works in combining rough sets and matroids.

Acknowledgments. 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, the Science and Technology Key Project of Fujian Province, China, under Grant No. 2012H0043, 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, Y., Miao, D., Wang, R., Wu, K.: A rough set approach to feature selection based on power set tree. Knowledge-Based Systems 24, 275–281 (2011)
2. Dai, J., Wang, W., Xu, Q., Tian, H.: Uncertainty measurement for interval-valued decision systems based on extended conditional entropy. Knowledge-Based Systems 27, 443–450 (2012)
3. Dougherty, R., Freiling, C., Zeger, K.: Networks, matroids, and non-shannon information inequalities. IEEE Transactions on Information Theory 53, 1949–1969 (2007)
4. Du, Y., Hu, Q., Zhu, P., Ma, P.: Rule learning for classification based on neighborhood covering reduction. Information Sciences 181, 5457–5467 (2011)
5. Edmonds, J.: Matroids and the greedy algorithm. Mathematical Programming 1, 127–136 (1971)
6. Hu, Q., Yu, D., Liu, J., Wu, C.: Neighborhood rough set based heterogeneous feature subset selection. Information Sciences 178, 3577–3594 (2008)

7. Jia, X., Liao, W., Tang, Z., Shang, L.: Minimum cost attribute reduction in decision-theoretic rough set models. Information Sciences 219, 151–167 (2013)
8. Kryszkiewicz, M.: Rough set approach to incomplete information systems. Information Sciences 112, 39–49 (1998)
9. Lai, H.: Matroid theory. Higher Education Press, Beijing (2001)
10. Lawler, E.: Combinatorial optimization: networks and matroids. Dover Publications (2001)
11. Leung, Y., Fung, T., Mi, J., Wu, W.: A rough set approach to the discovery of classification rules in spatial data. International Journal of Geographical Information Science 21, 1033–1058 (2007)
12. Li, Y.: Some researches on fuzzy matroids. PhD thesis, Shaanxi Normal University (2007)
13. Mao, H.: The relation between matroid and concept lattice. Advances in Mathematics 35, 361–365 (2006)
14. Matus, F.: Abstract functional dependency structures. Theoretical Computer Science 81, 117–126 (1991)
15. Min, F., Zhu, W.: Attribute reduction of data with error ranges and test costs. Information Sciences 211, 48–67 (2012)
16. Pawlak, Z.: Rough sets. ICS PAS Reports 431 (1981)
17. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
18. Qin, K., Yang, J., Pei, Z.: Generalized rough sets based on reflexive and transitive relations. Information Sciences 178, 4138–4141 (2008)
19. Rouayheb, S.Y.E., Sprintson, A., Georghiades, C.N.: On the index coding problem and its relation to network coding and matroid theory. IEEE Transactions on Information Theory 56, 3187–3195 (2010)
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. Yao, Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111, 239–259 (1998)
23. Yao, Y.: Constructive and algebraic methods of theory of rough sets. Information Sciences 109, 21–47 (1998)
24. Zhong, N.: Rough sets in knowledge discovery and data mining. Journal of Japan Society for Fuzzy Theory and Systems 13, 581–591 (2001)
25. Zhu, W., Wang, F.: Reduction and axiomization of covering generalized rough sets. Information Sciences 152, 217–230 (2003)
26. Zhu, W., Wang, S.: Matroidal approaches to generalized rough sets based on relations. International Journal of Machine Learning and Cybernetics 2, 273–279 (2011)
27. Zhu, W., Wang, S.: Rough matroid based on relations. Information Sciences 232, 241–252 (2013)

Optimal Approximations with Rough Sets

Ryszard Janicki and Adam Lenarčič

Department of Computing and Software,
McMaster University,
Hamilton, ON, L8S 4K1 Canada
{janicki,lenarcaj}@mcmaster.ca

Abstract. When arbitrary sets are approximated by more structured sets, it may not be *possible* to obtain an exact approximation that is equivalent to a given set. A proposal is presented for a 'metric' approach to Rough Sets. This includes a definition of the 'optimal' or best approximation with respect to a measure of similarity, and an algorithm to find it using the Jaccard Index. A definition of consistency also allows the algorithm to work for a larger class of similarity measures. Several consequences of these definitions are also presented.

1 Introduction and Motivation

It appears that the concept of approximation has two different intuitions in pure mathematics and science in general. The first one stems from the fact that often, empirical numerical data have errors, so in reality we seldom have the value x (unless the measurements are expressible in integers) but usually some interval $(x-\varepsilon, x+\varepsilon)$, i.e. the *lower* and *upper approximations*. Rough Sets [1,2] exploit this idea for general sets.

The second intuition can be illustrated by the *linear least squares approximation* of points in the two dimensional plane (credited to C. F. Gauss, 1795, c.f. [3]). Here we know or assume that the points should be on a straight line and we are trying to find the line that fits the data best. However, this is not the case of an upper, or lower approximation in the sense of Rough Sets. Even if we replace a solution $f(x) = ax+b$ by two lines $f_1(x) = ax+b-\delta$ and $f_2(x) = ax+b+\delta$, where δ is a standard error (c.f. [3]), there is no guarantee that any point resides between $f_1(x)$ and $f_2(x)$. This approach assumes that there is a well defined concept of *similarity* (or *distance*) and some techniques for finding maximal similarity (minimal distance) between entities and their approximations.

In this paper we will propose a 'metric' or standard of measurement for comparison within the framework of Rough Sets [1]. We start with an introduction to terminology, and then present axioms which should hold for any similarity function. We continue by providing four similarity measures for arbitrary sets, and a generalized definition of what it means for a set to be an optimal approximation (with respect to a given measure). Later we show properties of the classical Jaccard similarity index [4] and provide an efficient greedy algorithm using the index which yields an optimal approximation. We also recognize that in some situations, different similarity functions will be equivalent in showing which of two sets is a better approximation. If two indexes demonstrate this equivalence in all cases we will call them consistent. Based on their consistency,

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 87–98, 2013.

we demonstrate that the algorithm we derive using the Jaccard similarity index, can be used for the Dice-Sørenson similarity index as well [5,6].

2 Rough Sets and Borders

In this chapter we introduce, review, and also adapt for our purposes, some general ideas that are crucial to our approach.

The principles of Rough Sets [1,2] can be formulated as follows.

Let U be a finite and non-empty universe of elements, and let $E \subseteq U \times U$ be an *equivalence relation*. Recall that for each $E \subseteq U \times U$, $[x]_E$ will denote the equivalence class of E containing x, and U/E will denote the set of all equivalence classes of E.

The elements of $\mathfrak{Comp} = U/E$ are called *elementary sets* or *components* and they are interpreted as basic observable, measurable, or definable sets. We will denote the elements of $\mathfrak{Comp}$, i.e. equivalence classes of E, by bold symbols, and write for example $\mathbf{x} \in \mathbb{B} \subseteq \mathfrak{Comp}$.

The pair (U,E) is referred to as a Pawlak approximation space.

A non-empty set $X \subseteq U$ is approximated by two subsets of U; $\underline{\mathbf{A}}(X)$ and $\overline{\mathbf{A}}(X)$, called the lower and upper approximations of X respectively, and are defined as follows:

Definition 1 ([1,2]). *For each $X \subseteq U$,*

1. $\underline{\mathbf{A}}(X) = \bigcup\{\mathbf{x} \mid \mathbf{x} \in \mathfrak{Comp} \wedge \mathbf{x} \subseteq X\}$,
2. $\overline{\mathbf{A}}(X) = \bigcup\{\mathbf{x} \mid \mathbf{x} \in \mathfrak{Comp} \wedge \mathbf{x} \cap X \neq \emptyset\}$. □

Clearly $\underline{\mathbf{A}}(X) \subseteq X \subseteq \overline{\mathbf{A}}(X)$. There are many versions and many extensions of this basic model, see for example [7,8,9,10], as well as many various applications (cf. [11,12,9,13]). Even robotic locomotion can utilize this notion to ensure it remains within bounds, and could also use measures of similarity to move based on the best/optimal available (representable) approximation of its surroundings [14].

A set $A \subseteq U$ is *definable* (or *exact*) [2] if it is a union of some equivalence classes of the equivalence relation E. Let $\mathbb{D}$ denote the family of all definable sets defined by the space (U,E). Formally

$$\mathsf{A} \in \mathbb{D} \iff \exists \mathbf{x}_1,\ldots,\mathbf{x}_n \subseteq \mathfrak{Comp}.\ \mathsf{A} = \mathbf{x}_1 \cup \ldots \cup \mathbf{x}_n.$$

We would like to point out the duality of $\mathfrak{Comp}$ and $\mathbb{D}$. Each set of components $\mathsf{C} \subseteq \mathfrak{Comp}$ uniquely defines the *definable set* $\mathrm{dset}(\mathsf{C}) \in \mathbb{D}$, as $\mathrm{dset}(\mathsf{C}) = \bigcup_{\mathbf{x} \in \mathsf{C}} \mathbf{x}$, and each definable set $\mathsf{A} \in \mathbb{D}$ uniquely defines the *set of components* $\mathrm{comp}(\mathsf{A}) \subseteq \mathfrak{Comp}$, by $\mathrm{comp}(\mathsf{A}) = \{\mathbf{x} \mid \mathbf{x} \subseteq \mathsf{A}\}$.

Moreover, for each set of components $\mathsf{C} \subseteq \mathfrak{Comp}$ we have $\mathrm{comp}(\mathrm{dset}(\mathsf{C})) = \mathsf{C}$, and for each definable set $\mathsf{A} \in \mathbb{D}$ we have $\mathrm{dset}(\mathrm{comp}(\mathsf{A})) = \mathsf{A}$.

Clearly every lower and upper approximation is a definable set, i.e. $\underline{\mathbf{A}}(X) \in \mathbb{D}$ and $\overline{\mathbf{A}}(X) \in \mathbb{D}$ for every $X \subseteq U$. Furthermore, all definable sets are equal to their lower and upper approximations, as the below corollary shows.

Corollary 1. *For every $X \subseteq U$, $X \in \mathbb{D} \iff \underline{\mathbf{A}}(X) = \overline{\mathbf{A}}(X) = X$.* □

Since the definable sets in the area between the upper and lower approximations will play an important role in our model, we need to precisely define this area.

Definition 2. *For every $X \subseteq U$, we define the set of components $\mathfrak{B}(X) \subseteq \mathfrak{Comp}$ called the **border** of X, and the set of **border sets** of X called $\mathbb{B}(X) \subseteq \mathbb{D}$, as follows:*

1. $\mathbf{x} \in \mathfrak{B}(X) \iff \mathbf{x} \in \mathrm{comp}(\overline{\mathbf{A}}(X)) \setminus \mathrm{comp}(\underline{\mathbf{A}}(X))$,
2. $\mathsf{A} \in \mathbb{B}(X) \iff \mathsf{A} \subseteq \overline{\mathbf{A}}(X) \setminus \underline{\mathbf{A}}(X) \wedge \mathsf{A} \in \mathbb{D}$. □

The corollary below describes basic properties of borders and border sets.

Corollary 2. *For every $X \subseteq U$,*

1. $\mathrm{dset}(\mathfrak{B}(X)) = \overline{\mathbf{A}}(X) \setminus \underline{\mathbf{A}}(X) \in \mathbb{B}(X)$ *and* $\mathfrak{B}(X) \subseteq \mathbb{B}(X)$,
2. $\mathsf{A} \in \mathbb{B}(X) \iff \exists \mathbf{x}_1, \ldots, \mathbf{x}_n \subseteq \mathfrak{B}(X).\ \mathsf{A} = \mathbf{x}_1 \cup \ldots \cup \mathbf{x}_n$,
3. *if* $\mathsf{A} \in \mathbb{B}(X)$ *then* $\mathsf{A} \cap X \neq \emptyset$ *and* $\mathsf{A} \setminus X \neq \emptyset$. □

3 Similarity Measures and Optimal Approximations

The model that will be proposed in this paper requires the concept of some measure of *similarity* between two sets. It is important to point out that we need a similarity measure between *sets*, but *not* between *elements* (as for instance in [13]), and that this measure does not assume any specific interpretation of sets (as for instance in [12]). Under the present context, we assume that all elements are of equal importance, and their specific properties do not influence the similarity measure between sets.

Suppose that we have a (total) function $sim : 2^U \times 2^U \to [0,1]$ that measures *similarity* between sets. We assume that the function *sim* satisfies the following five, intuitive axioms. Namely, for all sets A, B, we have:

$$
\begin{array}{ll}
S1: & sim(A,B) = 1 \iff A = B, \\
S2: & sim(A,B) = sim(B,A), \\
S3: & sim(A,B) = 0 \iff A \cap B = \emptyset, \\
S4: & \text{if } a \in B \setminus A \text{ then } sim(A,B) < sim(A \cup \{a\}, B), \\
S5: & \text{if } a \notin A \cup B \text{ and } A \cap B \neq \emptyset \text{ then } sim(A,B) > sim(A \cup \{a\}, B)
\end{array}
$$

Depending on the area of application, similarity functions may have various properties, however all known versions seem to satisfy the above five axioms, or their equivalent formulation (c.f. [15,16]). The axioms $S1$–$S3$ are often expressed explicitly, sometimes enriched with additional axioms (c.f. [15]), while the axioms $S4$ and $S5$, although satisfied in most versions, are probably formulated explicitly for the first time.

The first axiom ensures that if and only if a similarity measure returns one, the two sets are equal. The second axiom is the reflexivity of similarity measures, meaning that one set is the same distance from a second set, as the second set is from the first, and the third axiom states that if two sets do not share any elements, their similarity is zero, and vice versa.

The axioms $S4$ and $S5$ deal with changing sizes of sets. We will call them *monotonicity axioms*. Axiom $S4$ dictates that if we add part of B to A, the result is closer to

B than A alone, while axiom $S5$ reduces to the notion that if we add to A some new element not in B, then the result is more distant from B than A alone. The axiom $S5$ is only applicable when the sets being compared have at least one common element, i.e. $sim(A,B) > 0$.

We will also say that a measure of similarity *sim* is *metrical* (i.e. it is a suitable tool to evaluate distance between two sets), if the function $diff(A,B) = 1 - sim(A,B)$ is a proper metric which holds for all pairs of rough sets in our universe [17].

The first similarity measure was proposed in 1901 by P. Jaccard [4]. It is still one of the most popular, however the following similarity measures are also prominent in the literature at this point in time:

- *Jaccard index* [4]: $sim_J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$,
- *Dice-Sørensen index* [5,6]: $sim_{DS}(X,Y) = \frac{2|X \cap Y|}{|X|+|Y|}$,
- *Tversky index* [16]: $sim_T^{a,b}(X,Y) = \frac{|X \cap Y|}{|X \cap Y| + a|X \setminus Y| + b|Y \setminus X|}$, where $a,b \geq 0$ are parameters. Note that for $a = b = 1$, $sim_T^{a,b}(X,Y) = sim_J(X,Y)$ and for $a = b = 0.5$, $sim_T^{a,b}(X,Y) = sim_{DS}(X,Y)$.
- *Fuzzy Sets index* [15]: $sim_{fs}(X,Y) = \frac{|X \cap Y|}{\max(|X|,|Y|)}$.

All the similarity indexes above have values between 0 and 1 and it can be shown that they all satisfy the similarity axioms $S1$–$S5$. The advantage of the Jaccard index is that it is metrical (i.e. $diff_J(X,Y) = 1 - sim_J(X,Y)$ is a proper metric), which is not true for the Dice-Sørensen and Fuzzy Sets indexes. Also note that $diff_J(X,Y) = \frac{|(X \setminus Y) \cup (Y \setminus X)|}{|X \cup Y|}$, which appears to have a natural interpretation, while $diff_{DS}(X,Y)$ and $diff_{fs}(X,Y)$ look rather artificial. The Tversky index looks quite flexible, however, this might make it difficult in practice to provide specific values of a and b (different from 1 or 0.5) with any justification. Other techniques could, however, be used to either determine them outright, or optimize them, possibly through some learning process.

We can now provide our *general* definition of optimal approximation.

Definition 3. *For every set $X \subseteq U$, a definable set $\mathsf{O} \in \mathbb{D}$ is an **optimal approximation** of X (w.r.t. a given* similarity measure *sim) if and only if:*

$$sim(X,\mathsf{O}) = \max_{\mathsf{A} \in \mathbb{D}}(sim(X,\mathsf{A}))$$

The set of all optimal approximations of X will be denoted by $\mathbf{Opt}_{sim}(X)$. □

A specific optimal approximation depends on the precise definition of the similarity measure *sim*. If $sim_1 \neq sim_2$ then clearly $\mathbf{Opt}_{sim_1}(X)$ might differ from $\mathbf{Opt}_{sim_2}(X)$ for some $X \subseteq U$.

Axioms $S4$ and $S5$ imply that all optimal approximations reside between lower and upper approximations (inclusive), for all similarity measures satisfying axioms $S1$–$S5$.

Proposition 1. *For every set $X \subseteq U$, and every $\mathsf{O} \in \mathbf{Opt}_{sim}(X)$, we have*

$$\underline{\mathbf{A}}(X) \subseteq \mathsf{O} \subseteq \overline{\mathbf{A}}(X)$$

Proof. Suppose that $C = \underline{\mathbf{A}}(X) \setminus \mathsf{O} \neq \emptyset$. Since $C \subseteq X$, then by axiom $S4$, $sim(\mathsf{O},X) < sim(\mathsf{O} \cup C,X)$, so O must not be optimal. Now suppose that $C = \mathsf{O} \setminus \overline{\mathbf{A}}(X) \neq \emptyset$. By axiom $S5$, $sim(\mathsf{O} \setminus C,X) > sim(\mathsf{O},X)$, so O must not be optimal again. □

One of the consequences of Proposition 1 is that any optimal approximation of X, is the union of the lower approximation of X and some element $\mathsf{A} \in \{\mathbb{B}(X) \cup \emptyset\}$.

Definition 4. *Let $X \subseteq U$, and $\mathsf{O} \in \mathbb{D}$. We say that O is an **intermediate approximation** of X, if*

$$\underline{\mathbf{A}}(X) \subseteq \mathsf{O} \subseteq \overline{\mathbf{A}}(X)$$

The set of all intermediate approximations of X will be denoted by $\mathbf{IA}_{sim}(X)$. □

From Proposition 1 we have:

Corollary 3. *For each set $X \subseteq U$, $\mathbf{Opt}_{sim}(X) \subseteq \mathbf{IA}_{sim}(X)$ and if $\mathsf{O} \in \mathbf{Opt}_{sim}(X)$ then there exist some $\mathsf{A},\mathsf{B} \in \{\mathbb{B}(X) \cup \emptyset\}$ such that $\mathsf{O} = \underline{\mathbf{A}}(X) \cup \mathsf{A} = \overline{\mathbf{A}}(X) \setminus \mathsf{B}$.* □

These are properties of optimal approximations. The set of them must be a portion of the intermediate approximations. Any optimal approximation must be the union of the lower approximation with some definable set which is in the upper but not the lower approximation (or is the empty set itself). It must also be possible to represent any optimal approximation by the upper approximation with some border set removed from it (or the empty set if the approximation is optimal).

The notion of optimal approximation also introduces some structure to the current available field of similarity measures, as certain different similarity indexes may generate the same optimal approximations.

Definition 5. *We say that two similarity indexes sim_1 and sim_2 are **consistent** if for all sets $A,B,C \subseteq U$,*

$$sim_1(A,B) < sim_1(A,C) \iff sim_2(A,B) < sim_2(A,C).$$ □

This clearly leads to the following result.

Corollary 4. *If sim_1 and sim_2 are consistent then for each $X \subseteq U$,*

1. $\mathbf{Opt}_{sim_1}(X) = \mathbf{Opt}_{sim_2}(X)$.
2. *sim_1 satisfies the axioms $S4$ and $S5$ if and only if sim_2 satisfies them.* □

This concept will allow us to extend results and algorithms designed for specific similarity indexes, to larger classes of consistent indexes.

So far we have not used any specific similarity measure. We only assumed that the function *sim* satisfies the axioms $S1$–$S5$. However to show more specific and detailed properties of optimal approximations, especially an efficient algorithm to find one, we need to choose a specific similarity measure.

4 Optimal Approximations with Jaccard Similarity Measure

In what follows we assume that similarity is defined as the Jaccard index, i.e. $sim_J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$, which seems suitable since it is the only popular index which is metrical. It may also seem more *intuitive* to use the size of the union of two sets–a standard set operator–than to use find the sum of the magnitudes of each set (thus counting the elements in the intersection twice, as in the Dice-Sørensen index), or to take only the larger set of the two (disregarding the size of the smaller set, as in the Fuzzy Set index). Though, the difficulty of measurement and calculation for Dice-Sørensen index is indeed identical since $|X| + |Y| = |X \cap Y| + |X \cup Y|$. Also note that in this section we write just $\mathbf{Opt}(X)$ instead of $\mathbf{Opt}_{sim_J}(X)$, and $\mathbf{IA}(X)$ instead of $\mathbf{IA}_{sim_J}(X)$.

First we show that the Jaccard index really satisfies the axioms $S1$–$S5$, so the property specified by Proposition 1 and Definition 4 is satisfied.

Proposition 2. *The function* $sim_J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$ *satisfies the axioms* $S1$–$S5$.

Proof. Only $S4$ and $S5$ are not immediately obvious. If $a \in B \setminus A$, then $a \notin A \cap B$, so $|(A \cup \{a\}) \cap B| = |A \cap B| + 1$. On the other hand $A \cup B = (A \cup \{a\}) \cup B$, so $sim_J(A,B) = \frac{|A \cap B|}{|A \cup B|} < \frac{|A \cap B|+1}{|A \cup B|} = sim_J(A \cup \{a\}, B)$. Hence, $S4$ holds, and a similar process can be performed to show $S5$ holds as well. □

Now, suppose that $\mathsf{O} \in \mathbf{IA}(X)$ is an intermediate approximation of X, and $\mathbf{x} \in \mathfrak{B}(X)$ is an element of the border of X which has no common element with O, i.e. $\mathsf{O} \cap \mathbf{x} = \emptyset$. To determine which definable set is a better approximation of X (more similar to X), O or $\mathsf{O} \cup \mathbf{x}$, we can use the lemma below.

Lemma 1. *Let* $X \subseteq U$, $\mathsf{O} \in \mathbf{IA}(X)$, $\mathsf{A}, \mathsf{B} \in \mathbb{B}(X)$, $\mathsf{A} \cap \mathsf{O} = \emptyset$, *and* $\mathsf{B} \subseteq \mathsf{O}$. *Then*

1. $sim_J(X, \mathsf{O} \cup \mathsf{A}) \geq sim_J(X, \mathsf{O}) \iff \frac{|\mathsf{A} \cap X|}{|\mathsf{A} \setminus X|} \geq \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$
2. $sim_J(X, \mathsf{O} \setminus \mathsf{B}) \leq sim_J(X, \mathsf{O}) \iff \frac{|\mathsf{B} \cap X|}{|\mathsf{B} \setminus X|} \geq \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$

Proof. (1) Let $|X \cap \mathsf{O}| = n$, $|X \cup \mathsf{O}| = m$, $|\mathsf{A} \setminus X| = l$, and $|\mathsf{A} \cap X| = k$. By Corollary 2(3), n, m, l, k are all bigger than zero.
We have $sim_J(X, \mathsf{O}) = \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$ and $sim_J(X, \mathsf{O} \cup \mathsf{A}) = \frac{|X \cap (\mathsf{O} \cup \mathsf{A})|}{|X \cup (\mathsf{O} \cup \mathsf{A})|}$. Because $\mathsf{A} \cap \mathsf{O} = \emptyset$, $|X \cap (\mathsf{O} \cup \mathsf{A})| = |X \cap \mathsf{O}| + |X \cap \mathsf{A}| = n + k$ and $|X \cup (\mathsf{O} \cup \mathsf{A})| = |X \cup \mathsf{O}| + |\mathsf{A} \setminus X| = m + l$. Hence, $sim_J(X, \mathsf{O} \cup \mathsf{A}) \geq sim_J(X, \mathsf{O}) \iff \frac{n+k}{m+l} \geq \frac{n}{m} \iff \frac{k}{l} \geq \frac{n}{m} \iff \frac{|\mathsf{A} \cap X|}{|\mathsf{A} \setminus X|} \geq \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$.
(2) Let $|X \cap \mathsf{O}| = n$, $|X \cup \mathsf{O}| = m$, $|\mathsf{B} \setminus X| = l$, and $|\mathsf{B} \cap X| = k$. By Corollary 2(3), n, m, l, k are all bigger than zero.
We have here $sim_J(X, \mathsf{O}) = \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$ and $sim_J(X, \mathsf{O} \setminus \mathsf{B}) = \frac{|X \cap (\mathsf{O} \setminus \mathsf{B})|}{|X \cup (\mathsf{O} \setminus \mathsf{B})|}$. Because $\mathsf{B} \subseteq \mathsf{O}$, $|X \cap (\mathsf{O} \setminus \mathsf{B})| = |X \cap \mathsf{O}| - |X \cap \mathsf{B}| = n - k$ and $|X \cup (\mathsf{O} \setminus \mathsf{B})| = |X \cup \mathsf{O}| - |\mathsf{B} \setminus X| = m - l$.
Thus, $sim_J(X, \mathsf{O} \setminus \mathsf{B}) \leq sim_J(X, \mathsf{O}) \iff \frac{n-k}{m-l} \leq \frac{n}{m} \iff \frac{k}{l} \geq \frac{n}{m} \iff \frac{|\mathsf{B} \cap X|}{|\mathsf{B} \setminus X|} \geq \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$. □

Clearly the above lemma also holds for $\mathsf{A} = \mathbf{x} \in \mathfrak{B}(X)$. Intuitively, if more than half of the elements of $\mathbf{x}$ also belong to X, or equivalently, if more elements of $\mathbf{x}$ belong to X than do not, the rough set $\mathsf{O} \cup \mathbf{x}$ should approximate X better than O. The results below support this intuition.

Corollary 5 ('Majority Rule'). *Let* $X \subseteq U$, $\mathsf{O} \in \mathbf{IA}(X)$, $\mathbf{x} \in \mathfrak{B}(X)$, *and* $\mathbf{x} \cap \mathsf{O} = \emptyset$. *Then:* $|\mathbf{x} \cap X| \geq |\mathbf{x} \setminus X| \iff \frac{|\mathbf{x} \cap X|}{|\mathbf{x}|} \geq \frac{1}{2} \implies sim_J(X, \mathsf{O} \cup \mathbf{x}) \geq sim_J(X, \mathsf{O})$.

Proof. Clearly $|\mathbf{x} \cap X| \geq |\mathbf{x} \setminus X| \iff \frac{|\mathbf{x} \cap X}{|\mathbf{x} \setminus X|} \geq 1$. But $\frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|} \leq 1$, so by Lemma 1, $sim_J(X, \mathsf{O} \cup \mathbf{x}) \geq sim_J(X, \mathsf{O})$. □

However, the reciprocal of Corollary 5 does not hold. It may happen that $\frac{|\mathbf{x} \cap X|}{|\mathbf{x}|} < \frac{1}{2}$, but the rough set $\mathsf{O} \cup \mathbf{x}$ still approximates X better than O.

We know from Proposition 1 that if $\mathsf{O} \in \mathbf{Opt}(X)$, then either $\mathsf{O} = \underline{\mathbf{A}}(X)$, or $\mathsf{O} = \underline{\mathbf{A}}(X) \cup \mathbf{x}_1 \cup \ldots \cup \mathbf{x}_k$, for some $k \geq 1$, where each $\mathbf{x}_i \in \mathfrak{B}(X)$, $i = 1, \ldots, k$. Lemma 1 allows us to explicitly define these $\mathbf{x}_i \in \mathfrak{B}(X)$ components.

Theorem 1. *For every* $X \subseteq U$, *the following two statements are equivalent:*

1. $\mathsf{O} \in \mathbf{Opt}(X)$
2. $\mathsf{O} \in \mathbf{IA}(X) \wedge \Big(\forall \mathbf{x} \in \mathfrak{B}(X).\, \mathbf{x} \subseteq \mathsf{O} \iff \frac{|\mathbf{x} \cap X|}{|\mathbf{x} \setminus X|} \geq \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|} = sim_J(X, \mathsf{O})\Big)$.

Proof. (1)⇒(2) By Proposition 1, $\mathsf{O} \in \mathbf{IA}(X)$. Let $\mathbf{x} \in \mathfrak{B}(X)$ and $\mathbf{x} \subseteq \mathsf{O}$. Suppose that $\frac{|\mathbf{x} \cap X|}{|\mathbf{x} \setminus X|} < \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$. Then by Lemma 1, $sim_J(X, \mathsf{O} \setminus \mathbf{x}) > sim_J(X, \mathsf{O})$, so O is not optimal. Let $\frac{|\mathbf{x} \cap X|}{|\mathbf{x} \setminus X|} \geq \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$. Suppose that $\mathbf{x} \in \mathfrak{B}(X)$ and $\mathbf{x} \cap \mathsf{O} = \emptyset$. By Corollary 2(3), $|\mathbf{x} \cap X| \neq 0$, so let $a \in \mathbf{x} \cap X$. Since $\mathbf{x} \cap \mathsf{O} = \emptyset$, then $a \in X \setminus \mathsf{O}$. Then by Proposition 2 and axiom *S*4, $sim_J(X, \mathsf{O} \cup \{a\}) > sim_J(X, \mathsf{O})$, so O is not optimal. Note that Lemma 1 gives only $sim_J(X, \mathsf{O} \cup \mathbf{x}) \geq sim_J(X, \mathsf{O})$ which is not strong enough.
(2)⇒(1) Suppose O satisfies (2) but $\mathsf{O} \notin \mathbf{Opt}(X)$. Let $\mathsf{Q} \in \mathbf{Opt}(X)$. Hence, by the proof (1)⇒(2), Q satisfies (2). We have to consider two cases $\mathsf{Q} \setminus \mathsf{O} \neq \emptyset$ and $\mathsf{O} \setminus \mathsf{Q} \neq \emptyset$.

(*Case 1*) Let $\mathsf{Q} \setminus \mathsf{O} \neq \emptyset$ and let $\mathbf{y} \in \mathfrak{B}(X)$ be such that $\mathbf{y} \subseteq \mathsf{Q} \setminus \mathsf{O}$. Since Q satisfies (2), we have $\frac{|\mathbf{y} \cap X|}{|\mathbf{y} \setminus X|} \geq \frac{|X \cap \mathsf{Q}|}{|X \cup \mathsf{Q}|} = sim_J(X, \mathsf{Q})$, and because $\mathsf{Q} \in \mathbf{Opt}(X)$, $sim_J(X, \mathsf{Q}) \geq sim_J(X, \mathsf{O})$. But this means that $\frac{|\mathbf{y} \cap X|}{|\mathbf{y} \setminus X|} \geq \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$. However O also satisfies (2) and $\mathbf{y} \in \mathfrak{B}(X)$, so by (2), $\mathbf{y} \subseteq \mathsf{O}$, a contradiction. Hence $\mathsf{Q} \setminus \mathsf{O} = \emptyset$.

(*Case 2*) Let $\mathsf{O} \setminus \mathsf{Q} = \{\mathbf{y}_1, \ldots, \mathbf{y}_p\} \subseteq \mathfrak{B}(X)$. Let $|X \cap \mathsf{O}| = n$, $|X \cup \mathsf{O}| = m$, and $|\mathbf{y}_i \setminus X| = l_i$, $|\mathbf{y}_i \cap X| = k_i$, for $i = 1, \ldots, p$. Since O satisfies (2), for each $i = 1, \ldots, p$, we have $\frac{|\mathbf{y}_i \cap X|}{|\mathbf{y}_i \setminus X|} \geq \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$, or equivalently $\frac{k_i}{l_i} \geq \frac{n}{m}$. Hence $(k_1 + \ldots + k_p)m \geq (l_1 + \ldots + l_p)n$. On the other hand, $sim_J(X, \mathsf{Q}) = sim_J(X, \mathsf{O} \setminus (\mathbf{y}_1 \cup \ldots \cup \mathbf{y}_p)) > sim_J(X, \mathsf{O})$, so by Lemma 1, $\frac{|(\mathbf{y}_1 \cup \ldots \cup \mathbf{y}_p) \cap X|}{|(\mathbf{y}_1 \cup \ldots \cup \mathbf{y}_p) \setminus X|} < \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|}$. Because $\mathbf{y}_i$ are components, we have $\mathbf{y}_i \cap \mathbf{y}_j = \emptyset$ when $i \neq j$. Thus $|(\mathbf{y}_1 \cup \ldots \cup \mathbf{y}_p) \cap X| = |\mathbf{y}_1 \cap X| + \ldots + |\mathbf{y}_p \cap X| = k_1 + \ldots + k_p$, and $|(\mathbf{y}_1 \cup \ldots \cup \mathbf{y}_p) \setminus X| = |\mathbf{y}_1 \setminus X| + \ldots + |\mathbf{y}_p \setminus X| = l_1 + \ldots + l_p$. This means $\frac{|(\mathbf{y}_1 \cup \ldots \cup \mathbf{y}_p) \cap X|}{|(\mathbf{y}_1 \cup \ldots \cup \mathbf{y}_p) \setminus X|} < \frac{|X \cap \mathsf{O}|}{|X \cup \mathsf{O}|} \iff \frac{k_1 + \ldots + k_p}{l_1 + \ldots + l_p} < \frac{n}{m}$, which yields $(k_1 + \ldots + k_p)m < (l_1 + \ldots + l_p)n$, a contradiction, i.e. $\mathsf{O} \setminus \mathsf{Q} = \emptyset$. Thus, $\mathsf{Q} \setminus \mathsf{O} = \emptyset$ and $\mathsf{O} \setminus \mathsf{Q} = \emptyset$, i.e., $\mathsf{Q} = \mathsf{O}$, so $\mathsf{O} \in \mathbf{Opt}(X)$. □

Theorem 1 gives the necessary and sufficient conditions for optimal approximations (with respect to the Jaccard index) of X in terms of the elements of $\mathfrak{B}(X)$. We will use it to build an efficient algorithm for finding optimal approximations.

- Let $X \subseteq U$. For every element $\mathbf{x} \in \mathfrak{B}(X)$, we define an index $\alpha(\mathbf{x}) = \frac{|\mathbf{x} \cap X|}{|\mathbf{x} \setminus X|}$.

By Theorem 1, the value of $\alpha(\mathbf{x})$ will indicate if $\mathbf{x} \in \mathfrak{B}(X)$ is a part of an optimal approximation of X, or not. Since $\mathfrak{B}(X)$ is finite, its elements can be enumerated by natural numbers $1,\ldots,|\mathfrak{B}(X)|$.

- Assume that $r = |\mathfrak{B}(X)|$, $\mathfrak{B}(X) = \{\mathbf{x}_1,\ldots,\mathbf{x}_r\}$ and also $i \leq j \iff \alpha(\mathbf{x}_i) \geq \alpha(\mathbf{x}_j)$.

In other words, we sort $\mathfrak{B}(X)$ by decreasing values of $\alpha(\mathbf{x})$. We will use this sorting to build a special sequence of intermediate approximations.

Let $\mathsf{O}_0, \mathsf{O}_1, \ldots, \mathsf{O}_r \in \mathbf{IA}(X)$ be the sequence of intermediate approximations of X defined for $i = 0,\ldots,r-1$ as follows: $\mathsf{O}_0 = \underline{\mathbf{A}}(X)$ and

$$\mathsf{O}_{i+1} = \begin{cases} \mathsf{O}_i \cup \mathbf{x}_{i+1} & \text{if } sim_J(X, \mathsf{O}_i \cup \mathbf{x}_{i+1}) \geq sim_J(X, \mathsf{O}_i) \\ \mathsf{O}_i & \text{otherwise.} \end{cases}$$

We claim that at least one of these O_i's is an optimal approximation. The following technical result is needed to prove this claim.

Lemma 2. *Let $k_1,\ldots,k_n$ and $l_1,\ldots,l_n$ be positive numbers such that $\frac{k_1}{l_1} \geq \frac{k_i}{l_i}$ for $i = 1,\ldots,n$. Then $\frac{k_1}{l_1} \geq \frac{k_1+\ldots+k_n}{l_1+\ldots+l_n}$.*

Proof. $\frac{k_1}{l_1} \geq \frac{k_i}{l_i}$ implies $k_1 l_i \geq k_i l_1$ for $i = 1,\ldots,n$. Hence $k_1 l_1 + k_1 l_2 + \ldots + k_1 l_n \geq k_1 l_1 + k_2 l_1 + \ldots + k_n l_1 \iff \frac{k_1}{l_1} \geq \frac{k_1+\ldots+k_n}{l_1+\ldots+l_n}$, which ends the proof. □

The essential properties of the sequence $\mathsf{O}_0, \mathsf{O}_1, \ldots, \mathsf{O}_r$ are provided by the following theorem.

Theorem 2. *For every $X \subseteq U$, we set $r = |\mathfrak{B}(X)|$, and we have*

1. *$sim_J(X, \mathsf{O}_{i+1}) \geq sim_J(X, \mathsf{O}_i)$, for $i = 0,\ldots,r-1$.*
2. *If $\alpha(\mathbf{x}_1) \leq sim_J(X, \underline{\mathbf{A}}(X))$ then $\underline{\mathbf{A}}(X) \in \mathbf{Opt}(X)$.*
3. *If $\alpha(\mathbf{x}_r) \geq sim_J(X, \overline{\mathbf{A}}(X))$ then $\overline{\mathbf{A}}(X) \in \mathbf{Opt}(X)$.*
4. *If $sim_J(X, \mathsf{O}_p) \leq \alpha(\mathbf{x}_p)$ and $sim_J(X, \mathsf{O}_{p+1}) > \alpha(\mathbf{x}_{p+1})$, then $\mathsf{O}_p \in \mathbf{Opt}(X)$, for $p = 1,\ldots,r-1$.*
5. *If $\mathsf{O}_p \in \mathbf{Opt}(X)$, then $\mathsf{O}_i = \mathsf{O}_p$ for all $i = p+1,\ldots,r$. In particular $\mathsf{O}_r \in \mathbf{Opt}(X)$.*
6. *$\mathsf{O} \in \mathbf{Opt}(X) \implies \mathsf{O} \subseteq \mathsf{O}_p$, where p is the smallest one from (5).*

Proof. (1) Immediately from Lemma 1 and the definition of the sequence $\mathsf{O}_0,\ldots,\mathsf{O}_r$.
(2) From Proposition 1 we have that if $\mathsf{O} \in \mathbf{Opt}(X)$, then either $\mathsf{O} = \underline{\mathbf{A}}(X)$ or $\mathsf{O} = \underline{\mathbf{A}}(X) \cup \mathbf{x}_{i_1} \cup \ldots \cup \mathbf{x}_{i_s}$ for some $i_j \in \{1,\ldots,r\}$. Since $\alpha(\mathbf{x}_1) \geq \alpha(\mathbf{x}_{i_j})$ for $j = 1,\ldots,s$ by Lemma 2, $\alpha(\mathbf{x}_1) \geq \frac{|(\mathbf{x}_{i_1} \cup \ldots \cup \mathbf{x}_{i_s}) \cap X|}{|(\mathbf{x}_{i_1} \cup \ldots \cup \mathbf{x}_{i_s}) \setminus X|}$. Hence $sim_J(X, \underline{\mathbf{A}}(X)) \geq \frac{|(\mathbf{x}_{i_1} \cup \ldots \cup \mathbf{x}_{i_s}) \cap X|}{|(\mathbf{x}_{i_1} \cup \ldots \cup \mathbf{x}_{i_s}) \setminus X|}$, so by Lemma 1, $sim_J(X, \underline{\mathbf{A}}(X)) \geq sim_J(X, \mathsf{O})$, which means $\underline{\mathbf{A}}(X) \in \mathbf{Opt}(X)$.
(3) Note that $\alpha(\mathbf{x}_r) \geq sim_J(X, \overline{\mathbf{A}}(X))$ implies $\alpha(\mathbf{x}_i) \geq sim_J(X, \overline{\mathbf{A}}(X))$ for all $i = 1,\ldots,r$. Hence by Theorem 1, $\overline{\mathbf{A}}(X) \in \mathbf{Opt}(X)$.
(4) and (5) Since $sim_J(X, \mathsf{O}_0) \leq sim_J(X, \mathsf{O}_1) \leq \ldots \leq sim_J(X, \mathsf{O}_r)$ and $\alpha(\mathbf{x}_1) \geq \alpha(\mathbf{x}_2) \geq \ldots \geq \alpha(\mathbf{x}_r)$, then $\mathsf{O}_i = \mathsf{O}_p$ for all $i = p+1,\ldots,r$. Moreover $\mathsf{O}_p = \underline{\mathbf{A}}(X) \cup \mathbf{x}_1 \cup \ldots \cup \mathbf{x}_p$ satisfies (2) of Theorem 1, so $\mathsf{O}_p \in \mathbf{Opt}(X)$.
(6) We have to show that if $\mathsf{O} = \underline{\mathbf{A}}(X) \cup \mathsf{A} \in \mathbf{Opt}(X)$, where $\mathsf{A} \in \mathbb{B}(X)$, then $\mathsf{A} \subseteq \mathbf{x}_1 \cup \ldots \cup \mathbf{x}_p$. Suppose $\mathbf{x}_j \subseteq \mathsf{A}$ and $j > p$. Then $\alpha(\mathbf{x}_j) < sim_J(X, \mathsf{O}_p) = sim_J(X, \mathsf{O})$, so O does not satisfy (2) of Theorem 1. Hence $\mathsf{A} \subseteq \mathbf{x}_1 \cup \ldots \cup \mathbf{x}_p$. □

Point (1) of Theorem 2 states that O_{i+1} is a better (or equal) approximation of X than O_i, (2) and (3) characterize the case when either $\underline{\mathbf{A}}(X)$ or $\overline{\mathbf{A}}(X)$ are optimal approximations, while (4) shows conditions when some O_p is an optimal approximation. Point (5) states that once O_p is found to be optimal, we may stop calculations as the remaining O_{p+i} are the same as O_p, and the last point, (6) indicates that O_p is the greatest optimal approximation.

Algorithm 1 (Finding the Greatest Optimal Approximation) *Let $X \subseteq U$.*

1. *Construct $\underline{\mathbf{A}}(X)$, $\overline{\mathbf{A}}(X)$, and $\mathfrak{B}(X)$. Assume $r = |\mathfrak{B}(X)|$.*
2. *For each $\mathbf{x} \in \mathfrak{B}(X)$, calculate $\alpha(\mathbf{x}) = \frac{|\mathbf{x} \cap X|}{|\mathbf{x} \setminus X|}$.*
3. *Order $\alpha(\mathbf{x})$ in decreasing order and number the elements of $\mathfrak{B}(X)$ by this order, so $\mathfrak{B}(X) = \{\mathbf{x}_1, \ldots, \mathbf{x}_r\}$ and $i \leq j \iff \alpha(\mathbf{x}_i) \geq \alpha(\mathbf{x}_j)$.*
4. *If $\alpha(\mathbf{x}_1) \leq sim_J(X, \underline{\mathbf{A}}(X))$ then $\mathsf{O} = \underline{\mathbf{A}}(X)$.*
5. *If $\alpha(\mathbf{x}_r) \geq sim_J(X, \overline{\mathbf{A}}(X))$ then $\mathsf{O} = \overline{\mathbf{A}}(X)$.*
6. *Calculate O_i from $i = 0$ until $sim_J(X, \mathsf{O}_{p+1}) > \alpha(\mathbf{x}_{p+1})$, for $p = 0, \ldots, r-1$, and set $\mathsf{O} = \mathsf{O}_p$.*

From Theorem 2 we have that O is the greatest optimal approximation, i.e. $\mathsf{O} \in \mathbf{Opt}(X)$, and for all $\mathsf{O}' \in \mathbf{Opt}(X)$, $\mathsf{O}' \subseteq \mathsf{O}$. We also know that $sim_J(X, \mathsf{O}') = sim_J(X, \mathsf{O})$ □

This *greedy* algorithm (because of the choice of $\alpha(\mathbf{x})$, c.f. [18]) has a complexity of $C + \mathrm{O}(r \log r)$, where C is the complexity of constructing $\underline{\mathbf{A}}(X)$, $\overline{\mathbf{A}}(X)$, and $\mathfrak{B}(X)$. Algorithms with $C = \mathrm{O}(|U|^2)$ can be found for example in [12].

The most crucial line of the algorithm, line (6), runs in $\mathrm{O}(r)$, but line (3) involves sorting which has complexity $\mathrm{O}(r \log r)$. Since $r < |U|$, the total complexity is $\mathrm{O}(|U|^2)$.

Algorithm 1 gives us the greatest optimal approximation O, however the whole set $\mathbf{Opt}(X)$ can easily be derived from O just by subtracting appropriate elements of $\mathfrak{B}(X)$.

Note that because of Corollary 4(1), Algorithm 1 is also effective for *any* similarity measure *sim* that is *consistent* with the Jaccard index sim_J.

Let us now consider a simple example.

Example 1. We define our universe of elements $U = \{h_1, \ldots, h_{12}\}$ to be an assortment of houses, each with a price or value associated with it, as shown in Table 1. Based on its price, each house belongs to a representative equivalence class as demonstrated in the second table. Our classes will be defined by each range of \$20,000, starting from \$280,000 and ending with \$400,000 (empty classes are excluded as $\emptyset \notin \mathfrak{Comp}$). We could say that all of the houses in each class are roughly equivalent in price.

Suppose we wish to select a subset which we are interested in. If houses $\mathsf{H} = \{h_1, h_3, h_8, h_9\}$ meet our requirements we could say that we have the financing available for each of the equivalence classes those houses belong to. Clearly $\underline{\mathbf{A}}(\mathsf{H}) = e_4$ and $\overline{\mathbf{A}}(\mathsf{H}) = e_1 \cup e_2 \cup e_3 \cup e_4$. Moreover, $\mathfrak{B}(\mathsf{H}) = \mathrm{comp}(\overline{\mathbf{A}}(\mathsf{H})) \setminus \mathrm{comp}(\underline{\mathbf{A}}(\mathsf{H})) = \{e_1, e_2, e_3\}$, and $\mathbf{IA}(\mathsf{H}) = \{\underline{\mathbf{A}}(\mathsf{H}), A_1, A_2, A_3, A_4, A_5, A_6, \overline{\mathbf{A}}(\mathsf{H})\}$ where $A_1 = e_1 \cup e_4$, $A_2 = e_2 \cup e_4$, $A_3 = e_3 \cup e_4$, $A_4 = e_1 \cup e_2 \cup e_4$, $A_5 = e_1 \cup e_3 \cup e_4$, and $A_6 = e_2 \cup e_3 \cup e_4$. We also have $sim_J(\mathsf{H}, \underline{\mathbf{A}}(\mathsf{H})) = \frac{|\mathsf{H} \cap \underline{\mathbf{A}}(\mathsf{H})|}{|\mathsf{H} \cup \underline{\mathbf{A}}(\mathsf{H})|} = \frac{1}{4}$, $sim_J(\mathsf{H}, \overline{\mathbf{A}}(\mathsf{H})) = \frac{|\mathsf{H} \cap \overline{\mathbf{A}}(\mathsf{H})|}{|\mathsf{H} \cup \overline{\mathbf{A}}(\mathsf{H})|} = \frac{2}{5}$, and $sim_J(\mathsf{H}, A_1) = \frac{2}{5}$,

Table 1. Pawlak's space of houses and their prices

House	Price ($)	Equiv. class
h_1	289,000	e_1
h_2	389,000	e_5
h_3	319,000	e_2
h_4	333,000	e_3
h_5	388,000	e_5
h_6	284,000	e_1
h_7	339,000	e_3
h_8	336,000	e_3
h_9	345,000	e_4
h_{10}	311,000	e_2
h_{11}	319,000	e_2
h_{12}	312,000	e_2

Class	Elements	Range ($)
e_1	h_1, h_6	280-299,999
e_2	$h_3, h_{10}, h_{11}, h_{12}$	300-319,999
e_3	h_4, h_7, h_8	320-339,999
e_4	h_9	340-359,999
		360-379,999
e_5	h_2, h_5	380-400,000

$sim_J(\mathsf{H}, A_2) = \frac{2}{7}$, $sim_J(\mathsf{H}, A_3) = \frac{1}{3}$, $sim_J(\mathsf{H}, A_4) = \frac{3}{8}$, $sim_J(\mathsf{H}, A_5) = \frac{3}{7}$, $sim_J(\mathsf{H}, A_6) = \frac{2}{7}$. From all these Jaccard indexes, $\frac{3}{7}$ is the biggest number, so $\mathbf{Opt}(\mathsf{H}) = \{A_5\} = \{e_1 \cup e_3 \cup e_4\}$.

What about our algorithm? We have $\mathfrak{B}(\mathsf{H}) = \{e_1, e_2, e_3\}$, and $\alpha(e_1) = 1$, $\alpha(e_2) = \frac{1}{3}$, and $\alpha(e_3) = \frac{1}{2}$. Hence $\alpha(e_1) > \alpha(e_3) > \alpha(e_2)$, so we rename the elements of $\mathfrak{B}(\mathsf{H})$ as $e_1 = \mathbf{x}_1$, $e_3 = \mathbf{x}_2$, $e_2 = \mathbf{x}_3$. Clearly $\alpha(\mathbf{x}_1) = 1 > sim_J(\mathsf{H}, \underline{\mathbf{A}}(\mathsf{H})) = \frac{1}{4}$ and $\alpha(\mathbf{x}_3) = \frac{1}{3} < sim_J(\mathsf{H}, \overline{\mathbf{A}}(\mathsf{H})) = \frac{2}{5}$, so neither step (4) nor (5) hold, so we go to the step (6), which is the most involved. We begin by setting $\mathsf{O}_0 = \underline{\mathbf{A}}(\mathsf{H}) = e_4$. Since $sim_J(\mathsf{H}, \mathsf{O}_0) = \frac{1}{4} < sim_J(\mathsf{H}, \mathsf{O}_0 \cup \mathbf{x}_1) = \frac{2}{5}$, we have $\mathsf{O}_1 = \mathsf{O}_0 \cup \mathbf{x}_1 = e_1 \cup e_4$, and since $sim_J(\mathsf{H}, \mathsf{O}_1) = \frac{2}{5} < sim_J(\mathsf{H}, \mathsf{O}_1 \cup \mathbf{x}_2) = \frac{3}{7}$, we have $\mathsf{O}_2 = \mathsf{O}_1 \cup \mathbf{x}_2 = e_1 \cup e_3 \cup e_4$. However $sim_J(\mathsf{H}, \mathsf{O}_2) = \frac{3}{7} < \alpha(\mathbf{x}_2) = \frac{1}{2}$, so $\mathsf{O}_1 \notin \mathbf{Opt}(\mathsf{H})$. Since $sim_J(\mathsf{H}, \mathsf{O}_2) = \frac{3}{7} > sim_J(\mathsf{H}, \mathsf{O}_2 \cup \mathbf{x}_3) = \frac{2}{5}$, we set $\mathsf{O}_3 = \mathsf{O}_2$. Now we have $sim_J(\mathsf{H}, \mathsf{O}_3) = sim_J(\mathsf{H}, \mathsf{O}_2) = \frac{3}{7} > \alpha(\mathbf{x}_3) = \frac{1}{3}$, which means that $\mathsf{O}_2 = \{h_1, h_4, h_6, h_7, h_8, h_9\} \in \mathbf{Opt}(\mathsf{H})$. Note also that $\mathsf{O}_1 = A_1$, and $\mathsf{O}_2 = A_5$, and $\mathbf{Opt}(\mathsf{H}) = \{O_2\}$. □

5 Optimal Approximations with Dice-Sørensen Similarity Measure

The Jaccard measure is only one of many possible similarity measures. If it is found for example, that the Jaccard index under-represents the common elements, the Dice-Sørensen index can be used instead [5,6]. We will show that the Dice-Sørensen index and the Jaccard index are consistent, so we can use Algorithm 1 for the former as well, however we will start with a slightly more general result.

Lemma 3 (Partial consistency of Jaccard and Tversky indexes). *If $a = b > 0$, then for all $A, B, C \subseteq U$,*

$$sim_J(A,B) < sim_J(A,C) \iff sim_T^{a,b}(A,B) < sim_T^{a,b}(A,C).$$

Proof. If $A = C$ then $sim_J(A,C) = sim_T^{a,b}(A,C) = 1$, so the equivalence holds. Assume $A \neq C$. Since $sim_J(A,C) > 0$, then $A \cap C \neq \emptyset$. Moreover $A \setminus C \neq \emptyset$ or $C \setminus A \neq \emptyset$. Hence: $sim_J(A,B) < sim_J(A,C) \iff \frac{|A\cap B|}{|A\cup B|} < \frac{|A\cap C|}{|A\cup C|} \iff \frac{|A\cap B|}{|A\cap B|+|A\setminus B|+|B\setminus A|} < \frac{|A\cap C|}{|A\cap C|+|A\setminus C|+|C\setminus A|}$ $\iff |A\cap B|(|A\setminus C|+|C\setminus A|) < |A\cap C|(|A\setminus B|+|B\setminus A|) \iff \frac{|A\cap B|}{|A\cap C|} < \frac{|A\setminus B|+|B\setminus A|}{|A\setminus C|+|C\setminus A|} \iff$ $\frac{|A\cap B|}{|A\cap C|} < \frac{a|A\setminus B|+a|B\setminus A|}{a|A\setminus C|+a|C\setminus A|} \iff \frac{|A\cap B|}{|A\cap B|+a|A\setminus B|+a|B\setminus A|} < \frac{|A\cap C|}{|A\cap C|+a|A\setminus C|+a|C\setminus A|} \iff sim_T^{a,a}(A,B) <$ $sim_T^{a,a}(A,C)$. □

The above Lemma immediately implies that the Dice-Sørensen and Jaccard indexes are consistent.

Corollary 6 (Consistency of Jaccard and Dice-Sørensen indexes). *For all* A,B, $C \subseteq U$,

$$sim_J(A,B) < sim_J(A,C) \iff sim_{DS}(A,B) < sim_{DS}(A,C).$$

Proof. Since $sim_{DS}(A,B) = sim_T^{a,b}(A,B)$ with $a = b = 0.5$. □

For the case from Example 1 we have $\mathbf{Opt}_{sim_J}(\mathsf{H}) = \mathbf{Opt}_{sim_{DS}}(\mathsf{H}) = \{O_2\}$, where $\mathsf{O}_2 = \{h_1,h_4,h_6,h_7,h_8,h_9\}$, and $sim_J(\mathsf{H},\mathsf{O}_2) = \frac{3}{7}$, while $sim_{DS}(\mathsf{H},\mathsf{O}_2) = \frac{3}{5}$.

We have used the Jaccard index as our base for the design of Algorithm 1, as we found it more intuitive for a general, domain independent, case; however, since the Dice-Sørensen index does not have $|A \cup B|$, only $|A| + |B|$, it might actually be easier to use it for proving more sophisticated properties in future.

Neither the Fuzzy Sets index nor the Tversky index with $a \neq b$ are consistent with the Jaccard index. To show that the Fuzzy Sets index and the Jaccard index are inconsistent, consider the case of $A = \{a_1,a_2,a_3,a_4\}$, $B = \{a_1,a_2,a_3,a_5,...,a_{21}\}$, and $C = \{a_1,a_4,a_{22},...,a_{32}\}$. We have here $|A| = 4, |B| = 20, |C| = 13, |A \cap B| = 3$ and $|A \cap C| = 2$. Hence $sim_J(A,B) = \frac{4}{21} > sim_J(A,C) = \frac{2}{15}$, while $sim_{FS}(A,B) = \frac{3}{20} < sim_{FS}(A,C) = \frac{2}{13}$.

For the Tversky index consider $A = \{a_1,a_2,a_3,a_4\}$, $B = \{a_1,a_2,a_3,a_4,a_6,...,a_{12}\}$, and $C = \{a_3,a_4,a_5\}$. In this case $sim_J(A,B) = \frac{2}{11} < sim_J(A,C) = \frac{1}{5}$, but for any a and b such that $\frac{a}{b} > \frac{5}{4}$, we have $sim_T^{a,b}(A,B) > sim_T^{a,c}(A,C)$. For example for $a = 1.5$ and $b = 1.0$ we have $sim_T^{a,b}(A,B) = \frac{1}{6} > sim_T^{a,b}(A,C) = \frac{2}{13}$.

6 Final Comments

In the above we have proposed a novel approach to rough set approximation. In addition to lower and upper approximations, we introduced and analyzed the concept of *optimal* approximation, which required the concept of a similarity measure, and a notion of border and border sets. We provided five simple similarity measure axioms, and then referenced four measures of similarity which satisfy them. Only the Jaccard index [4] however, can naturally be interpreted as a measure of distance as well, so with this in mind, we used the index to design an algorithm which accepts a non-empty universe of elements (with an equivalence relation) and a subset $X \subseteq U$, and returns the optimal approximation. The algorithm runs in $\mathsf{O}(r\log r)$ time where r is the number of elements in the 'border set', and thus has total time complexity $\mathsf{O}(|U|^2)$.

We also introduced the concept of consistent similarity measures. Since consistent similarity indexes have identical optimal approximations, many results obtained for one index can be applied to all consistent indexes. As the Jaccard and the Dice-Sørensen indexes are consistent, all results of this paper hold for both indexes.

It appears that most of the results of Section 4 could be generalized beyond the class of similarity measures consistent with the Jaccard index, to some bigger class of generic similarity measures satisfying some additional axioms (note that here one argument of *sim* is always a definable set, i.e. a subset of $\mathbb{D}$).

Other particular similarity indexes that are not consistent with the Jaccard index, especially the Tversky index [16] are also worth analyzing in detail.

References

1. Pawlak, Z.: Rought Sets. International Journal of Computer and Information Sciences 34, 557–590 (1982)
2. Pawlak, Z.: Rough Sets. Kluwer, Dordrecht (1991)
3. Bretcher, O.: Linear Algebra with Applications. Prentice Hall (1995)
4. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturalles 37, 547–549 (1901)
5. Dice, L.R.: Measures of the Amount of Ecologic Association Between Species. Ecology 26(3), 297–302 (1945)
6. Sørensen, T.: A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application ro analysis of the vegetation on Danish commons. Biologishe Skrifter 5(4), 1–34 (1957)
7. Janicki, R.: Property-Driven Rough Sets Approximations of Relations. In: [9], pp. 333–357
8. Skowron, A., Stepaniuk, J.: Tolarence approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
9. Skowron, A., Suraj, Z. (eds.): Rough Sets and Intelligent Systems. ISRL, vol. 42. Springer, Heidelberg (2013)
10. Yao, Y.Y., Wang, T.: On Rough Relations: An Alternative Formulation. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 82–91. Springer, Heidelberg (1999)
11. Janicki, R.: Approximations of Arbitrary Binary Relations by Partial Orders: Classical and Rough Set Models. In: Peters, J.F., Skowron, A., Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) Transactions on Rough Sets XIII. LNCS, vol. 6499, pp. 17–38. Springer, Heidelberg (2011)
12. Saquer, J., Deogun, J.S.: Concept approximations based on Rough sets and similarity measures. Int. J. Appl. Math. Comput. Sci. 11(3), 655–674 (2001)
13. Słowiński, R., Vanderpooten, D.: A Generalized Definition of Rough Approximations Based on Similarity. IEEE Tran. on Knowledge and Data Engineering 12(2), 331–336 (2000)
14. Düntsch, I., Gediga, G., Lenarcic, A.: Affordance Relations. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS, vol. 5908, pp. 1–11. Springer, Heidelberg (2009)
15. Rezai, H., Emoto, M., Mukaidono, M.: New similarity Measure Between Two Fuzzy Sets. Journal of Advanced Computational Intelligence and Intelligent Informatics 10(6), 946–953 (2006)
16. Tversky, A.: Features of similarity. Psychological Reviews 84(4), 327–352 (1977)
17. Halmos, P.: Measure Theory, Van Nostrand (1950)
18. Kleinberg, J., Tardos, E.: Algorithm Design. Addison-Weslay (2005)

Partial Approximation of Multisets and Its Applications in Membrane Computing

Tamás Mihálydeák[1] and Zoltán Ernő Csajbók[2]

[1] Department of Computer Science, Faculty of Informatics, University of Debrecen
Kassai út 26, H-4028 Debrecen, Hungary
mihalydeak.tamas@inf.unideb.hu
[2] Department of Health Informatics, Faculty of Health, University of Debrecen,
Sóstói út 2-4, H-4400 Nyíregyháza, Hungary
csajbok.zoltan@foh.unideb.hu

Abstract. Partial nature of real–life problems requires working out partial approximation schemes. Partial approximation of sets is based on classical set theory. Its generalization for multisets gives a plausible opportunity to introduce an abstract concept of "to be close enough to a membrane" in membrane computing. The paper presents important features of general (maybe partial) multiset approximation spaces, their lattice theory properties, and shows how partial multiset approximation spaces can be applied to membrane computing.

Keywords: Rough set theory, multiset theory, partial approximation of multisets, lattice theory, membrane computing.

1 Introduction

Studies of set approximations were originally invented by Pawlak in the early 1980's [1, 2]. There are many different generalizations of classical Pawlakian rough set theory, among others, for multisets. A possible approach may rely on equivalence multiset relations [3], or general multirelations [4].

Partial nature of real–life problems, however, requires working out partial approximation schemes. The framework called the partial approximation of sets [5, 6] is based on classical set theory similarly to rough set theory. It was generalized for multisets [7, 8] in connection with membrane computing introduced by Păun in 2000 [9–11]. Membrane computing was motivated by biological and chemical processes in which an object has to be close enough to a membrane in order to be able to pass through it. Looking at regions as multisets, partial approximation of multisets gives a plausible opportunity to introduce the abstract, not necessarily space–like, concept of "to be close enough to a membrane". The paper presents the most important features of partial multiset approximation spaces, their lattice theory properties and applications to membrane computing.

The paper is organized as follows. Having reviewed the fundamental notions of multiset theory, Section 3 presents the concept of general multiset approximation space. Section 4 shows its generalized Pawlakian variant which is applied to membrane computing in Section 5.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 99–108, 2013.

2 Fundamental Notions of Multiset Theory

Let U be a finite nonempty set. A *multiset* M, or *mset* M for short, over U is a mapping $M : U \to \mathbb{N} \cup \{\infty\}$, where $\mathbb{N}$ is the set of natural numbers. If $M(a) \neq 0$, it is said that a belongs to M, otherwise a does not belong to M. The set $M^* = \{a \in U \mid M(a) \neq 0\}$ is called the *support* of M.

The mset M is the *empty mset*, denoted by $\emptyset$ if $M^* = \emptyset$. An mset M is *finite* if $M(a) < \infty$ for all $a \in M^*$.

Let $\mathcal{MS}(U)$ denote the set of all msets over U.

Basic set–theoretical relations can be generalized for msets as follows.

Definition 1. *Let M, M_1, M_2 be msets over U.*

1. Multiplicity *relation for an mset M over U is: $a \in M$ $(a \in U)$ if $M(a) \geq 1$.*
2. *Let $n \in \mathbb{N}^+$ be a positive integer.* n–times multiplicity *relation for an mset M over U is the following: $a \in^n M$ $(a \in U)$ if $M(a) = n$.*
3. *$M_1 = M_2$ if $M_1(a) = M_2(a)$ for all $a \in U$ (mset* equality *relation).*
4. *$M_1 \sqsubseteq M_2$ if $M_1(a) \leq M_2(a)$ for all $a \in U$ (mset* inclusion *relation).*

The next definitions give the generalizations for msets of the basic set–theoretical operations.

Definition 2. *Let $M, M_1, M_2 \in \mathcal{MS}(U)$ be msets over U and $\mathcal{M} \subseteq \mathcal{MS}(U)$ be a set of msets over U.*

1. *$(M_1 \sqcap M_2)(a) = \mathsf{min}\{M_1(a), M_2(a)\}$ for all $a \in U$ (*intersection*).*
2. *$(\bigsqcap \mathcal{M})(a) = \mathsf{min}\{M(a) \mid M \in \mathcal{M}\}$ for all $a \in U$.*
3. *$(M_1 \sqcup M_2)(a) = \mathsf{max}\{M_1(a), M_2(a)\}$ for all $a \in U$ (*set–type union*).*
4. *$(\bigsqcup \mathcal{M})(a) = \mathsf{sup}\{M(a) \mid M \in \mathcal{M}\}$ for all $a \in U$. By definition, $\bigsqcup \emptyset = \emptyset$.*
5. *$(M_1 \oplus M_2)(a) = M_1(a) + M_2(a)$ for all $a \in U$ (*mset addition*).*
6. *For any $n \in \mathbb{N}$,* n-times addition *of M, denoted by $\oplus_n M$, is given by the following inductive definition:*
   (a) $\oplus_0 M = \emptyset$;
   (b) $\oplus_1 M = M$;
   (c) $\oplus_{n+1} M = \oplus_n M \oplus M$.
7. *$(M_1 \ominus M_2)(a) = \mathsf{max}\{M_1(a) - M_2(a), 0\}$ for all $a \in U$ (*mset subtraction*).*

By the n-times addition, the *n-times inclusion* relation ($\sqsubseteq^n$) can be defined.

Definition 3. *Let $M_1 \neq \emptyset, M_2$ be two msets over U.*
For any $n \in \mathbb{N}$, $M_1 \sqsubseteq^n M_2$ if $\oplus_n M_1 \sqsubseteq M_2$ but $\oplus_{n+1} M_1 \not\sqsubseteq M_2$.

Corollary 1. *Let $M_1 \neq \emptyset, M_2$ be two msets over U and $n \in \mathbb{N}$.*

1. *$M_1 \sqsubseteq^n M_2$ if and only if $nM_1(a) \leq M_2(a)$ for all $a \in U$ and there is an $a' \in U$ such that $(n+1)M_1(a') > M_2(a')$.*
2. *$M_1 \sqsubseteq^0 M_2$ if and only if $M_1 \not\sqsubseteq M_2$.*
3. *For all $n \in \mathbb{N}^+$, $M_1 \sqsubseteq^n M_2$ if and only if $\oplus_n M_1 \sqsubseteq^1 M_2$.*

3 Some Lattice Theory Properties of Set of Multisets

The next proposition is an immediate consequence of Definition 1 and 2 (for the lattice theory notions, see, e.g., [12–14]).

Proposition 1. *$\langle \mathcal{MS}(U), \sqcap, \sqcup \rangle$ is a complete lattice, that is*

1. *(a) operations $\sqcup$ and $\sqcap$ are idempotent, commutative and associative;*
   (b) operations $\sqcup$ and $\sqcap$ fulfill the absorption laws for all $M_1, M_2 \in \mathcal{MS}(U)$:
   $$M_1 \sqcap (M_1 \sqcup M_2) = M_1 \text{ and } M_1 \sqcup (M_1 \sqcap M_2) = M_1;$$
2. *$\bigsqcup \mathcal{M}$ and $\bigsqcap \mathcal{M}$ exist for every $\mathcal{M} \subseteq \mathcal{MS}(U)$.*

In addition, $\langle \mathcal{MS}(U), \sqsubseteq \rangle$ is a partially ordered set in which $M_1 \sqsubseteq M_2$ if and only if $M_1 \sqcup M_2 = M_2$, or equivalently, $M_1 \sqcap M_2 = M_1$ for all $M_1, M_2 \in \mathcal{MS}(U)$.

A set $\mathcal{M}$ of finite msets over U is called a *macroset* $\mathcal{M}$ over U [15].
We define the following two fundamental macrosets:

1. $\mathcal{MS}^n(U)$ $(n \in \mathbb{N})$ is the set of all msets M over U such that $M(a) \leq n$ for all $a \in U$, and
2. $\mathcal{MS}^{<\infty}(U) = \bigcup_{n=0}^{\infty} \mathcal{MS}^n(U)$.

Note that $\mathcal{MS}^0(U) = \emptyset$ and $\mathcal{MS}^n(U) \subsetneqq \mathcal{MS}^{n+1}(U)$ $(n = 0, 1, 2, \dots)$. Moreover, $\mathcal{MS}^n(U)$ $(n \in \mathbb{N})$ is finite and $\mathcal{MS}^{<\infty}(U)$ is countably infinite.

$M_1 \sqcup M_2, M_1 \sqcap M_2 \in \mathcal{MS}^n(U)$ $(M_1, M_2 \in \mathcal{MS}^n(U))$ and the finiteness of $\mathcal{MS}^n(U)$ immediately imply that $\langle \mathcal{MS}^n(U), \sqcup, \sqcap \rangle$ $(n \in \mathbb{N}^+)$ is a complete sublattice of the lattice $\langle \mathcal{MS}(U), \sqcup, \sqcap \rangle$. Its top element is the mset M such that $M^* = U$, $M(a) = n$ $(a \in U)$, and its bottom element is the empty mset $\emptyset$.

$\langle \mathcal{MS}^{<\infty}(U), \sqcup, \sqcap \rangle$ is also a sublattice of the lattice $\langle \mathcal{MS}(U), \sqcup, \sqcap \rangle$. However, it is not a complete lattice since it lacks a top element. Nevertheless, $\langle \mathcal{MS}^{<\infty}(U), \sqsubseteq \rangle$ is a meet-semilattice such that $\bigsqcap \mathcal{M}$ exists in $\mathcal{MS}^{<\infty}(U)$ for every nonempty $\mathcal{M} \subseteq \mathcal{MS}^{<\infty}(U)$. Consequently, $\bigsqcup \mathcal{M}$ exists in $\mathcal{MS}^{<\infty}(U)$ for every subset $\mathcal{M} \subseteq \mathcal{MS}^{<\infty}(U)$ which has an upper bound in $\mathcal{MS}^{<\infty}(U)$, and

$$\bigsqcup \mathcal{M} = \bigsqcap \{M' \in \mathcal{MS}^{<\infty}(U) \mid \forall M \in \mathcal{M}(M \sqsubseteq M')\}.$$

4 General Multiset Approximation Spaces

A general mset approximation space has four components:

- a *domain* of the approximation space whose members are approximated;
- some distinguished members of the domain as the *basis* of approximations;
- *definable msets* deriving from base msets in some way as possible approximations of the members of the domain;
- an *approximation pair* determining the lower and upper approximations of the msets of the domain using definable msets.

Definable msets represent our available knowledge about the domain. They can be thought of as *tools*, in particular, base msets as *primary tools*, definable msets as *derived tools*. The way of getting derived tools from primary tools shows how primary tools are used. An approximation pair prescribes the *utilization* of primary and derived tools in a whole approximation process.

Definition 4. *The ordered 5–tuple* $\mathsf{MAS}(U) = \langle \mathcal{MS}^{<\infty}(U), \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathsf{l}, \mathsf{u} \rangle$ *is a* (general) mset approximation space *over* U *with the domain* $\mathcal{MS}^{<\infty}(U)$ *if*

1. $\mathfrak{B} \subseteq \mathcal{MS}^{<\infty}(U)$ *and if* $B \in \mathfrak{B}$*, then* $B \neq \emptyset$ *(in notation* $\mathfrak{B} = \{B_\gamma \mid \gamma \in \Gamma\}$*);* $\mathfrak{B}$ *is called the* base system*, its members are called the* base msets*;*
2. $\mathfrak{D}_{\mathfrak{B}} \subseteq \mathcal{MS}^{<\infty}(U)$ *is an extension of* $\mathfrak{B}$ *satisfying the following minimal requirement: if* $B \in \mathfrak{B}$*, then* $\oplus_n B \in \mathfrak{D}_{\mathfrak{B}}$ *for all* $n \in \mathbb{N}$*; members of* $\mathfrak{D}_{\mathfrak{B}}$ *are called* definable msets*;*
3. *the functions* $\mathsf{l}, \mathsf{u} : \mathcal{MS}^{<\infty}(U) \to \mathcal{MS}^{<\infty}(U)$ *(called* lower *and* upper approximation functions*) form a* weak approximation pair $\langle \mathsf{l}, \mathsf{u} \rangle$ *if*
   - *(C0)* $\mathsf{l}(\mathcal{MS}^{<\infty}(U)), \mathsf{u}(\mathcal{MS}^{<\infty}(U)) \subseteq \mathfrak{D}_{\mathfrak{B}}$ *(*definability *of* l, u*);*
   - *(C1) the functions* l *and* u *are monotone, i.e., for all* $M_1, M_2 \in \mathcal{MS}^{<\infty}(U)$ *if* $M_1 \sqsubseteq M_2$*, then* $\mathsf{l}(M_1) \sqsubseteq \mathsf{l}(M_2)$*,* $\mathsf{u}(M_1) \sqsubseteq \mathsf{u}(M_2)$ *(*monotonicity *of* l, u*);*
   - *(C2)* $\mathsf{u}(\emptyset) = \emptyset$ *(*normality *of* u*);*
   - *(C3) if* $M \in \mathcal{MS}^{<\infty}(U)$*, then* $\mathsf{l}(M) \sqsubseteq \mathsf{u}(M)$ *(*weak approximation property*).*

Corollary 2. $\mathsf{l}(\emptyset) = \emptyset$ *(*normality *of* l*).*

$\mathsf{MAS}(U)$ is *total* if for any $M \in \mathcal{MS}^{<\infty}(U)$ there is a definable mset $D \in \mathfrak{D}_{\mathfrak{B}}$ such that $M \sqsubseteq D$, and it is *partial* otherwise. If $\mathfrak{D}_{\mathfrak{B}}$ is the smallest set of msets satisfying condition 2 in Definition 4, $\mathsf{MAS}(U)$ is total if and only if there is a $B \in \mathfrak{B}$ such that $B(a) \geq 1$ for all $a \in U$.

There may be more than one msets with the same lower and upper approximations. If $M \in \mathcal{MS}^{<\infty}(U)$, the set

$$\mathcal{RM}(M) = \{M' \in \mathcal{MS}^{<\infty}(U) \mid \mathsf{l}(M) = \mathsf{l}(M') \text{ and } \mathsf{u}(M) = \mathsf{u}(M')\}$$

is called the *rough mset connected to* M.

Of course, l and u are neither additive nor multiplicative in general.

Proposition 2. *Let* $\mathsf{MAS}(U) = \langle \mathcal{MS}^{<\infty}(U), \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathsf{l}, \mathsf{u} \rangle$ *be a general mset approximation space over* U*. Then, for any* $M_1, M_2 \in \mathcal{MS}^{<\infty}(U)$*,*

1. $\mathsf{l}(M_1) \sqcup \mathsf{l}(M_2) \sqsubseteq \mathsf{l}(M_1 \sqcup M_2)$*,* $\mathsf{l}(M_1 \sqcap M_2) \sqsubseteq \mathsf{l}(M_1) \sqcap \mathsf{l}(M_2)$*,*
2. $\mathsf{u}(M_1) \sqcup \mathsf{u}(M_2) \sqsubseteq \mathsf{u}(M_1 \sqcup M_2)$*,* $\mathsf{u}(M_1 \sqcap M_2) \sqsubseteq \mathsf{u}(M_1) \sqcap \mathsf{u}(M_2)$*,*

i.e., lower and upper approximations are superadditive and submultiplicative.

Proof. $M_1, M_2 \sqsubseteq M_1 \sqcup M_2$ and $M_1 \sqcap M_2 \sqsubseteq M_1, M_2$, and so, by the monotonicity of l, $\mathsf{l}(M_1), \mathsf{l}(M_2) \sqsubseteq \mathsf{l}(M_1 \sqcup M_2)$ and $\mathsf{l}(M_1 \sqcap M_2) \sqsubseteq \mathsf{l}(M_1), \mathsf{l}(M_2)$, and the statement (1) immediately follows. Statement (2) can be proved similarly. □

It is reasonable to assume that the base msets and their n-times additions are exactly approximated from "lower side". In certain cases, it is also required of definable msets.

Definition 5. *A weak approximation pair* $\langle \mathsf{l}, \mathsf{u} \rangle$ *is*

(C4) granular *if* $B \in \mathfrak{B}$ *implies* $\mathsf{l}(\oplus_n B) = \oplus_n B$ *($n \in \mathbb{N}$) (in other words,* l *is granular),*
(C5) standard *if* $D \in \mathfrak{D}_{\mathfrak{B}}$ *implies* $\mathsf{l}(D) = D$ *(in other words,* l *is standard).*

Of course, if l is standard, the granularity of l also holds. The next proposition gives a necessary and sufficient condition that l is standard.

Proposition 3. *Let* $\mathsf{MAS}(U) = \langle \mathcal{MS}^{<\infty}(U), \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathsf{l}, \mathsf{u} \rangle$ *be a general mset approximation space over* U*.*

l *is standard if and only if* $\mathsf{l}(\mathcal{MS}^{<\infty}(U)) = \mathfrak{D}_{\mathfrak{B}}$ *and* l *is idempotent, i.e.,* $\forall M \in \mathcal{MS}^{<\infty}(U)\,(\mathsf{l}(\mathsf{l}(M)) = \mathsf{l}(M))$.

Proof. ($\Rightarrow$) By *(C0)*, $\mathsf{l}(\mathcal{MS}^{<\infty}(U)) \subseteq \mathfrak{D}_{\mathfrak{B}}$. On the other hand, for any $D \in \mathfrak{D}_{\mathfrak{B}}$, $\mathsf{l}(D) = D \in \mathsf{l}(\mathcal{MS}^{<\infty}(U))$, since l is standard, i.e., $\mathfrak{D}_{\mathfrak{B}} \subseteq \mathsf{l}(\mathcal{MS}^{<\infty}(U))$. Thus, $\mathsf{l}(\mathcal{MS}^{<\infty}(U)) = \mathfrak{D}_{\mathfrak{B}}$.

Further, let $M \in \mathcal{MS}^{<\infty}(U)$. $\mathsf{l}(M) \in \mathfrak{D}_{\mathfrak{B}}$ according to the condition *(C0)*, and so $\mathsf{l}(\mathsf{l}(M)) = \mathsf{l}(M)$, since l is standard.

($\Leftarrow$) Let $D \in \mathfrak{D}_{\mathfrak{B}}$. Since $\mathfrak{D}_{\mathfrak{B}} = \mathsf{l}(\mathcal{MS}^{<\infty}(U))$, there exists at least one $M \in \mathsf{l}(\mathcal{MS}^{<\infty}(U))$ such that $D = \mathsf{l}(M)$. l is idempotent, and so

$$\mathsf{l}(D) = \mathsf{l}(\mathsf{l}(M)) = \mathsf{l}(M) = D,$$

that is, l is standard. □

An important question is how lower and upper approximations relate to the approximated mset.

Definition 6. *A weak approximation pair* $\langle \mathsf{l}, \mathsf{u} \rangle$ *is*

(C6) lower semi–strong *if* $\mathsf{l}(M) \sqsubseteq M$ *($M \in \mathcal{MS}^{<\infty}(U)$) (*$\mathsf{l}$ *is* contractive*);*
(C7) upper semi–strong *if* $M \sqsubseteq \mathsf{u}(M)$ *($M \in \mathcal{MS}^{<\infty}(U)$) (*$\mathsf{u}$ *is* extensive*);*
(C8) strong *if it is lower and upper semi–strong simultaneously, i.e., each subset* $M \in \mathcal{MS}^{<\infty}(U)$ *is bounded by* $\mathsf{l}(M)$ *and* $\mathsf{u}(M)$*:* $\mathsf{l}(M) \sqsubseteq S \sqsubseteq \mathsf{u}(M)$.

Definition 7. *The general mset approximation space* $\mathsf{MAS}(U)$ *is a* weak/granular/standard/lower semi–strong/upper semi–strong/strong *mset approximation space if the approximation pair* $\langle \mathsf{l}, \mathsf{u} \rangle$ *is weak/granular/standard/lower semi–strong/upper semi–strong/strong, respectively.*

5 Generalized Pawlakian Multiset Approximation Spaces

It is a natural assumption that $\mathfrak{D}_{\mathfrak{B}}$ is obtained (derived) from $\mathfrak{B}$ by some sorts of set and/or mset type transformations (for the most important cases, see [8]). In this case, an mset approximation space is surely partial if there exists at least one object in U which does not belong to any base mset.

In order to build a generalized Pawlakian mset approximation space, first, we define $\mathfrak{D}_{\mathfrak{B}}$ as follows.

Definition 8. $\mathsf{MAS}(U)$ *is a* strictly set–union type *mset approximation space if* $\mathfrak{D}_{\mathfrak{B}}$ *is given by the following inductive definition:*

1. $\emptyset \in \mathfrak{D}_{\mathfrak{B}}$;
2. $\mathfrak{B} \subseteq \mathfrak{D}_{\mathfrak{B}}$;
3. *if* $\mathfrak{B}^{\oplus} = \{\oplus_n B \mid B \in \mathfrak{B},\ n = 1, 2, \dots\}$ *and* $\mathfrak{B}' \subseteq \mathfrak{B}^{\oplus}$, *then* $\bigsqcup \mathfrak{B}' \in \mathfrak{D}_{\mathfrak{B}}$.

In a general mset approximation space $\mathsf{MAS}(U)$, $\bigsqcup\{D' \in \mathfrak{D}_{\mathfrak{B}} \mid D' \sqsubseteq D\} \sqsubseteq D$. On the other hand, D is definable, and so $D \in \{D' \in \mathfrak{D}_{\mathfrak{B}} \mid D' \sqsubseteq D\}$, i.e., $D \sqsubseteq \bigsqcup\{D' \in \mathfrak{D}_{\mathfrak{B}} \mid D' \sqsubseteq D\}$ also holds. Thus,

$$D = \bigsqcup\{D' \in \mathfrak{D}_{\mathfrak{B}} \mid D' \sqsubseteq D\}.$$

This formula indicates set–union nature of definable sets which can be sharpened in strictly set–union type mset approximation spaces as follows.

Proposition 4. *Let* $\mathsf{MAS}(U) = \langle \mathcal{MS}^{<\infty}(U), \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathsf{I}, \mathsf{u}\rangle$ *be a strictly set–union type mset approximation space over* U.

1. *For any definable set* $D \in \mathfrak{D}_{\mathfrak{B}}$,
$$D = \bigsqcup\{\oplus_n B \mid n \in \mathbb{N}^+, B \in \mathfrak{B}, B \sqsubseteq^n D\}.$$
2. *If* $\mathsf{MAS}(U)$ *is also granular and lower semi–strong, for any* $M \in \mathcal{MS}^{<\infty}(U)$,
$$\mathsf{I}(M) = \bigsqcup\{\oplus_n B \mid n \in \mathbb{N}^+, B \in \mathfrak{B}, B \sqsubseteq^n M\}.$$

Proof.

1. Since $\mathsf{MAS}(U)$ is strictly set–union type, by Definition 8, there exists $\mathfrak{B}' \subseteq \mathfrak{B}^{\oplus}$ for any $D' \in \mathfrak{D}_{\mathfrak{B}}$ such that $D' = \bigsqcup \mathfrak{B}'$. Hence,
$$\begin{aligned} D &= \bigsqcup\{D' \in \mathfrak{D}_{\mathfrak{B}} \mid D' \sqsubseteq D\} \\ &= \bigsqcup\{\oplus_n B \mid n \in \mathbb{N}^+, B \in \mathfrak{B}, \oplus_n B \sqsubseteq D\} \\ &= \bigsqcup\{\oplus_n B \mid n \in \mathbb{N}^+, B \in \mathfrak{B}, B \sqsubseteq^n D\}. \end{aligned}$$
2. By Corollary 1(3), $B \sqsubseteq^n M$ if and only if $\oplus_n B \sqsubseteq^1 M$ $(n \in \mathbb{N}^+)$. Thus, for any $n \in \mathbb{N}^+$ and $\oplus_n B \sqsubseteq^1 M$ $(B \in \mathfrak{B})$, the granularity and the monotone property of I imply that $\oplus_n B = \mathsf{I}(\oplus_n B) \sqsubseteq \mathsf{I}(M)$, therefore
$$\bigsqcup\{\oplus_n B \mid n \in \mathbb{N}^+, B \in \mathfrak{B}, B \sqsubseteq^n M\} \sqsubseteq \mathsf{I}(M).$$
On the other hand, $\mathsf{I}(M) \in \mathfrak{D}_{\mathfrak{B}}$ and so by Proposition 4(1), and since I is contractive, we obtain
$$\begin{aligned} \mathsf{I}(M) &= \bigsqcup\{\oplus_n B \mid n \in \mathbb{N}^+, B \in \mathfrak{B}, B \sqsubseteq^n \mathsf{I}(M)\} \\ &\sqsubseteq \bigsqcup\{\oplus_n B \mid n \in \mathbb{N}^+, B \in \mathfrak{B}, B \sqsubseteq^n M\}. \end{aligned}$$
Thus, $\mathsf{I}(M) = \bigsqcup\{\oplus_n B \mid n \in \mathbb{N}^+, B \in \mathfrak{B}, B \sqsubseteq^n M\}$. □

Next, we generalize the Pawlakian approximation pair for msets in strictly set–union type mset approximation spaces.

Definition 9. *Let* $\mathsf{MAS}(U) = \langle \mathcal{MS}^{<\infty}(U), \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathsf{l}, \mathsf{u} \rangle$ *be a strictly set–union type mset approximation space.*

The functions $\mathsf{l}, \mathsf{u} : \mathcal{MS}^{<\infty}(U) \to \mathcal{MS}^{<\infty}(U)$ *form a* (generalized) Pawlakian mset approximation pair $\langle \mathsf{l}, \mathsf{u} \rangle$ *if for any mset* $M \in \mathcal{MS}^{<\infty}(U)$,

1. $\mathsf{l}(M) = \bigsqcup\{\oplus_n B \mid n \in \mathbb{N}^+, B \in \mathfrak{B} \text{ and } B \sqsubseteq^n M\}$,
2. $\mathsf{b}(M) = \bigsqcup\{\oplus_n B \mid B \in \mathfrak{B}, B \not\sqsubseteq M,\ B \sqcap M \neq \emptyset \text{ and } B \sqcap M \sqsubseteq^n M\}$,
3. $\mathsf{u}(M) = \mathsf{l}(M) \sqcup \mathsf{b}(M)$,

where the function b *gives the* boundary *of mset* M.

It is easy to check the next proposition by Definition 9.

Proposition 5. *Let* $\mathsf{MAS}(U) = \langle \mathcal{MS}^{<\infty}(U), \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathsf{l}, \mathsf{u} \rangle$ *be a strictly set–union type mset approximation space with a Pawlakian mset approximation pair.*

Then $\mathsf{MAS}(U)$ *is a lower semi–strong mset approximation space and* l *is granular. In other words,* $\mathsf{MAS}(U)$ *fulfills the conditions (C0)–(C3), (C4), (C6).*

Definition 10. *A strictly set–union type approximation space with a Pawlakian mset approximation pair is called a* Pawlakian mset approximation space.

Proposition 6. *Let* $\mathsf{MAS}(U) = \langle \mathcal{MS}^{<\infty}(U), \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathsf{l}, \mathsf{u} \rangle$ *be a Pawlakian mset approximation space. Then*

$$\mathsf{u}(M) = (\mathsf{l}(M) \oplus \mathsf{b}(M)) \ominus (\mathsf{l}(M) \sqcap \mathsf{b}(M)).$$

Proof. For all $a \in U$,

$$\begin{aligned}
\mathsf{u}(M)(a) &= ((\mathsf{l}(M) \oplus \mathsf{b}(M)) \ominus (\mathsf{l}(M) \sqcap \mathsf{b}(M)))(a) \\
&= \mathsf{max}\{(\mathsf{l}(M) \oplus \mathsf{b}(M))(a) - (\mathsf{l}(M) \sqcap \mathsf{b}(M))(a), 0\} \\
&= \mathsf{max}\{\mathsf{l}(M)(a) + \mathsf{b}(M)(a) - \mathsf{min}\{\mathsf{l}(M)(a), \mathsf{b}(M)(a)\}, 0\} \\
&= \begin{cases} \mathsf{max}\{\mathsf{l}(M)(a), 0\}, & \text{if } \mathsf{l}(M)(a) \geq \mathsf{b}(M)(a); \\ \mathsf{max}\{\mathsf{b}(M)(a), 0\}, & \text{if } \mathsf{l}(M)(a) < \mathsf{b}(M)(a); \end{cases} \\
&= \mathsf{max}\{\mathsf{l}(M)(a), \mathsf{b}(M)(a)\} \\
&= (\mathsf{l}(M) \sqcup \mathsf{b}(M))(a).
\end{aligned}$$

□

6 Applications in Membrane Computing

In the membrane application we focus on hierarchical membrane systems with communication rules.

A membrane structure μ of degree m ($m \in \mathbb{N}^+$) is a rooted tree with m nodes. It can be represented by the set $R_\mu \subseteq \{1, \ldots, m\} \times \{1, \ldots, m\}$ where $\langle i, j \rangle \in R_\mu$ means that there is an edge from i (parent) to j (child) of the tree μ which is formulated by $\mathsf{parent}(j) = i$.

Let V be a finite alphabet. The tuple

$$\Pi = \langle V, \mu, w_1, w_2, \ldots, w_m, R_1, R_2, \ldots, R_m \rangle$$

is called a *membrane system* or *P system* if $w_i \in \mathcal{MS}^{<\infty}(V)$ is the *region* of Π, and R_i is a finite set of *rules* of the form symport and antiport ($i = 1, 2, \ldots, m$). For the precise definition, see [8], Definition 6.

If the P system $\Pi = \langle V, \mu, w_1, w_2, \ldots, w_m, R_1, R_2, \ldots, R_m \rangle$ is given, let $\mathsf{MAS}(\Pi) = \langle \mathcal{MS}^{<\infty}(V), \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathsf{l}, \mathsf{u} \rangle$ be a strictly set–union type mset approximation space with a generalized Pawlakian approximation pair $\langle \mathsf{l}, \mathsf{u} \rangle$. $\mathsf{MAS}(\Pi)$ is called a *joint membrane approximation space*.

Having given a membrane system Π and its joint membrane approximation space $\mathsf{MAS}(\Pi)$, we can define the boundaries of the regions $w_1, w_2, \ldots, w_m$ as msets with the help of approximative function b specified in Definition 9.

Definition 11. *Let $\Pi = \langle V, \mu, w_1, w_2, \ldots, w_m, R_1, R_2, \ldots, R_m \rangle$ be a P system and $\mathsf{MAS}(\Pi) = \langle \mathcal{MS}^{<\infty}(V), \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathsf{l}, \mathsf{u} \rangle$ be its joint membrane approximation space. If $B \in \mathfrak{B}$ and $i = 1, 2, \ldots, m$, let*

$$N(B,i) = \begin{cases} 0, & \text{if } B \sqsubseteq w_i \text{ or } B \sqcap w_i = \emptyset; \\ n, & \text{if } i = 1 \text{ and } B \sqcap w_1 \sqsubseteq^n w_1; \\ \min\{k, n \mid B \sqcap w_i \sqsubseteq^k w_i, \text{ and } B \ominus w_i \sqsubseteq^n w_{\mathsf{parent}(i)}\}, & \text{otherwise.} \end{cases}$$

Then, for $i = 1, \ldots, m$,

$\mathsf{bnd}(w_i) = \bigsqcup\{\oplus_{N(B,i)} B \mid B \in \mathfrak{B}\}$;
$\mathsf{bnd}^{\mathsf{out}}(w_i) = \mathsf{bnd}(w_i) \ominus w_i$;
$\mathsf{bnd}^{\mathsf{in}}(w_i) = \mathsf{bnd}(w_i) \ominus \mathsf{bnd}^{\mathsf{out}}(w_i)$.

The functions $\mathsf{bnd}(w_i)$, $\mathsf{bnd}^{\mathsf{out}}(w_i)$, $\mathsf{bnd}^{\mathsf{in}}(w_i)$ give *membrane boundaries*, *outside membrane boundaries* and *inside membrane boundaries*, respectively.

The general notion of boundaries given in Definition 9 cannot be used here, because membrane boundaries have to follow the given membrane structure μ. The Pawlakian lower approximations $\mathsf{l}(w_i)$ ($i = 1, \ldots, m$) surely obey the membrane structure, and the Pawlakian upper approximation $\mathsf{u}(w_1)$ and the boundary $\mathsf{b}(w_1)$ are completely within the environment of the membrane structure.

However, the Pawlakian upper approximation $\mathsf{u}(w_i)$, therefore the boundary $\mathsf{b}(w_i)$ ($i = 2, \ldots, m$) do not obey the membrane structure in general. Thus, the Pawlakian boundaries have to be adjusted to the membrane structure by the function bnd. Of course, $\mathsf{b}(w_1) = \mathsf{bnd}(w_1)$, but $\mathsf{b}(w_i) \neq \mathsf{bnd}(w_i)$ ($i = 2, \ldots, m$) in general. Moreover, membrane boundaries $\mathsf{bnd}(w_i)$ ($i = 1, \ldots, m$) are split into two parts, inside and outside membrane boundaries.

As an illustrative example for the membrane boundary, let us take a membrane structure with 1 node, and let the base system $\mathfrak{B}$ consist of three base msets: B_1, B_2, B_3. In the figures below, they are represented by circle, triangle, and square, respectively. For the sake of clarity, only a fragment of the whole mset approximation space is depicted focusing on the membrane boundary solely.

Fig. 1 shows the membrane boundary of the region w_1.

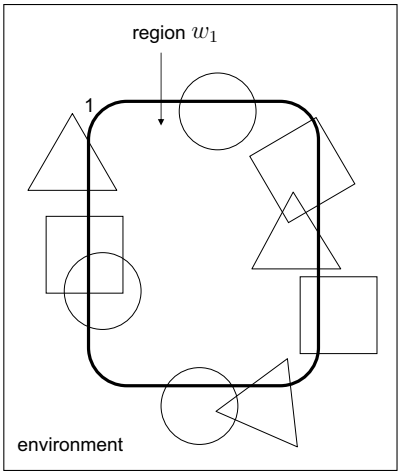


Fig. 1. A membrane boundary

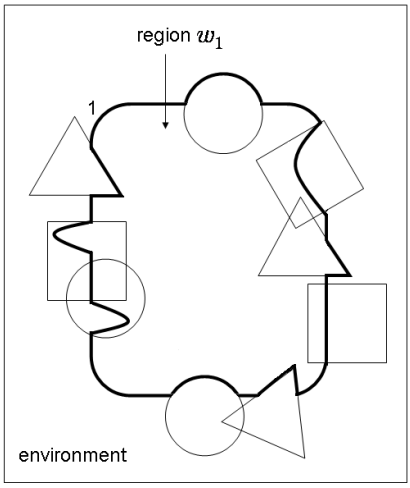


Fig. 2. The membrane boundary, after the membrane computation

Using membrane boundaries, the following constraints for rule executions are prescribed: a rule $r \in R_i$ of a membrane i has to work only in the boundaries of its region. It can be shown that the membrane computation actually works in the membrane boundaries ([8], Theorem 1). Fig. 2 illustrates the membrane boundary just after the membrane computation has halted.

In [8], the authors gave the pseudocode of the whole computation process as well.

7 Conclusion

In the paper, the authors have defined general multiset approximation spaces and have discussed their fundamental approximative properties. Their lattice theory properties have been shown as well. These properties hold not only in Pawlakian but also in general mset approximation spaces.

The importance of defined general multiset approximation spaces can be found, for instance, in their applications in membrane computing. By using the partial multiset approximation technique, the notion of "to be close enough to a membrane", even from inside and outside, has been specified in an abstract way. Thus, by constraining the communication rule executions on these abstract membrane boundaries, the membrane computation can be controlled.

Acknowledgements. 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. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
2. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
3. Girish, K.P., John, S.J.: Relations and functions in multiset context. Information Sciences 179(6), 758–768 (2009)
4. Grzymala-Busse, J.: Learning from examples based on rough multisets. In: Proceedings of the Second International Symposium on Methodologies for Intelligent Systems, pp. 325–332. North-Holland Publishing Co., Amsterdam (1987)
5. Csajbók, Z.E.: Approximation of sets based on partial covering. In: Peters, J.F., Skowron, A., Ramanna, S., Suraj, Z., Wang, X. (eds.) Transactions on Rough Sets XVI. LNCS, vol. 7736, pp. 144–220. Springer, Heidelberg (2013)
6. Csajbók, Z., Mihálydeák, T.: Partial approximative set theory: A generalization of the rough set theory. International Journal of Computer Information Systems and Industrial Management Applications 4, 437–444 (2012)
7. Mihálydeák, T., Csajbók, Z.: Membranes with local environments. In: Csuhaj-Varjú, E., Gheorghe, M., Vaszil, G. (eds.) Proceedings of the 13th International Conference on Membrane Computing, CMC13, Budapest, Hungary, August 28-31, pp. 311–322. MTA SZTAKI, The Computer and Automation Research Institute of the Hungarian Academy of Sciences (2012)
8. Mihálydeák, T., Csajbók, Z.E.: Membranes with boundaries. In: Csuhaj-Varjú, E., Gheorghe, M., Rozenberg, G., Salomaa, A., Vaszil, G. (eds.) CMC 2012. LNCS, vol. 7762, pp. 277–294. Springer, Heidelberg (2013)
9. Păun, G.: Computing with membranes. Journal of Computer and System Sciences 61(1), 108–143 (2000)
10. Păun, G.: Membrane Computing. An Introduction. Springer, Berlin (2002)
11. Păun, G., Rozenberg, G., Salomaa, A. (eds.): The Oxford Handbook of Membrane Computing. Oxford Handbooks. Oxford University Press, Inc., New York (2010)
12. Birkhoff, G.: Lattice theory, 3rd edn. Colloquium Publications, vol. 25. American Mathematical Society, Providence, Providence (1967)
13. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order, 2nd edn. Cambridge University Press, Cambridge (2002)
14. Grätzer, G.: General Lattice Theory. Birkhäuser Verlag, Basel und Stuttgart (1978)
15. Kudlek, M., Martín-Vide, C., Păun, G.: Toward a formal macroset theory. In: Calude, C.S., Păun, G., Rozenberg, G., Salomaa, A. (eds.) Multiset Processing. LNCS, vol. 2235, pp. 123–134. Springer, Heidelberg (2001)

A Formal Concept Analysis Based Approach to Minimal Value Reduction

Mei-Zheng Li[1,2], Guoyin Wang[1,2,3,⋆], and Jin Wang[2]

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China
[2] Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[3] Institute of Electronic Information Technology, Chongqing Institute of Green and Intelligent Technology, CAS, Chongqing 401122, China
llimeizhengfirst@my.swjtu.edu.cn, wanggy@ieee.org, wangjin@cqupt.edu.cn

Abstract. Reduction is a core issue in Rough Set Theory. Current reductions falls into 3 categories: tuple reduction, attribute reduction and value reduction. From the reduced tables, decision rules can be derived. For the purpose of storage and better understanding, minimization of the rule set is desired, and it is NP-hard. To tackle this problem, a heuristic approach to approximate minimal value reduct set is proposed based on Formal Concept Analysis in this paper. Experiments show that our approach is valid with a higher accuracy.

Keywords: value reduction, rule acquisition, rough set, formal concept analysis, positive hypotheses.

1 Introduction

Pioneered by Pawlak [14, 16] in 1982, Rough Set Theory (RST) has become a powerful mathematical tool for dealing with the vagueness and uncertainty inherent in various practical problems. Among its various applications, data mining catches most attentions. Reduction is a core issue in RST. It aims to reduce superfluous knowledge on the condition of maintaining the decision-making ability. Currently reductions can complemented from three aspects: row (tuple) reduction, column (attribute) reduction, and cell (value) reduction. Row reduction is simply to merge duplicate rows, attribute reduction is to find important attributes, and value reduction is to simplify decision rules. Many attribute reduction algorithms have been proposed [18]. The main approaches to value reductions falls into 4 categories: naive approaches [16, 17], heuristic approaches [17], matrix approaches [17, 22, 23], and inductive approaches [17]. These algorithms are mostly based on some attribute information.

In this paper, we propose an approach to attribute value reduction based on the extensions (objects). To minimize the obtained rule set, finding the rules that

⋆ Corresponding author.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 109–120, 2013.

can be supported by as many objects as possible is necessary, in other words, finding the common value reduct shared by as many objects as possible is needed. Formal Concept Analysis is a suitable tool to find the common attributes shared by some objects.

Formal Concept Analysis (FCA), proposed by Wille in 1982 [4, 20], is used as a knowledge representation mechanism and as a conceptual clustering method. Recently, researchers have payed much attention to data reduction with combination methods of RST and FCA, for example [8, **?**, 10, 11], but most of them focus on attribute reduction, and some heuristic methods are mostly based on the attribute information (for example, [8]), little research has been done on value reduct. In this paper, we try to obtain the value reduct rule set with the smallest size by using FCA. First, a decision table is transformed to a formal context, then the context is divided into some sub-contexts according to the decisions to obtain the maximal "consistent" concepts of every sub-context, i.e. the concepts who do not conflict with the concepts of the other sub-contexts, whose extents form a cover of the corresponding sub-universe. Then decision rules are derived from the maximal "consistent" concepts.

The paper is organized as follows. Section 2 recalls some basic notions in RST and FCA. From the point of value reduct, the relationship between the basic concepts in RST and the concepts in FCA is surveyed in section 3, and then a heuristic algorithm is proposed. Experiment results are shown in section 4 to illustrate the validity of our algorithm. Conclusions are drawn in section 5.

2 Preliminaries

In this section, some basic notions and theorems in RST and FCA related to this paper are recalled.

2.1 Rough Set Theory [14, 16]

In RST, a decision table is a 4-tuple as follows $(U, A = C \cup D, V, f)$, where U is the universe, A is a set of attributes, with C being the conditional attribute set and D, the decision attribute set, $V = \bigcup_{a \in C \cup D} V_a$ is the set of all attribute values, and $f : U \times A \mapsto V$ assigns every object x in U with an attribute value for each attribute, in some literature, $a(x)$ is used instead of $f(x, a)$.

Let $S = (U, A = C \cup D, V, f)$ be a decision table, and every $P \subseteq A$ generates an indiscernibility relation $Ind(P)$ on U, with

$$Ind(P) = \{(x, y) \in U \times U | f(x, a) = f(y, a), \forall a \in P\}. \tag{1}$$

$U/Ind(P) = \{[x]_P | x \in U\}$ is a partition of U by P, where $[x]_P = \{y | (x, y) \in Ind(P)\}$. For simplicity, we use $[x]_a$ instead of $[x]_{\{a\}}$.

Upper and lower approximations are use to depict "concepts" in RST.

Definition 1. *Suppose $P \subseteq A$ and $X \subseteq U$ (a concept X), the P upper and lower approximations of set X are defined as*

$$\overline{P}X = \cup\{[x]_P | [x]_P \cap X \neq \emptyset\}, \quad \underline{P}(X) = \cup\{[x]_P | [x]_P \subseteq X\}. \tag{2}$$

Definition 2. *Let $P, Q \subseteq A$. The P-positive region of Q is defined as*

$$Pos_P(Q) = \bigcup_{X \in U/Ind(Q)} \underline{P}X \tag{3}$$

S is called a consistent decision table if $Pos_P(Q) = U$.

Definition 3. *Let S be a decision table, and $B \subseteq C$. If (1) $Pos_B(D) = Pos_C(D)$ holds, (2) $Pos_{B_1}(D) = Pos_C(D)$ does not hold for any $B_1 \subset B$, then B is called a reduct of S.*

Definition 4. *In a decision table S, a decision rule $\mathbb{R}$ is depicted as*

$$\mathbb{R} : des([x]_C) \longmapsto des([x]_D), \tag{4}$$

where $des([x]_C)$ ($des([x]_D)$) is the description of the equivalence class $[x]_C$ ($[x]_D$). In this case, object x satisfies rule $\mathbb{R}$. If $\frac{|[x]_C \cap [x]_D|}{|[x]_C|} = 1$, then rule $\mathbb{R}$ is definite.

Definition 5. *The support set of rule $\mathbb{R}$ is defined as*

$$Supp(\mathbb{R}) = \{x \in U | x \text{ satisfies } \mathbb{R}\}. \tag{5}$$

Definition 6. *Let $S = (U, A, V, f)$ be a decision table, $U/Ind(D) = \{D_1, D_2, \cdots, D_k\}$, $[x]_C \subseteq D_i$ for some $i, 1 \leq i \leq k$, $B \subseteq C$. If (1) $[x]_B \subseteq D_i$ holds, and (2) for all $B_1 \subset B$, $[x]_{B_1} \not\subseteq D_i$, then B is called an attribute value reduct of x.*

Definition 6 is equivalent to the definition of value reduct in [16].

Every object may have more than one value reducts. Denote $VR(x) = \{vr|$ is a value reduct of $x\}$. For all $x \in U$, $vr_x \in VR(x)$, then $\{vr_x | x \in U\}$ is called a value reduct set.

Example 1 is used to illustrate attribute reduction and attribute value reduction.

Example 1. [16] Suppose we are given the following decision table S (Table 1), where $C = \{a, b, c, d\}$ and $D = \{e\}$. $\{a, b, d\}$ is an attribute reduct. After attribute reduction, Table 2 is obtained.

Table 1. A decision table S

	a	b	c	d	e
1	1	0	0	1	1
2	1	0	0	0	1
3	0	0	0	0	0
4	1	1	0	1	0
5	1	1	0	2	2
6	2	1	0	2	2
7	2	1	0	2	2
8	2	2	2	2	2

Table 2. S after attribute reduction

	a	b	d	e
1	1	0	1	1
2	1	0	0	1
3	0	0	0	0
4	1	1	1	0
5	1	1	2	2
6	2	1	2	2
7	2	1	2	2
8	2	2	2	2

Table 3. A decision table S

	a	b	d	e
1	1	0	1	1
2	1	0	0	1
3	0	0	0	0
4	1	1	1	0
5	1	1	2	2
6	2	1	2	2
7	2	2	2	2

Table 4. S after value reduction

	a	b	d	e
1	1	0	-	1
$1'$	-	0	1	1
2	1	0	-	1
$2'$	1	-	0	1
3	0	-	-	0
4	-	1	1	0
5	-	-	2	2
6	2	-	-	2
$6'$	-	-	2	2
7	2	-	-	2
$7'$	-	2	-	2
$7''$	-	-	2	2

- means 'don't care'

Then, reductant rows are deleted (Table 3). Besides, every rule (every row) in Table 3 can be further simplified without conflict (Table 4). For example, rule 1 can be simplified as $a = 1 \wedge b = 1 \Longrightarrow e = 1$.

From Table 4, we can see that, for objects 1 and 2 we have two value reducts respectively. Decision objects 3, 4 and 5 have only one value reduct respectively. The remaining objects 6 and 7 contain two and three value reducts respectively. Thus there are $4 \times 2 \times 3$ (not necessarily different) solutions.

2.2 Formal Concept Analysis [4, 20]

In FCA, a triplet (G, M, I) is called a (formal) context, if G is a non-empty set of objects, M is a non-empty set of attributes, and $I \subseteq G \times M$ is a binary relation from G to M, $(g, m) \in I$ if object x has attribute a.

A Galois connection between $(G, \subseteq)$ and $(M, \subseteq)$ is defined as follows:

$$X' = \{m \in M | \ \forall\ g \in X,\ (g, m) \in I\},\ \ B' = \{g \in G | \ \forall\ m \in M,\ (g, m) \in I\}.$$

where $X \subseteq G$, $B \subseteq M$. We write X'' for $(X')'$ etc., and similarly, B'' for $(B')'$ etc., what's more, $\{g\}'$ is denoted by g' etc., and $\{m\}'$, by m' for convenience.

A concept of (G, M, I) is a pair (X, B) with $X' = B$, and $B' = X$. X is called the extent and B, the intent of the concept (X, B).

Proposition 1. *For any $X, X_1, X_2 \subseteq G$, $B, B_1, B_2 \subseteq M$,*
(1) *$X_2' \supseteq X_1'$ if $X_1 \subseteq X_2$, $B_2' \supseteq B_1'$ if $B_1 \subseteq B_2$;*
(2) *$X \subseteq X''$, $B \subseteq B''$;*
(3) *$X' = X'''$, $B' = B'''$;*
(4) *$X \subseteq B' \Longleftrightarrow B \subseteq B' \Longleftrightarrow X \times B \subseteq I$;*
(5) *$(\bigcup_{t \in T} X_t)' = \bigcap_{t \in T} X_t'$, $(\bigcup_{t \in T} B_t)' = \bigcap_{t \in T} B_t'$, where T is an index set.*

From Proposition 1 , for any $X \subseteq G$, $B \subseteq M$, (X'', X') and (B', B'') are both concepts.

The set of all concepts of (G, M, I) is denoted by $\mathfrak{B}(G, M, I)$, and a partial order $\leq$ on it is given by

$$(A_1, B_1) \leq (A_2, B_2) \Longleftrightarrow A_1 \subseteq A_2 \text{ (equivalently } B_2 \subseteq B_1). \tag{6}$$

Theorem 1. *Let (G, M, I) be a context, and $\mathfrak{B}(G, M, I)$ be the set of all concepts of (G, M, I). $(X_t, B_t) \in \mathfrak{B}(G, M, I)$, $t \in T$, then*

$$\bigwedge_{t\in T}(X_t, B_t) = (\bigcap_{t\in T} X_t, (\bigcup_{t\in T} B_t)''), \ \bigvee_{t\in T}(X_t, B_t) = ((\bigcup_{t\in T} X_t)'', \bigcap_{t\in T} B_t), \tag{7}$$

are concepts. Thus, $\underline{\mathfrak{B}}(G, M, I) = (\mathfrak{B}(G, M, I), \leq)$ is a complete lattice, which is called a concept lattice.

In many cases, attributes may be many-valued (for example, "color", "shape", etc.) in contrast to the one-valued attributes considered above. Correspondingly, there are many-valued contexts.

Definition 7. *A many-valued context (G, M, W, I) consists of sets G, M, W, and a ternary relation I between G, M and W (i.e., $I \subseteq G \times M \times W$) from which it holds that $(g, m, w) \in I$ and $(g, m, v) \in I$ always imply $w = v$.*

Sometimes, people write $m(g) = w$ instead of $(g, m, w) \in I$ [4]. In this way, a decision table is actually a many-valued context with decisions.

To assign concepts to a many-valued context, Ganter and Wille [4] use scales to transformed the many-valued contexts into one-valued contexts, and the concepts of the derived contexts are interpreted as those of the many-valued context.

Definition 8. *A scale for the attribute m of a many valued-context is a (one-valued) context $\mathbb{S}_m = \{G_m, M_m, I_m\}$ with $m(G) \subseteq G_m$.*

Definition 9. *If (G, M, W, I) is a many-valued context, and $\mathbb{S}_m$, $m \in M$ are scale contexts, then the context with respect to plain scaling is the context (G, N, J) with $N = \bigcup_{m\in M} \dot{M}_m$, and $gJ(m, n) \Longleftrightarrow m(g) = w$ and $wI_m n \Longleftrightarrow m(g)I_m n$, where $\dot{M}_m = \{m\} \times M$.*

Nominal scale $\mathbb{I}_n = (\{1, 2, \cdots, n\}, \{1, 2, \cdots, n\}, =)$ is the most common scale.

Example 2. The reduced decision table S in Table 3 can be converted into a formal context (shown in Table 5) by using nominal scale.

3 FCA Based Approach to Minimal Value Reduct Set

In this paper, only definite rules are concerned.

A set of decision rules can be obtained from the decision table. They have advantages in the situation that someone reads them and understands the meanings. When considering a situation in which one reads rules, a rule set is desired

Table 5. The derived formal context $\mathbb{K}$ of S

	$(a,0)$	$(a,1)$	$(a,2)$	$(b,0)$	$(b,1)$	$(b,2)$	$(d,0)$	$(d,1)$	$(d,2)$
1	0	1	0	1	0	0	0	1	0
2	0	1	0	1	0	0	1	0	0
3	1	0	0	1	0	0	1	0	0
4	0	1	0	0	1	0	0	1	0
5	0	1	0	0	1	0	0	0	1
6	0	0	1	0	1	0	0	0	1
7	0	0	1	0	0	1	0	0	1

to satisfy the following three conditions, possibly at the same time. 1) They can explain most of possible situations as a rule set, 2) The size of a rule set is small and thus memorable and manageable, 3) Description of each rule is simple enough for understanding the meaning [13]. Both 2) and 3) can be tackled by finding the minimal value reduct set.

In the sequel, a heuristic method based on FCA is proposed to find the approximate minimal value reduct set. First, some results with respect to the basic concepts in RST and formal concepts in FCA are introduced.

3.1 Basic Concepts in RST and Concepts Extents in FCA

In RST, the antecedent of every definite rule in a consistent decision table is obtained from a basic concept. Research have shown that there is a close relation between RST and FCA [5, 21, 24] from the view of knowledge representation.

For a decision table (also called a many-valued context in FCA) $S = (U, A = C \cup D, V, F)$, its derived one-value context by means of nominal scale $\mathbb{I}_n$ (decision attributes omitted) is (U, N, J), where $N = \bigcup_{a \in C}\{a\} \times V_a$ and $(x, (a, v_a)) \in J \Longleftrightarrow f(x, a) = a(x) = v_a$. Then the basic concepts of the former and the formal concepts of the latter are essentially the same, which is formally described as follows.

Theorem 2. *$S = (U, A, V, F)$ is a decision table, then $\sigma(U/Ind(C)) = \mathfrak{B}_G(U, N, J)$, where $\sigma(U/Ind(C))$ is the σ-algebra generated by $U/Ind(C)$.*

Proof. (1) $\forall \mathcal{C} \in \sigma(U/Ind(C))$, there must be an object $x \in U$, and an attribute subset $B \subseteq C$ such that $\mathcal{C} = [x]_B = \bigcap_{a \in B}[x]_a$.

On the one hand, $\forall y \in [x]_B$, $\forall a \in B$, we have $f(x, a) = f(y, a)$. Let $f(x, a) = v_a$, then, in the derived one-value context (U, N, J), $yJ(a, v_a)$, so $y \in (a, v_a)'$, which yields $[x]_a \subseteq (a, v_a)'$.

On the other hand, for all $y \in (a, v_a)'$, $yJ(a, v_a) \Longleftrightarrow a(y) = v_a \Longleftrightarrow f(y, a) = v_a = f(x, a) \Longrightarrow y \in [x]_a$, which yields $[x]_a \supseteq (a, v_a)'$. So $[x]_a = (a, v_a)'$ follows.

Now, we have, $\mathcal{C} = [x]_B = \bigcap_{a \in B}[x]_a = \bigcap_{a \in B}(a, f(x, a))' = N_1'$ with $N_1 = \{(a, f(x, a) | a \in B)\}$. So, $\mathcal{C} \in \mathfrak{B}_G(U, N, J)$. $\sigma(U/Ind(C)) \subseteq \mathfrak{B}_G(U, N, J)$ follows.

(2) For all $X \in \mathfrak{B}_G(U, N, J)$, $\exists N_1 \subseteq N$, such that $X = N_1' = \bigcap_{(a,v) \in N_1}(a, v)'$. $v = f(x, a)$, $\exists x \in X$, from (1), we have $(a, v)' = [x]_a$, so $X = \bigcap_{(a,v) \in N_1}(a, v)\ ' = \bigcap_{(a,v) \in N_1}[x]_a = \bigcap_{a \in B}[x]_a = [x]_B$, with $B = \{a | (a, v) \in N_1\}$, from which we can conclude $X \in \sigma(U/Ind(C))$. Together with (1), the conclusion is reached.

From Theorem 2, if a concept is "consistent" w.r.t. some decision class, then a definite rule can be derived. However, not all concepts can derive definite rules.

Lemma 1. *Let* $\mathbb{K}_i = (U_i, N, J_i)$, $i = 1, 2, \cdots, k$ *be a family of one-valued contexts, with* $U_i \cap U_j = \emptyset$, *for all* $i \neq j$, *and* $\mathbb{K} = (\bigcup_{i=1}^{k} U_i, N, \bigcup_{i=1}^{k} I_i)$. $\forall\ (X, B) \in \underline{\mathfrak{B}}(\mathbb{K})$, $\exists B_i \subseteq A$, *such that* $(X \cap U_i, B_i) \in \underline{\mathfrak{B}}(\mathbb{K}_i)$.

Proof. Since $\forall\ (X, B) \in \underline{\mathfrak{B}}(\mathbb{K})$, $X = B' = \bigcup_{i=1}^{k} B'^i$, where B'^i denotes the objects who shares all attributes from B in $\mathbb{K}_i$, $X \cap U_i = B'^i \in \mathfrak{B}_G(\mathbb{K}_i)$. Denote $B_i = (B'^i)'$, $(X \cap U_i, B_i) \in \underline{\mathfrak{B}}(\mathbb{K}_i)$.

To get definite rules, we only need take the hypotheses into consideration.

Definition 10. [6, 7] *Let* $U = \bigcup_{i=1}^{k} U_i$ *with* $U_i \cap U_j = \emptyset$, $i \neq j$. *If* $(X, B) \in \underline{\mathfrak{B}}(\mathbb{K}_i)$ *for some* i, *and* B *does not conflict with other intents of concepts in* $\underline{\mathfrak{B}}(\bigcup_{j \neq i} \mathbb{K}_j)$, *then* (X, B) *is referred to a positive hypothesis with respect to* U_i.

For the positive hypothesis (X, B) w.r.t. D_i, rules in the following form can be obtained: $B \Longrightarrow des([x]_D)$ where $x \in X$. To make sure the rules we get are those derived from the value reducts, we need to get the shrunk intents.

Definition 11. *Let* $\mathbb{K}$ *be a formal context, and* $(X, B) \in \underline{\mathfrak{B}}(\mathbb{K})$, $N_1 \subseteq B$. N_1 *is called a shrunk intent of* (X, B) *iff* $N_1' = X$, *and* $\forall\ a \in T$, $(N_1 - \{a\})' \neq X$.

From Definition 11, we can see the shrunk intents coincident with the "proper predictors" in [2].

Theorem 3. *Let* $S = (U, A, V, F)$ *be a decision table, and* $\mathbb{K} = (U, N, J)$ *is its derived one-valued context.* $T_x : S \mapsto \mathbb{K}, T_x(B) = \{(a, f(x, a)) \in N | a \in B\}$. *For* $X \in \sigma(U/Ind(C))$, $B \subseteq C$, *if* B *is an attribute value reduct for all* $\mathbb{R}_x$, $x \in X \in \sigma(U/Ind(C))$, *and* $X = [x]_B$, *then* (X, X') *is a positive hypothesis with respect to* $D_i \in U/Ind(D)$ *for some* i, *and* $T_x(B)$ $(\forall x \in X)$ *is the shrunk intent of* (X, X') *in* (U, N, J).

Proof. Since B is an attribute value reduct of $x \in X$, then $X = [x]_B \subseteq D_i$ for some i. It follows that (X, X') is a positive hypothesis with respect to D_i.

From Definition 6, for all $x \in X$, B is an attribute value reduct, $(X, X') = ([x]_B, [x]'_B) = (\bigcap_{a \in B}[x]_a, [x]'_B) = (\bigcap_{a \in B}(a, f(x, a))', [x]'_B) = ((T_x(B))', [x]'_B)$, $(T_x(B))' = X$. If $T_x(B)$ is not a shrunk intent of (X, X'), there must be an $(a_0, v_0) \in T_x(B)$, with $v_0 = f(x, a_0)$ (denote $N_1 = (T_x(B)) - \{(a_0, v_0)\}$) such that $N_1' = X = [x]_B$, then $N_1' = \bigcap_{a \in B - \{a_0\}}(a, f(x, a))' = [x]_{B - \{a_0\}} = X \subseteq D_i$, so we deduce that $B - \{a_0\}$ contains a value reduct of $\mathbb{R}_x$, which contradicts to the premise that B is a value reduct of $\mathbb{R}_x$. So $T_x(B)$ is a shrunk intent of (X, X').

Theorem 4. *Let* $S = (U, A, V, F)$ *be a decision table, and* $\mathbb{K} = (U, N, J)$ *be its derived one-valued context.* $T_x : S \mapsto \mathbb{K}$, $T_x(B) = \{(a, f(x, a)) \in N | a \in B\}$. *If* (X, X') *is a maximal element of all positive hypotheses with respect to* D_i *for some* i, *and* $N_1 \subseteq N$ *is the shrunk intent of* (X, X') *in* (U, N, J), *then* $T_x^{-1}(N_1)$ *is an attribute value reduct for all* $\mathbb{R}_x$, $x \in X$ *with* $X = [x]_{T_x^{-1}(N_1)}$.

Proof. (X, X') is a concept, from Theorem 2, then there must be an attribute subset $B \subseteq C$ such that $X = [x]_B$, $\forall x \in X$, and for all $B_1 \supsetneqq B$, $X \neq [x]_{B_1}$. We have $X' = T_x(B)$. This is because $X = [x]_B = \bigcap_{a\in B}[x]_a = \bigcap_{a\in B}(a, f(x, a))' = (\bigcup_{a\in B}(a, f(x, a)))' = (T_x(B))'$, which implies $T_x(B) \subseteq X'$. If $\exists(a_0, v_0) \in X' - T_x(B)$, then $xJ(a_0, v_0)$ $\forall x \in X$ with $a_0 \notin B$, so $f(x, a_0) = v_0$ $\forall x \in X$, from which we can deduce that $[x]_B = X \subseteq [x]_{a_0}$, $X = [x]_B \cap [x]_{a_0} = [x]_{B\cup\{a_0\}}$, which contradicts to the fact that for all $B_1 \supsetneqq B$, $X \neq [x]_{B_1}$.

Since N_1 is a shrunk intent of (X, X'), $N_1 \subseteq T_x(B)$, $T_x^{-1}(N_1) \subseteq B$ then for all $a \in T_x^{-1}(N_1)$, and $\forall x, y \in X$, $f(x, a) = f(y, a)$. So $X = N_1' = \bigcap_{(a,v)\in N_1}(a, v)' = \bigcap_{(a,v)\in N_1}(a, f(x, a))' = \bigcap_{(a,v)\in N_1}[x]_a = \bigcap_{a\in T_x^{-1}(N_1)}[x]_a = [x]_{T_x^{-1}(N_1)} \subseteq D_i$, $\forall x \in X$. For all $a \in T_x^{-1}(N_1)$, denote $B_2 = T_x^{-1}(N_1) - \{a\}$. Then we have $X = [x]_{T_x^{-1}(N_1)} \subseteq [x]_{B_2}$. This is because if $X = [x]_{B_2}$, we can deduce that $X = (T_x(B_2))' = (T_x^{-1}(N_1) - \{a\})'$, which is contradict to the premise that N_1 is a shrunk intent of (X, X') . If $X \subsetneqq [x]_{B_2}$, from the premise that (X, X') is a maximal element of all positive hypotheses w.r.t. D_i for some i, we can conclude that $[x]_B$ is not an extent of a positive hypotheses w.r.t. D_i, i.e., $[x]_{B_2} \not\subseteq D_i$, which yields $T_x^{-1}(N_1)$ is a value reduct for all $\mathbb{R}_x$, $x \in X$.

Theorem 3 and Theorem 4 tell us that the shrunk intents of the maximal positive hypotheses are value redcuts. On the other hand, the maximal positive hypotheses are the ones whose intents are shared by as many objects as possible. So we design a heuristic approach to minimal value reduct set. A most relative topic has been discussed in [3] from the aspect of the overall feature set of a decision class.

3.2 Heuristic Approach to Approximate Minimal Value Reduct Set Based on FCA

The main steps of our approach is stated as follows. First, a decision table S is transformed to its derived one-value context $\mathbb{K}$ by means of nominal scale $\mathbb{I}_n$ (decision attributes are deleted), then $\mathbb{K}$ is divided into some sub-contexts according to the decisions. For every sub-context, the extents of positive hypotheses are found, then they are sorted according to Principles 1 and 2, and finally choose the extents top ranked in the candidates to generate a rule.

Principle 1: For two concepts (X_i, B_i) , (X_j, B_j), if $i > j$ and $|X_i| < |X_j|$, swap (X_i, B_i) and (X_j, B_j);

Principle 2: For two concepts (X_i, B_i) , (X_j, B_j), with $i > j$ and $|X_i| = |X_j|$, and current object subset which is not covered by the extents of positive hypotheses being U', if $|U' \cap X_i| > |U' \cap X_j|$, then swap (X_i, B_i) and (X_j, B_j).

Example 3 illustrates how our method works.

Example 3. The decision table is shown in (Table 1), its reduct is a, b, d. it is transformed into a formal context $\mathbb{K}$ (shown in Table 5).

Then $\mathbb{K}$ is divided into 3 sub-contexts according to the decisions: $\mathbb{K}_0$, $\mathbb{K}_1$ and $\mathbb{K}_2$ (shown in Table 6, Table 7 and Table 8).

The concepts of $\mathbb{K}_0$ are $CPT_1 = (\{3, 4\}, \emptyset)$, $CPT_2 = (\{3\}, \{(a, 0), (b, 0), (d, 0)\})$, $CPT_3 = (\{4\}, \{(a, 1), (b, 1), (d, 1)\})$ and $CPT_4 = (\emptyset, A')$. The extents of positive

Algorithm 1. FCA Based Algorithm for Approximate Minimal Value Reduct Rule Set

Require:
A decision table, $S = (U, C \cup D, V, F)$;
An attribute reduct of S, red;
Ensure:
An approximate optimal minimal value reduct set, $RULESET$.
1: $RULESET = \emptyset$;
2: Transform the reduced decision table $S_1 = (U, red \cup D, V_1, F_1)$ into its derived context $\mathbb{K} = (U, N, J)$;
3: Divide $\mathbb{K}$ into sub-contexts $\mathbb{K}_i = (U_i, N, J_i)$ according to the decisions,$RULE_i = \emptyset$, $cov = \emptyset$; $//RULE_i$ stores the rules derived from $\underline{\mathfrak{B}}(\mathbb{K}_i)$, cov stores the objects covered by the extents of the positive concepts w.r.t. D_i.
4: **for** each sub-context $\mathbb{K}_i$ **do**
5: $uncov = U_i - cov$; $//uncov$ stores the objects that are not covered by the extents of the positive concepts currently.
6: Generate all concepts of $\mathbb{K}_i$, choose the positive hypotheses w.r.t. D_i as candidates;
7: **if** $uncov \neq \emptyset$ **then**
8: Sort the concepts descending according to Principles 1 and 2. The sorted positive hypotheses are stored in SC;
9: $RULE_i = RULE_i \bigcup \{shrunk(X, B) \rightarrow des([X]_D)\}$, where (X, B) is the first in SC; shrunk(X,B) is the shrunt intent of (X,B) in $\mathbb{K}$.
10: cov=cov$\cup\{X\}$;
11: SC=SC-$\{X_1 | X_1 \in$ SC, $X_1 \subseteq cov\}$;
12: **end if**
13: Go to 5
14: **end for**
15: $RULESET = \bigcup RULE_i$;
16: **return** $RULESET$;

hypotheses w.r.t. D_0 $(e = 0)$ are CPT_2 and CPT_3, and $\{\{3\}, \{4\}\}$ is a cover of {3,4}. So we get two rules $shrunk(CPT_2) \rightarrow e = 0$, that is $(a, 0) \rightarrow e = 0$ (or equivalently $a = 0 \rightarrow e = 0$) and $shrunk(CPT_3) \rightarrow e = 0$ $(b, 1) \wedge (d, 1) \rightarrow d = 0$ (or equivalently $b = 1 \wedge d = 1 \rightarrow d = 0$).

Similarly , we obtain rule: $(a, 1) \wedge (b, 0) \rightarrow e = 1$ from $\mathbb{K}_1$, and rule $(d, 2) \rightarrow e = 2$ from $\mathbb{K}_2$. The four rules we get corresponds to the minimal value reduct set of S.

4 Experiments

The effectiveness of the proposed algorithm is tested with five-fold cross validation method on a collection of nine benchmark data sets from UCI machine learning repository [1]. Before employing RST and FCA, the real value data must be discretized.

The detailed information of the benchmark data sets are summarized in Table 9, where "Data Set" denotes data set name, $|U|$ stands for the the number

Table 6. $\mathbb{K}_0$

$U_0 \backslash A'$	$(a,0)$	$(a,1)$	$(a,2)$	$(b,0)$	$(b,1)$	$(b,2)$	$(d,0)$	$(d,1)$	$(d,2)$
3	1	0	0	1	0	0	1	0	0
4	0	1	0	0	1	0	0	1	0

Table 7. $\mathbb{K}_1$

$U_1 \backslash A'$	$(a,0)$	$(a,1)$	$(a,2)$	$(b,0)$	$(b,1)$	$(b,2)$	$(d,0)$	$(d,1)$	$(d,2)$
1	0	1	0	1	0	0	0	1	0
2	0	1	0	1	0	0	1	0	0

Table 8. $\mathbb{K}_2$

$U_2 \backslash A'$	$(a,0)$	$(a,1)$	$(a,2)$	$(b,0)$	$(b,1)$	$(b,2)$	$(d,0)$	$(d,1)$	$(d,2)$
5	0	1	0	0	1	0	0	0	1
6	0	0	1	0	1	0	0	0	1
7	0	0	1	0	0	1	0	0	1

Table 9. Benchmark data sets information

NO.	Data Set	$\|U\|$	$\|C\|$	$\|U/Ind(D)\|$	$\|U/Ind(C)\|$	MinD	MaxD
1	Soybean	47	34	4	7	1	2
2	Zoo	101	16	7	22	1	7
3	Iris Data	150	4	3	19	2	9
4	Glass	214	9	6	145	5	50
5	Liver Disorder	345	6	7	315	129	186
6	Monks' Problem	432	6	2	36	18	18
7	WDBC	569	30	2	401	113	268
8	Tic-Tac-Toe	958	9	2	958	332	636
9	Car Evaluation	1728	6	4	1728	65	1210

Table 10. Results of five alogrithms

	Alg 1		Alg2		Alg3		Alg4		Alg5	
NO.	AR(%)	RSS	AR(%)	RSS	AR(%)	RSS	AR(%)	RSS	AR(%)	RSS
1	100.00	5	100.00	5	100.00	5	100.00	5	100	5
2	92.00	**12**	**94.00**	13	92.00	14	94.00	12.2	94.00	12.4
3	96.67	**8.6**	97.33	9	97.33	9	96.67	8.8	**98**	9.2
4	65.24	**47.8**	67.44	92.6	62.79	85.8	65.12	71	**70.23**	69.4
5	**66.86**	**117.4**	52.75	173.6	43.19	175.2	57.97	137.4	59.71	131.6
6	100.00	22	100.00	22	100.00	22	100.00	22	100	22
7	**93.51**	62	89.30	92.6	84.21	219.6	91.75	68.8	91.23	**56.4**
8	**83.35**	**163.6**	69.74	352.6	81.36	473.8	75.18	196.4	74.45	171.4
9	**89.86**	**189.8**	83.75	247.4	68.88	207.6	78.49	194.6	71.69	213

of objects, $|C|$ ($|D|$)denotes the number of conditional (decision) attributes, and $|U/Ind(C)|$ ($|U/Ind(D)|$) represents the number of conditional (decision) equivalence classes, MinD (MaxD) stands for the minimum (maximum) number of condition equivalent classes in some decision classes.

The average performances of five-fold cross validation are shown in Table 10, where Alg1 stands for the approach proposed in this paper, and Alg2-Alg5 denote naive approach, heuristic approach, matrix approach, and inductive approach respectively. "AR" means Accuracy Rate, and "RSS" is the abbreviation for "Rule Set Size". The last four approaches are implemented by RIDAS [19].

Form Table 9 and Table 10, we can conclude that, when the number of the conditional equivalence classes is small ($\leq$ 145 in our experiments), the maximum number and minimum number of condition classes in decision classes are small too, Alg1 performs as good as the other four, though it can obtain a rule set with a smaller size. When the number of the condition equivalent classes is large ($\geq$ 315 in our experiments), the maximum number and minimum number of condition classes in decision classes are relatively large, Alg1 has a much better performance in both size of rule set and accuracy in most cases.

5 Conclusions

Reduction is a core issue in Rough Set Theory. Current reductions falls into 3 categories: attribute reduction, tuple reduction and value reduction. From the reduced tables, decision rules can be derived. For the purpose of storage and better understanding, minimization of the rule set is desired, and it is NP-hard. To tackle this problem, a heuristic approach to minimal value reduct set based on Formal Concept Analysis is proposed in this paper. Experiments show that our approach is valid with a higher accuracy, especially when the number of the condition equivalent classes are relatively large.

Acknowledgement. This research has been supported by the National Natural Science Foundation of P. R. China (NSFC) under grant No. 61073146 and No.61272060, Natural Science Foundation of Chongqing under grant No. cstc2012jjA40047, cstc2012jjA1649 and NO.cstc2012jjA40032 and research program of the Municipal Education Committee of Chongqing under grant No.KJ110512.

References

[1] `http://archive.ics.uci.edu/ml/datasets.html`
[2] Ganter, B., Kuznetsov, S.O.: Hypotheses and version spaces. In: Ganter, B., de Moor, A., Lex, W. (eds.) ICCS 2003. LNCS (LNAI), vol. 2746, pp. 83–95. Springer, Heidelberg (2003)
[3] Ganter, B., Kuznetsov, S.O.: Scale coarsening as feature selection. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 217–228. Springer, Heidelberg (2008)
[4] Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer-Verlag New York, Inc. (1997)

[5] Kent, R.E.: Rough concept analysis: A synthesis of rough sets and formal concept analysis. Fundamenta Informaticae 27(2), 169–181 (1996)
[6] Kuznetsov, S.O.: Complexity of learning in concept lattices from positive and negative examples. Discrete Applied Mathematics 142(1), 111–125 (2004)
[7] Kuznetsov, S.: Mathematical aspects of concept analysis. Journal of Mathematical Sciences 80(2), 1654–1698 (1996)
[8] Li, J., Mei, C., Lv, Y.: A heuristic knowledge-reduction method for decision formal contexts. Computers & Mathematics with Applications 61(4), 1096–1106 (2011)
[9] Li, T.-J.: Knowledge reduction in formal contexts based on covering rough sets. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 128–135. Springer, Heidelberg (2009)
[10] Liu, M., Shao, M., Zhang, W., Wu, C.: Reduction method for concept lattices based on rough set theory and its application. Computers & Mathematics with Applications 53(9), 1390–1410 (2007)
[11] Medina, J., Ojeda-Aciego, M.: Towards attribute reduction in multi-adjoint concept lattices. In: Proceedings of the 7th International Conference on Concept Lattices and Their Applications, pp. 92–103 (2010)
[12] Nian, F., Li, M.: Attribute value reduction in variable precision rough set. In: Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 904–906. IEEE (2005)
[13] Omura, K., Aoki, K.A., Kudo, M.: Attribute value reduction for gaining simpler rules. In: Proceedings of 2011 IEEE International Conference on Granular Computing (GrC), pp. 527–532. IEEE (2011)
[14] Pawlak, Z.: Rough sets. International Journal of Computer & Information Sciences 11(5), 341–356 (1982)
[15] Pawlak, Z.: On superfluous attributes in knowledge representation system. Technical Sciences 32(34) (1984)
[16] Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Norwell (1991)
[17] Wang, G.: Rough set theory and knowledge acquisition. Xian Jiaotong University Press, Xi'an (2001)
[18] Wang, G., Yao, Y., Yu, H.: A survey on rough set theory and applications. Chinese Journal of Computers 32(7), 1229–1246 (2009)
[19] Wang, G., Zheng, Z., Zhang, Y.: Ridas-a rough set based intelligent data analysis system. In: Proceedings of 2002 International Conference on Machine Learning and Cybernetics, vol. 2, pp. 646–649. IEEE (2002)
[20] Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets. NATO Advanced Study Institutes Series, vol. 83, pp. 445–470 (1982)
[21] Wolff, K.E.: A conceptual view of knowledge bases in rough set theory. In: Ziarko, W., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, p. 220. Springer, Heidelberg (2001)
[22] Yao, M., Yang, J., Zhang, H., Wu, W.: An attribute value reduction algorithm based on set operations. In: Proceedings of 2009 First International Workshop on Database Technology and Applications, pp. 181–183 (2009)
[23] Zhang, B., Wang, N.: Research of discernible matrix-based algorithm for attribute value reduction. In: Proceedings of 2010 International Conference on Intelligent Computing and Cognitive Informatics (ICICCI), pp. 349–352. IEEE (2010)
[24] Zhao, J., Liu, L.: Construction of concept granule based on rough set and representation of knowledge-based complex system. Knowledge-Based Systems 24(6), 809–815 (2011)

Comparison of Two Models of Probabilistic Rough Sets

Bing Zhou[1] and Yiyu Yao[2]

[1] Department of Computer Science, Sam Houston State University
Huntsville, Texas, USA 77340
zhou@shsu.edu
[2] Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada, S4S 0A2
yyao@cs.uregina.ca

Abstract. To generalize the classical rough set model, several proposals have been made by considering probabilistic information. Each of the proposed probabilistic models uses three regions for approximating a concept. Although the three regions are similar in form, they have different semantics and therefore are appropriate for different applications. In this paper, we present a comparative study of a decision-theoretic rough set model and a confirmation-theoretic rough set model. We argue that the former deals with drawing conclusions based on available evidence and the latter concerns evaluating difference pieces of evidence. By considering both models, we can obtain a more comprehensive understanding of probabilistic rough sets.

Keywords: rough sets, probabilistic approximations, Bayesian inference, decision-theoretic rough sets, confirmation-theoretic rough sets.

1 Introduction

Rough set theory was introduced by Pawlak [17] as a tool for analyzing data represented in a tabular form. Two central notions of the theory are the indiscernibility of objects and the induced approximation of a set due to indiscernibility. The approximation can be represented either as a pair of lower and upper approximations or as three pair-wise disjoint positive, negative and boundary regions. Approximations in the classical model are defined by using qualitative relationships between two sets, namely, set inclusion and non-empty set intersection. To overcome limitations of such a qualitative model, probabilistic rough set models have been proposed [7, 10, 11, 19, 20, 22–25, 31], in which probabilistic relationships between the two sets are considered.

Yao *et al.* [30, 31] proposed a probabilistic model, called a decision-theoretic rough set (DTRS) model, by introducing a pair of thresholds on the conditional probabilities $Pr(C|[x])$ for defining probabilistic approximations, where C is the set to be approximated and $[x]$ is the equivalence class containing x. Based on the well established Bayesian decision theory, the pair of thresholds can be

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 121–132, 2013.

systematically calculated and interpreted in terms of more practically operable notions such as cost, risk, benefit, and so on. Yao and Zhou [32] introduced a naive Bayesian rough set (NBRS) model to give a practical method for estimating the required conditional probability of a probabilistic rough set model. The variable precision rough set (VPRS) model [33] uses $1 - Pr(C|[x])$ to quantify classification error induced by $[x]$, or inclusion degree of $[x]$ in C, and can be viewed as a special case of the decision-theoretic rough set model.

Greco *et al.* [7–9] proposed a parameterized rough set model by using a pair of thresholds on a Bayesian confirmation measure, in addition to a pair of thresholds on the conditional probabilities. The mixture of a Bayesian confirmation measure and the conditional probabilities may deserve further investigations. Ślęzak and Ziarko [21, 22, 20] introducing a Bayesian rough set model by drawing correspondence between the fundamental notions of rough sets and statistics. One the one hand, their model is related to decision-theoretic rough set models in the sense that the *a priori* probability $Pr(C)$ is used as a threshold on the conditional probabilities $Pr(C|[x])$. On the other hand, their model is also related to parameterized model of Greco *et al.* In the sense that $Pr(C|[x]) - Pr(C)$ is a Bayesian confirmation measure, in this case, threshold 0 used on the measure $Pr(C|[x]) - Pr(C)$.

All these probabilistic rough set models share the same form of approximations. That is, they all use probabilistic lower and upper approximations or three probabilistic regions. Different types of probabilistic three regions are obtained from the different ways in which the conditional probabilities are used. Several questions arrive naturally: what are the main differences between different models? do we really need different models? when it is appropriate to apply a particular model? We search for answers to these questions by examining of semantics of various probabilistic rough set models [28].

Although three regions are similar in form, they have different semantics interpretations and are therefore appropriate for different applications. From this point of view, the main objective of the paper is to compare two particular models, namely, decision-theoretic rough set models and confirmation-theoretic rough set models. It should be noted that our formulation of confirmation-theoretic rough set models is obtained by separating the Bayesian confirmation part the parameterized model of Greco *et al.* [7–9]. An in-depth understanding of the semantics differences between the two models enables us to reveal two different aspects of Bayesian reasoning with rough sets. Decision-theoretic rough sets can be used for Bayesian classification and confirmation-theoretic rough sets can be used for weighting or evaluating evidence. Their integration may lead to a better and unified probabilistic rough set model that is capable of selecting equivalence relations on the one hand and producing approximations on the other.

2 Pawlak and Probabilistic Models

This section summarize the main results of Pawlak rough sets and probabilistic rough sets.

2.1 Pawlak Rough Set Model

In Pawlak rough set model [17], information about a finite set of objects is represented in an information table with a finite set of attributes. Formally, an information table can be expressed as: $S = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$, where U is a finite nonempty set of objects called the universe, At is a finite nonempty set of attributes, V_a is a nonempty set of values for $a \in At$, and $I_a : U \to V_a$ is an information function. The information function I_a maps an object in U to a value of V_a for an attribute $a \in At$, that is, $I_a(x) \in V_a$. For a subset of attributes $A \subseteq At$, $I_A(x)$ denotes the values of x on A.

Rough sets are contracted based on equivalence relations induced subsets of attributes. Given a subset of attributes $A \subseteq At$, an indiscernibility relation on U, R_A, or simply R, is defined as [17]:

$$\begin{aligned} xRy &\iff \forall_{a \in A} I_a(x) = I_a(y) \\ &\iff I_A(x) = I_A(y). \end{aligned}$$

The relation R is an equivalence relation, that is, R is reflexive, symmetric and transitive. Two objects x and y in U are equivalent or indiscernible by the set of attributes A if and only if they have the same values on all attributes in A. The equivalence class containing x is given by:

$$[x] = \{y \in U \mid xRy\}.$$

The equivalence relation R induces a partition of U, denoted by $U/R = \{[x] \mid x \in U\}$.

The equivalence classes in U/R are the building blocks to construct rough set approximations. For a subset $C \subseteq U$, the lower and upper approximations of C with respect to U/R are defined by [17]:

$$\begin{aligned} \underline{apr}(C) &= \{x \in U \mid [x] \subseteq C\} \\ &= \bigcup \{[x] \in U/R \mid [x] \subseteq C\}; \\ \overline{apr}(C) &= \{x \in U \mid [x] \cap C \neq 0\} \\ &= \bigcup \{[x] \in U/R \mid [x] \cap C \neq \emptyset\}. \end{aligned} \tag{1}$$

The lower approximation is the union of equivalence classes that are included in C, and the upper approximation is the union of equivalence classes that have an non-empty overlap with C. Based on the rough set approximations of C, one can divide the universe U into three pair-wise disjoint regions [17]:

$$\begin{aligned} &\mathrm{POS}(C) = \underline{apr}(C), \\ &\mathrm{NEG}(C) = U - \overline{apr}(C) = (\overline{apr}(C))^c, \\ &\mathrm{BND}(C) = \overline{apr}(C) - \underline{apr}(C), \end{aligned} \tag{2}$$

where $(\cdot)^c$ is the set complement. The positive region $\mathrm{POS}(C)$ is the lower approximation, the negative region $\mathrm{NEG}(C)$ is the complement of the upper approximation, and the boundary region $\mathrm{BND}(C)$ is the difference between the upper and lower approximations. It can be verified that $\mathrm{NEG}(C) = \mathrm{POS}(C^c)$.

2.2 Decision-Theoretic Rough Set Models

The positive and negative regions of a set in Pawlak rough sets must be completely certain. An equivalence class is in the positive region if and only if it is fully contained in the set. An equivalence class is in the negative region if and only if it has an empty intersection with the set. This may be too restrictive to be practically useful in real applications. By allowing some level of uncertainty in the positive and negative regions, decision-theoretic rough sets [30, 31] use conditional probability $Pr(X|[x])$ to quantify the degree of overlap between equivalence classes $[x]$ and a set C. The conditional probability is the the probability that an object belongs to C given that the object is in $[x]$. Accordingly, the Pawlak three regions can be equivalently defined by [25, 28]:

$$\begin{aligned} \text{POS}(C) &= \{x \in U \mid Pr(C|[x]) = 1\}, \\ \text{BND}(C) &= \{x \in U \mid 0 < Pr(C|[x]) < 1\}, \\ \text{NEG}(C) &= \{x \in U \mid Pr(C|[x]) = 0\}. \end{aligned} \tag{3}$$

They are defined by using the two extreme values, 0 and 1, of probabilities. They are of a qualitative nature; the magnitude of the value $Pr(C|[x])$ is not taken into account.

The main result of decision-theoretic rough sets [30, 31] is the introduction of a pair of parameters α and β to replace 1 and 0, respectively:

$$\begin{aligned} \text{POS}_{(\alpha,\beta)}(C) &= \{x \in U \mid Pr(C|[x]) \geq \alpha\}, \\ \text{BND}_{(\alpha,\beta)}(C) &= \{x \in U \mid \beta < Pr(C|[x]) < \alpha\}, \\ \text{NEG}_{(\alpha,\beta)}(C) &= \{x \in U \mid Pr(C|[x]) \leq \beta\}, \end{aligned} \tag{4}$$

where the pair of thresholds satisfies the condition $\alpha > \beta$, ensuring that the three regions are pair-wise disjoint. The pair of thresholds (α, β) can be determined and interpreted from the loss or cost of various decisions using Bayesian decision theory.

Unlike the qualitative Pawlak approximations, probabilistic approximations introduce certain levels of error in both the positive and boundary regions. Pawlak regions and (α, β)-probabilistic regions are linked together by:

$$\begin{aligned} &\text{POS}(C) \subseteq \text{POS}_{(\alpha,\beta)}(C), \\ &\text{BND}_{(\alpha,\beta)}(C) \subseteq \text{BND}(C), \\ &\text{NEG}(C) \subseteq \text{NEG}_{(\alpha,\beta)}(C). \end{aligned} \tag{5}$$

Probabilistic three regions may be interpreted in terms of costs of different types of classification decisions [26, 27]. One obtains larger positive and negative regions by introducing classification errors in trade of a smaller boundary region so that the total classification cost is minimum. Considering the errors introduced, the three regions are semantically interpreted as the following three-way decisions [26, 27, 29]. We accept an object x to be a member of C if the conditional probability is greater than or equal to α, with an understanding that it comes

with an $(1-\alpha)$-level acceptance error and associated cost. We reject x to be a member of C if the conditional probability is less than or equal to β, with an understanding that it comes with an β-level of rejection error and associated cost. We neither accept nor reject x to be a member of C if the conditional probability is between of α and β, instead, we make a decision of deferment. The boundary region does not involve acceptance and rejection errors, but it is associated with cost of deferment. The three probabilistic regions are obtained by considering a trade-off between various classification costs.

2.3 Confirmation-Theoretic Rough Set Models

Greco *et al.* [7] introduced a parameterized rough set model by considering a pair of thresholds on a Bayesian confirmation measure, in addition to a pair of thresholds on probability. The Bayesian confirmation measure is denoted by $c([x], C)$ which indicates the degree to which an equivalence class $[x]$ confirms the hypothesis C. Given a Bayesian confirmation measure $c([x], C)$ and a pair of thresholds (s, t) with $t < s$, three (α, β, s, t)-parameterized regions are defined by:

$$\begin{aligned}
\mathrm{PPOS}_{(\alpha,\beta,s,t)}(C) &= \{x \in U \mid Pr(C|[x]) \geq \alpha \wedge c([x], C) \geq s\}, \\
\mathrm{PBND}_{(\alpha,\beta,s,t)}(C) &= \{x \in U \mid (Pr(C|[x]) > \beta \vee c([x], C) > t) \wedge \\
&\qquad (Pr(C|[x]) < \alpha \vee c([x], C) < s)\}, \\
\mathrm{PNEG}_{(\alpha,\beta,s,t)}(C) &= \{x \in U \mid Pr(C|[x]) \leq \beta \wedge c([x], C) \leq t\}. \qquad (6)
\end{aligned}$$

There is no general agreement on a Bayesian confirmation measure. Choosing an appropriate confirmation measure for a particular application may not be an easy task. The ranges of the values of different confirmation measures are different. This makes it an even more difficult task to interpret and set the thresholds (s, t).

Although the use of two pairs of thresholds provides additional flexibility of a probabilistic rough set model, there is still a lack of framework on how to interpret the interactions and trade-off between the four thresholds. For this reason, we consider a simple confirmation-theoretic model that produces the following three regions:

$$\begin{aligned}
\mathrm{CPOS}_{(s,t)}(C) &= \{[x] \in U/R \mid c([x], C) \geq s\}, \\
\mathrm{CBND}_{(s,t)}(C) &= \{[x] \in U/R \mid t < c([x], C) < s)\}, \\
\mathrm{CNEG}_{(s,t)}(C) &= \{[x] \in U/R \mid c([x], C) \leq t\}. \qquad (7)
\end{aligned}$$

In contrast to Greco *et al.*'s model, we divide the partition U/R, instead of the universe, into three regions. Each equivalence class may be viewed as a piece of evidence. An equivalence class in the positive region supports C to a degree at or above s, an equivalence class in the negative region supports to a degree at or below t and may be viewed as against C, and an equivalence class in the boundary region is interpreted as neutral towards C.

2.4 Bayesian Rough Set Models

Instead of using arbitrary pairs of thresholds, Ślęzak and Ziarko [21, 22] suggested the use of the *a priori* probability $Pr(C)$ as a threshold. They introduced a Bayesian rough set (BRS) model that divides U into three regions as follows:

$$\begin{aligned}
\mathrm{POS}_B(C) &= \{x \in U \mid Pr(C|[x]) > Pr(C)\},\\
\mathrm{BND}_B(C) &= \{x \in U \mid Pr(C|[x]) = Pr(C)\},\\
\mathrm{NEG}_B(C) &= \{x \in U \mid Pr(C|[x]) < Pr(C)\}.
\end{aligned} \tag{8}$$

In this way, the Bayesian rough sets may be related to decision-theoretic rough sets in which $\alpha = \beta = Pr(C)$. Alternatively, one may interpret $Pr(C|[x])-Pr(C)$ as a confirmation measure. In this case, by setting $s = t = 0$, one can establish a connection to confirmation-theoretic rough sets.

When neither the *a posteriori* probability $Pr(C|[x])$ nor the *a priori* probability $Pr(C)$ is derivable from data, one may compare two likelihood functions $Pr([x]|C)$ and $Pr([x]|C^c)$ directly [21, 22]. That is,

$$\begin{aligned}
\mathrm{POS}_B(C) &= \{x \in U \mid Pr([x]|C) > Pr([x]|C^c)\},\\
\mathrm{BND}_B(C) &= \{x \in U \mid Pr([x]|C) = Pr([x]|C^c)\},\\
\mathrm{NEG}_B(C) &= \{x \in U \mid Pr([x]|C) < Pr([x]|C^c)\}.
\end{aligned} \tag{9}$$

Ślęzak [20] further drew a natural correspondence between the fundamental notions of rough sets and statistics. The set to be approximated corresponds to a hypothesis and an equivalence class to a piece of evidence; the three probabilistic regions correspond to the cases that the hypothesis is verified positively, negatively, or undecided based on the evidence. Based on such a correspondence, Ślęzak introduced a rough Bayesian model [20], in which probabilistic approximations are defined based on a pair of thresholds on the ratio of the *a priori* and the *a posteriori* probabilities.

3 Interpreting Two Probabilistic Models

From the viewpoint of semantics, we examine differences between decision-theoretic and confirmation-theoretic models and their corresponding applications.

3.1 Semantics Issues of Probabilistic Rough Set Models

A key to unlocking the differences and scopes of various probabilistic rough set models may be the semantics of these models. A probabilistic rough set model must address at least the following three issues:

(i) Interpretation and computation of thresholds;
(ii) Estimation of conditional probability $Pr(C|[x])$;
(iii) Interpretation and applications of three regions in data analysis.

Table 1. Comparison of probabilistic rough set models

RS models	main features	unsolved/partially solved issues
Decision-theoretic RS model	a pair of thresholds	(ii)
Confirmation-theoretic RS model	a pair of thresholds	(i), (ii) & (iii)
Variable precision RS model	one threshold or a pair of thresholds	(i), (ii) & (iii)
Parameterized RS model	two pairs of thresholds	(i), (ii) & (iii)
Bayesian RS model	*a priori* probability as threshold	(ii) & (iii)

Table 1 lists the issues that have not been fully solved in each of the probabilistic rough set models. The main results related to the three issues are summarized below.

Issue (i): Interpretation and computation of thresholds. The original decision-theoretic rough set model is the only model that fully considers issue (i) by giving a sound theoretical and practical basis for interpreting and computing the required three threshold. Several more recent attempts include a game-theoretic framework [1, 12], a cost-sensitive model of decision makeing [14], a model based on an optimization viewpoint [13], a method using probabilistic model criteria [16], and an information-theoretic framework [4].

Issue (ii): Estimation of conditional probabilities. The required conditional probability is commonly estimated as:

$$Pr(C|[x]) = \frac{|C \cap [x]|}{|[x]|}, \tag{10}$$

where $|\cdot|$ denotes the cardinality of a set. The conditional probability defined in this way is also known as a rough membership function [18]. This simple way of estimation is of limited value due to the requirement of a large-sized sample. Dembczyński *et al.* [3] suggested a statistical model in which probabilities are estimated based on the maximization of a likelihood function. In the naive Bayesian rough set (NBRS) model introduced by Yao and Zhou [32], the estimation of the *a posteriori* probability is translated into the estimation of the likelihood function based on Bayes' theorem and naive conditional independence assumption. Liu et al. [15] used logistic regression method for estimating the required conditional probability.

Issue (iii): Interpretation and application of three regions. Each of the probabilistic models introduces three approximation regions. Although three regions are similar in form, they have different semantics interpretations. More specifically, the three regions defined, respectively, by Ślęzak *et al.* [21, 22] and Greco *et al.* [7–9] have a very different interpretation from those of the decision-theoretic rough set models. It may not be appropriate to interpret the former as probabilistic approximations of C. Rather, they are interpreted as classification of pieces of evidence (i.e., equivalence classes).

The interpretation and application of three regions for real world applications remain to be partially unsolved. A recently proposed theory of three-way decisions [29] for interpreting three regions is a promising direction.

3.2 Applications of the Two Models

We discuss two applications of Bayesian inference with probabilistic rough set models. Decision-theoretic rough set models [30, 31] concern drawing conclusions based on available evidence. Confirmation-theoretic rough set models focus on evaluating pieces of evidence.

Bayesian inference uses probability for quantifying uncertainty in inferences. In Bayesian data analysis, the *a priori* probability of a hypothesis is updated into the *a posteriori* probability after observing some evidence [2]. Bayesian inference can be used to address two different issues and, hence, two types of applications. First, we use the degree to which evidence supports a hypothesis to classify objects based on their satisfiability of the hypothesis. That is, we classify an object as satisfying or not satisfying the hypothesis if the object positively supports or is against the hypothesis beyond a certain level. Second, we can evaluate the quality of different pieces of evidence (i.e., how much the *a posteriori* probability increases or reduces after observing the evidence). That is, we can either weigh or select pieces of evidence according to their confirmations of the hypothesis.

An understanding of the semantics differences behind these two applications enables us to demonstrate two types of probabilistic rough set models. The main ideas of the two applications of Bayesian inference are illustrated by examples. Suppose we have a data table from a hospital historical database. In this table, there are a set of patients and a set of attributes indicating patients' symptoms (e.g., cough) with regard to a certain disease (e.g., lung cancer).

The first application concerns the diagnosis. Given a patient with certain symptoms (i.e., the evidence), what are the chances that the patient has lung cancer (i.e., the hypothesis)?

In an ideal case, the information in the data table is complete, the probability of the patient has lung cancer can be estimated by the number of patients who have lung cancer and symptom cough divides the number of people who has symptom cough. In many cases, we may only have limited information on hand. Can we still predict the probability? Bayesian inference provides an answer to this question. Suppose that we have some prior knowledge about lung cancer (i.e., the probability of an arbitrary person having lung cancer). When the doctor sees a new patient, he/she receives evidence (i.e., cough) about lung cancer. The evidence is related to lung cancer by a conditional probability, called likelihood. Bayes' theorem, also called Bayes' law or Bayes' rule named after Thomas Bayes, offered a solution for this problem. In Bayes' theorem, the *a posteriori* probability can be calculated from the *a priori* probability and the likelihood function,

$$Pr(lung\ cancer|cough) = \frac{Pr(cough|lung\ cancer) \cdot Pr(lung\ cancer)}{Pr(cough)},$$

where $Pr(lung\ cancer)$ is the *a priori* probability of an arbitrary people with lung cancer. $Pr(lung\ cancer \mid cough)$ is the *a posteriori* probability that a patient with lung cancer after observing evidence cough, and $Pr(cough \mid lung\ cancer)$ is the likelihood of evidence cough related to lung cancer.

Once we obtain the *a posteriori* probability, how do we make decisions based on the value of the *a posteriori* probability? Rough set theory provides us a way for three-way decision making [27]. When applying Bayesian inference to rough sets, one may view the set C as a hypothesis that an object is in C and an equivalence class as evidence that an object is in the equivalence class. This immediately leads to the definition of three probabilistic regions defined in decision-theoretic rough set models (equation (6)). They can be used to build a ternary classifier for three-way decisions. The doctor can make a decisions of treatment when the probability is greater than or equal to α, namely, $Pr(lung\ cancer|cough) \geq \alpha$, and of not treatment when $Pr(lung\ cancer|cough) \leq \beta$. In the case when the probability lies in between α and β, the doctor can perform a medical test to further examine the patient.

The second application concerns which symptoms or medical tests provide more information when diagnosing a disease. A doctor may decide to perform a particular test in order to revise the *a priori* probability in the process of diagnosing and treating lung cancer.

For example, if the probability of a patient has lung cancer given that he/she has cough increased from the *a priori* probability (i.e., the probability without seeing any evidence), then cough is considered as supporting evidence for lung cancer. If there are many possible tests that may be used, how do we decide which one is more informative? Assume there are two tests that can be performed. Consider a Bayesian confirmation measure defined by [5, 6] $Pr(C|[x]) - Pr(C)$. The three probabilistic regions are defined in confirmation-theoretic rough set models as follows [9, 21, 22],

$$\begin{aligned}
\mathrm{CPOS}_{(s,t)}(C) &= \{x \in U \mid Pr(C|[x]) - Pr(C) \geq s\},\\
\mathrm{CBND}_{(s,t)}(C) &= \{x \in U \mid t < Pr(C|[x]) - Pr(C) < s\},\\
\mathrm{CNEG}_{(s,t)}(C) &= \{x \in U \mid Pr(C|[x]) - Pr(C) \leq t\},
\end{aligned}$$

where (s, t) is a pair of thresholds and $s > t$. The results of a Bayesian confirmation model also offer three-way decisions for evaluating symptoms or tests. If $Pr(C|[x]) - Pr(C) \geq s$, the symptoms of x support C or these tests should be performed to confirm that x has lung cancer. If $Pr(C|[x]) - Pr(C) \leq t$, the symptoms of x are against C or these tests should be performed to rule out x has lung cancer. If $t < Pr(C|[x]) - Pr(C) < s$, the symptoms of x are neutral to C or these tests are not informative.

Alternatively, the value of a Bayesian confirmation measure can be used to help the doctor to decide which medical test should be performed. Based on the

historical data, the results of $Test_1$ provide a better indication that a patient has lung cancer than the results of $Test_2$ if

$$Pr(lung\ cancer|Test_1) - Pr(lung\ cancer) > Pr(lung\ cancer|Test_2) - Pr(lung\ cancer),$$

a doctor might want to perform $Test_1$ instead of $Test_2$.

4 Conclusions

Probabilistic rough set models can be categorized based on two types of applications. The first application is to classify objects based on their satisfiability of the hypothesis. The second application is to evaluate the quality of different pieces of evidence. Our comparison results show that although these two models of probabilistic rough sets are similar in forms, they have very different semantics interpretations and therefore lead to different applications. As future research, we will investigate these two types of applications.

Acknowledgements. This work is partially supported by an NSERC Discovery Grant.

References

1. Azam, N., Yao, J.T.: Multiple criteria decision analysis with game-theoretic rough sets. 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. 399–408. Springer, Heidelberg (2012)
2. Bayes, T., Price, R.: An essay towards solving a problem in the doctrine of chance. Philosophical Transactions of the Royal Society of London 53, 370–418 (1763)
3. Dembczyński, K., Greco, S., Kotłowski, W., Słowiński, R.: Statistical model for rough set approach to multicriteria classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 164–175. Springer, Heidelberg (2007)
4. Deng, X., Yao, Y.Y.: An information-theoretic interpretation of thresholds in probabilistic rough sets. 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. 369–378. Springer, Heidelberg (2012)
5. Festa, R.: Bayesian Confirmation. In: Galavotti, M., Pagnini, A. (eds.) Experience, Reality, and Scientific Explanation, pp. 55–87. Kluwer Academic Publishers, Dordrecht (1999)
6. Fitelson, B.: Studies in Bayesian Confirmation Theory. Ph.D. Dissertation, University of Wisconsin (2001), `http://fitelson.org/thesis.pdf`
7. Greco, S., Pawlak, Z., Słowiński, R.: Bayesian confirmation measures within rough set approach. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 264–273. Springer, Heidelberg (2004)

8. Greco, S., Matarazzo, B., Słowiński, R.: Rough membership and Bayesian confirmation measures for parameterized rough sets. In: Ślęzak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 314–324. Springer, Heidelberg (2005)
9. Greco, S., Matarazzo, B., Słowiński, R.: Parameterized rough set model using rough membership and Bayesian confirmation measures. International Journal of Approximate Reasoning 49, 285–300 (2009)
10. Grzymala-Busse, J.W.: Generalized probabilistic approximations. In: Peters, J.F., Skowron, A., Ramanna, S., Suraj, Z., Wang, X. (eds.) Transactions on Rough Sets XVI. LNCS, vol. 7736, pp. 1–16. Springer, Heidelberg (2013)
11. Grzymala-Busse, J.W., Yao, Y.Y.: Probabilistic rule induction with the LERS data mining system. International Journal of Intelligent Systems 26, 518–539 (2011)
12. Herbert, J.P., Yao, J.T.: Game-theoretic rough sets. Fundamenta Informaticae 108, 267–286 (2011)
13. Jia, X.Y., Liao, W.H., Tang, Z.M., Shang, L.: Minimum cost attribute reduction in decision-theoretic rough set models. Information Sciences 219, 151–167 (2013)
14. Li, H.X., Zhou, X.Z.: Risk decision making based on decision-theoretic rough set: A three-way view decision model. International Journal of Computational Intelligence Systems 4, 1–11 (2011)
15. Liu, D., Li, T.R., Liang, D.C.: Incorporating logistic regression to decision-theoretic rough sets for classifications. International Journal of Approximate Reasoning (2013), http://dx.doi.org/10.1016/j.ijar.2013.02.013
16. Liu, D., Li, T.R., Ruan, D.: Probabilistic model criteria with decision-theoretic rough sets. Information Science 181, 3709–3722 (2011)
17. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
18. Pawlak, Z., Skowron, A.: Rough membership functions. In: Yager, R.R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in the Dempster-Shafer Theory of Evidence, John Wiley and Sons, New York, pp. 251–271. John Wiley and Sons, Chichester (1994)
19. Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough sets: Probabilistic versus deterministic approach. International Journal of Man-Machine Studies 29, 81–95 (1988)
20. Ślęzak, D.: Rough sets and Bayes factor. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 202–229. Springer, Heidelberg (2005)
21. Ślęzak, D., Ziarko, W.: Bayesian rough set model. Proceedings of FDM 2002, pp. 131–135 (2002)
22. Ślęzak, D., Ziarko, W.: The investigation of the Bayesian rough set model. International Journal of Approximate Reasoning 40, 81–91 (2005)
23. Wong, S.K.M., Ziarko, W.: A probabilistic model of approximate classification and decision rules with uncertainty in inductive learning. Technical Report CS-85-23, Department of Computer Science, University of Regina (1985)
24. Yao, Y.Y.: Probabilistic approaches to rough sets. Expert Systems 20, 287–297 (2003)
25. Yao, Y.Y.: Probabilistic rough set approximations. International Journal of Approximation Reasoning 49, 255–271 (2008)
26. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180, 341–353 (2010)
27. Yao, Y.Y.: The superiority of three-way decisions in probabilistic rough set models. Information Sciences 181, 1080–1096 (2011)
28. Yao, Y.Y.: Two semantic issues in a probabilistic rough set model. Fundamenta Informaticae 108, 249–265 (2011)

29. Yao, Y.Y.: An outline of a theory of three-way decisions. In: Yao, J., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 1–17. Springer, Heidelberg (2012)
30. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximating concepts. International Journal of Man-machine Studies 37, 793–809 (1992)
31. Yao, Y.Y., Wong, S.K.M., Lingras, P.: A decision-theoretic rough set model. In: Ras, Z.W., Zemankova, M., Emrich, M.L. (eds.) Methodologies for Intelligent Systems, vol. 5, pp. 17–24. North-Holland, Amsterdam (1990)
32. Yao, Y.Y., Zhou, B.: Naive Bayesian rough sets. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS, vol. 6401, pp. 719–726. Springer, Heidelberg (2010)
33. Ziarko, W.: Variable precision rough sets model. Journal of Computer and Systems Sciences 46, 39–59 (1993)

Empirical Risk Minimization for Variable Precision Dominance-Based Rough Set Approach

Yoshifumi Kusunoki[1], Jerzy Błaszczyński[2], Masahiro Inuiguchi[3], and Roman Słowiński[2,4]

[1] Graduate School of Engineering, Osaka University, 2-1, Yamadaoka, Suita, Osaka 565-0871, Japan
kusunoki@eei.eng.osaka-u.ac.jp
[2] Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland
{jblaszczynski,rslowinski}@cs.put.poznan.pl
[3] Graduate School of Engineering Science, Osaka University, 1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan
inuiguti@sys.es.osaka-u.ac.jp
[4] Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland

Abstract. In this paper, we characterize Variable Precision Dominance-based Rough Set Approach (VP-DRSA) from the viewpoint of empirical risk minimization. VP-DRSA is an extension of the Dominance-based Rough Set Approach (DRSA) that admits some degree of misclassification error. From a definable set, we derive a classification function, which indicates assignment of an object to a decision class. Then, we define an empirical risk associated with the classification function. It is defined as mean hinge loss function. We prove that the classification function minimizing the empirical risk function corresponds to the lower approximation in VP-DRSA.

Keywords: rough sets, variable precision dominance-based rough set approach, empirical risk minimization.

1 Introduction

Rough set theory [5] provides a framework for data analysis under partial inconsistency. An analysed data set has the form of a decision table, which consists of objects described by condition attributes and classified into a finite number of decision classes. When objects having the same description by condition attributes are classified into different decision classes, the classification is considered as inconsistent with respect to the condition attributes. The inconsistency is a key issue in the definition of lower and upper (rough) approximations of a decision class, which are consistent sets of certainly and possibly classified objects into the decision class, respectively.

There are two important extensions of rough set theory. One is Variable Precision Rough Set Model (VP-RSM) [9], which admits some degree of misclassification error in the definition of lower approximations. In VP-RSM, for a decision

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 133–144, 2013.

class and an object, a membership degree of the object to the decision class is defined. It is called rough membership degree. The lower approximation of the decision class is defined by the set of objects whose rough membership degrees are not less than a given threshold. The other extension is the Dominance-based Rough Set Approach (DRSA) [2]. In DRSA, the values of condition attributes and the decision classes are totally ordered, and inconsistency with respect to monotonicity between the ordered condition attributes and the decision classes is captured. In DRSA, instead of single decision classes, lower and upper approximations concern upward and downward unions of decision classes. Moreover, there are two variable-precision-like extensions of DRSA: Variable Consistency DRSA (VC-DRSA) [1] and Variable Precision DRSA (VP-DRSA) [4]. They are based on different definitions of membership degrees.

Several authors [6–8] pointed out that the rough membership degree in VP-RSM can be interpreted as conditional probability. This fact stimulated development of Decision-Theoretic Rough Set Model (DTRSM) [8]. In DTRSM, approximations are determined by Bayes risk minimization of statistical decision theory. The membership degree in VC-DRSA can be also interpreted as conditional probability [3]. However, in VP-DRSA, the membership degree, which has some desirable properties, cannot be interpreted by conditional probability.

Defining approximations can be seen as a classification problem in statistical learning theory. It consists in finding a classification function, which indicates assignment of an object to a decision class. The best function is selected from predefined category of functions by minimizing its empirical risk for a decision table with respect to a specific loss function. In this paper, we characterize the approximations of VP-DRSA from the viewpoint of empirical risk minimization. We define classification functions corresponding to consistent sets of objects, and define an empirical risk function for classification functions. We prove that the classification function corresponding to the lower approximation minimizes the empirical risk function.

This paper is organized as follows. In Section 2, VP-DRSA is briefly introduced. The membership degree and the lower and upper approximations are defined. In Section 3, we define the empirical risk function, and characterize the lower approximation from the viewpoint of risk minimization. Concluding remarks are given in Section 4.

2 Variable Precision Dominance-Based Rough Set Approach

2.1 Decision Table and DRSA

Dominance-based Rough Set Approach (DRSA) [2] is an extension of rough set approach which involves dominance relation instead of the usual indiscernibility relation in the treatment of ordinal data organized in a decision table. Let us recall briefly DRSA and related topics for the sake of introduction.

A decision table is defined by $(U, AT = C \cup \{d\}, V)$, where $U = \{u_1, ..., u_n\}$ is a finite set of objects, $C = \{c_1, ..., c_m\}$ is a finite set of condition attributes, d is a

decision attribute, and V is a set of attribute values. For each $u \in U$ and $a \in AT$, $a(u) \in V$ is an attribute value of u with respect to a. We denote by $V_a \subseteq V$ a set of attribute values with respect to a. For $A \subseteq AT$, let $V_A = \prod_{a \in A} V_a$. The attribute set is divided into AT_N and AT_C. Attributes in AT_C are called criteria. For a criterion $a \in AT_C$, we assume a total order $\geq$ on its value set V_a. Moreover, we assume that all attributes from AT_C are of the gain-type, i.e., the greater the better. The decision attribute assigns each object to one of totally ordered decision classes specified by V_d: as such, it may be considered as a criterion.

For $A \subseteq C$, a dominance relation D_A on U is defined by:

$$D_A = \left\{ (u, u') \in U^2 \;\middle|\; \begin{array}{l} a(u) \geq a(u'), \forall a \in AT_C \cap A \\ \text{and } a(u) = a(u'), \forall a \in AT_N \cap A \end{array} \right\}. \tag{1}$$

D_A satisfies reflexivity and transitivity. For $u \in U$, its dominating set and its dominated set are defined, respectively, by:

$$D_A^+(u) = \{u' \in U \mid (u', u) \in D_A\},\ D_A^-(u) = \{u' \in U \mid (u, u') \in D_A\}. \tag{2}$$

Let $V_d = \{1, 2, ..., p\}$, and assume a gain-type preference order in this value set, as $1 < 2 < \cdots < p$. Decision attribute d partitions U into $\{X_1, X_2, \ldots, X_p\}$, each of which is called a decision class, where $X_i = \{u \in U \mid d(u) = i\}$. Since decision classes are ordered $X_1 < X_2 < \cdots < X_p$, one can define an upward union of decision classes $X_i^{\geq}$ and a downward union of decision classes $X_i^{\leq}$ with respect to each class X_i, $i = 1, 2, \ldots, p$, as follows:

$$X_i^{\geq} = \bigcup_{j \geq i} X_j,\ X_i^{\leq} = \bigcup_{j \leq i} X_j. \tag{3}$$

For convenience, $X_0^{\leq} = X_{p+1}^{\geq} = \emptyset$. We have $X_i^{\geq} = \neg X_{i-1}^{\leq}$, where $\neg X$ is the complement set of $X \subseteq U$.

In DRSA, it is supposed that if an object u is better than or equal to another object u' with respect to all condition attributes, then the class of u should not be worse than that of u'. This is called the dominance principle. Given a decision table, the inconsistency with respect to the dominance principle is captured by the difference between upper and lower approximations of the unions of decision classes. Given a condition attribute set $A \subseteq C$, and $i \in \{1, 2, \ldots, p\}$, the lower approximation $\underline{A}(X_i^{\geq})$ of $X_i^{\geq}$ and the upper approximation $\overline{A}(X_i^{\geq})$ of $X_i^{\geq}$ are defined, respectively, by:

$$\underline{A}(X_i^{\geq}) = \{u \in U \mid D_A^+(u) \subseteq X_i^{\geq}\},\ \overline{A}(X_i^{\geq}) = \{u \in U \mid D_A^-(u) \cap X_i^{\geq} \neq \emptyset\}. \tag{4}$$

Similarly, the lower approximation $\underline{A}(X_i^{\leq})$ of $X_i^{\leq}$ and upper approximation $\overline{A}(X_i^{\leq})$ of $X_i^{\leq}$ are defined, respectively, by:

$$\underline{A}(X_i^{\leq}) = \{u \in U \mid D_A^-(u) \subseteq X_i^{\leq}\},\ \overline{A}(X_i^{\leq}) = \{u \in U \mid D_A^+(u) \cap X_i^{\leq} \neq \emptyset\}. \tag{5}$$

2.2 VP-DRSA

For $A \subseteq C$, $X_i^{\geq}$, $X_i^{\leq}$, $i \in \{1, 2, \ldots, p\}$, and $u \in U$, we define a membership degree of x in $X_i^{\geq}$ and in $X_i^{\leq}$ with respect to A by:

$$\mu^A_{X_i^{\geq}}(u) = \frac{|D_A^-(u) \cap X_i^{\geq}|}{|D_A^-(u) \cap X_i^{\geq}| + |D_A^+(u) \cap X_{i-1}^{\leq}|}, \tag{6}$$

$$\mu^A_{X_i^{\leq}}(u) = \frac{|D_A^+(u) \cap X_i^{\leq}|}{|D_A^+(u) \cap X_i^{\leq}| + |D_A^-(u) \cap X_{i+1}^{\geq}|}. \tag{7}$$

These membership degrees have been used to define lower and upper approximations in the Variable Precision DRSA (VP-DRSA) [4]. For a complementary pair $X_i^{\geq}$ and $X_{i-1}^{\leq}$, it is clear that $\mu^A_{X_i^{\geq}}(u) + \mu^A_{X_{i-1}^{\leq}}(u) = 1$. When D_A is symmetric, i.e., $A \subseteq AT_N$, (6) and (7) are reduced to the rough membership degree of the Variable Precision Rough Set Model (VP-RSM) [9]. Namely, in this case, we have $D_A(u) = D_A^+(u) = D_A^-(u)$, and thus we obtain $\mu^A_{X_i^{\geq}}(u) = \frac{|D_A(u) \cap X_i^{\geq}|}{|D_A(u)|}$, $\mu^A_{X_i^{\leq}}(u) = \frac{|D_A(u) \cap X_i^{\leq}|}{|D_A(u)|}$.

The membership degrees are kinds of consistency measures. In [1], monotonicity properties required for consistency measures were proposed. For $A \subseteq C$, $X_i^{\geq}$, $X_i^{\leq}$ and $u \in U$, let $f^A_{X_i^{\geq}}(u)$ and $f^A_{X_i^{\leq}}(u)$ be gain type consistency measures. The monotonicity properties are defined as follows.

(m1). Let $X_i^{\geq}$, $X_i^{\leq}$ and $u \in U$ be given. For $B \subseteq A \subseteq C$, it holds that $f^B_{X_i^{\geq}}(u) \leq f^A_{X_i^{\geq}}(u)$ and $f^B_{X_i^{\leq}}(u) \leq f^A_{X_i^{\leq}}(u)$.

(m2). Let $A \subseteq C$, $X_i^{\geq}$, $X_i^{\leq}$ and $u \in U$ be given. If new objects $\Delta X_i^{\geq}$ are assigned to $X_i^{\geq}$ then $f^A_{X_i^{\geq}}(u) \leq f^A_{X_i^{\geq} \cup \Delta X_i^{\geq}}(u)$. Contrarily, if new objects $\Delta X_i^{\leq}$ are assigned to $X_i^{\leq}$ then $f^A_{X_i^{\leq}}(u) \leq f^A_{X_i^{\leq} \cup \Delta X_i^{\leq}}(u)$.

(m3). Let $A \subseteq C$ and $u \in U$ be given. For $X_i^{\geq}$, $X_j^{\geq}$, $X_i^{\leq}$, $X_j^{\leq}$ such that $i \leq j$, it holds that $f^A_{X_i^{\geq}}(u) \geq f^A_{X_j^{\geq}}(u)$ and $f^A_{X_i^{\leq}}(u) \leq f^A_{X_j^{\leq}}(u)$.

(m4). Let $A \subseteq C$, $X_i^{\geq}$, $X_i^{\leq}$ be given. For u, u' such that uD_Au', it holds that $f^A_{X_i^{\geq}}(u) \geq f^A_{X_i^{\geq}}(u')$ and $f^A_{X_i^{\leq}}(u) \leq f^A_{X_i^{\leq}}(u')$.

The membership degrees $\mu^A_{X_i^{\geq}}$ and $\mu^A_{X_i^{\leq}}$ satisfy properties (m2), (m3) and (m4), but not (m1).

Now, consider approximations of VP-DRSA. Let $0 \leq \beta < \alpha \leq 1$ be precision parameters. For an upward union of classes $X_i^{\geq}$, the lower approximation $\underline{A}(X_i^{\geq}|\alpha)$ and the upper approximation $\overline{A}(X_i^{\geq}|\beta)$ are defined, respectively, by:

$$\underline{A}(X_i^{\geq}|\alpha) = \{u \in U \mid \mu^A_{X_i^{\geq}}(u) \geq \alpha\}, \tag{8}$$

$$\overline{A}(X_i^{\geq}|\beta) = \{u \in U \mid \mu^A_{X_i^{\geq}}(u) > \beta\}. \tag{9}$$

Similarly, the lower and upper approximations of downward union of classes $X_i^{\leq}$ are defined:

$$\underline{A}(X_i^{\leq}|\alpha) = \{u \in U \mid \mu^A_{X_i^{\leq}}(u) \geq \alpha\}, \tag{10}$$

$$\overline{A}(X_i^{\leq}|\beta) = \{u \in U \mid \mu^A_{X_i^{\leq}}(u) > \beta\}. \tag{11}$$

We can easily prove dual properties of the lower and upper approximations, such that $\overline{A}(X_i^{\geq}|\beta) = \neg\underline{A}(X_{i-1}^{\leq}|1-\beta)$ and $\overline{A}(X_i^{\leq}|\beta) = \neg\underline{A}(X_{i+1}^{\geq}|1-\beta)$ when $\beta < 0.5$. From monotonicity property (m4) of $\mu^A_{X_i^{\geq}}$ and $\mu^A_{X_i^{\geq}}$, these approximations are upward and downward definable sets, respectively [3], where a set X represented by a union of dominating (dominated) sets of $u \in X$ is called an upward (downward) definable set.

$$\underline{A}(X_i^{\geq}|\alpha) = \bigcup_{u \in \underline{A}(X_i^{\geq}|\alpha)} D_A^+(u), \qquad \overline{A}(X_i^{\geq}|\beta) = \bigcup_{u \in \overline{A}(X_i^{\geq}|\beta)} D_A^+(u), \tag{12}$$

$$\underline{A}(X_i^{\leq}|\alpha) = \bigcup_{u \in \underline{A}(X_i^{\leq}|\alpha)} D_A^-(u), \qquad \overline{A}(X_i^{\leq}|\beta) = \bigcup_{u \in \overline{A}(X_i^{\leq}|\beta)} D_A^-(u). \tag{13}$$

In the rest of this paper, we discuss only the two class case and a fixed condition attribute subset. Hence, we drop class index i and condition attribute subset A from the previous notation. Let $X^{\geq}$ and $X^{\leq}$ be upward and downward classes. Note that $X^{\geq} \cap X^{\leq} = \emptyset$ and $X^{\geq} \cup X^{\leq} = U$. The lower approximation of $X^{\geq}$ (resp. $X^{\leq}$) is denoted by $\underline{X}^{\geq}(\alpha)$ (resp. $\underline{X}^{\leq}(\alpha)$), and called the positive (resp. negative) region. Because the upper approximation of $X^{\geq}$ (resp. $X^{\leq}$) with β is equal to the complement of the lower approximation of $X^{\leq}$ (resp. $X^{\geq}$), we can only consider the lower approximations. Generalization of the rest discussion to the multiclass case is straightforward.

3 Empirical Risk Minimization

We show a relation between the approximations in VP-DRSA and an empirical risk minimization problem. We associate the lower approximations of upward class $X^{\geq}$ and downward class $X^{\leq}$ with the classification problem where we find an optimal classifier for $X^{\geq}$ and $X^{\leq}$ with respect to a risk function, which is defined as an expected value of a supposed loss function. First, we only consider $X^{\geq}$, and then we consider both, $X^{\geq}$ and $X^{\leq}$.

3.1 Classifiers, Inference Rules, and Loss Functions

Candidates of the optimal classifier are requested to be upward definable sets, because the lower approximation of $X^{\geq}$ is a upward definable set. Therefore, the classifier is restricted by a family $\mathcal{W}_P$ of all upward definable sets, i.e.,

$$\mathcal{W}_P = \left\{ W \subseteq U \;\middle|\; W = \bigcup_{u \in W} D^+(u) \right\}. \tag{14}$$

An inference rule of a classifier $W_P \in \mathcal{W}_P$ is defined by:

$$u \text{ is classified to } \begin{cases} X^{\geq} & \text{if } u \in W_P, \\ X^{\leq} & \text{if } u \notin W_P. \end{cases} \tag{15}$$

A loss function is defined for an object u and a classifier W_P and represents a cost of misclassification of u. The 0-1 loss function is a simple and usually used loss function, which takes 0 if the classifier is correct and 1 otherwise. For $u \in U$ and $W_P \in \mathcal{W}_p$, the 0-1 loss $L_{01}(u, W_P)$ is defined by:

$$L_{01}(u, W_P) = \begin{cases} 0 & (u \in X^{\geq} \text{ and } u \in W_P) \text{ or } (u \notin X^{\geq} \text{ and } u \notin W_P), \\ 1 & \text{otherwise.} \end{cases} \tag{16}$$

However, in this paper, we use another loss function, called a hinge loss function, which is used for support vector machines. To introduce the hinge loss function, we consider a real-valued classifier $f^-_{W_P}$ corresponding to $W_P \in \mathcal{W}_P$:

$$f^-_{W_P}(u) = \begin{cases} |D^-(u) \cap W_P| & u \in W_P, \\ -|D^+(u) \cap \neg W_P| & u \notin W_P. \end{cases} \tag{17}$$

An inference rule of $f^-_{W_P}$ is defined by:

$$u \text{ is classified to } \begin{cases} X^{\geq} & \text{if } f^-_{W_P}(u) > 0, \\ X^{\leq} & \text{if } f^-_{W_P}(u) < 0. \end{cases} \tag{18}$$

When $f^-_{W_P}(u) = 0$, the classification is undecided or arbitrarily decided. Note that for $u \in U$ we have $f^-_{W_P}(u) \geq 1$ or $f^-_{W_P}(u) \leq -1$. Moreover, we introduce a function $y^{\geq}$ to represent set $X^{\geq}$: for $u \in U$,

$$y^{\geq}(u) = \begin{cases} 1 & u \in X^{\geq}, \\ -1 & u \notin X^{\geq}. \end{cases} \tag{19}$$

It is remarkable that when $y^{\geq}(u)$ and $f^-_{W_P}(u)$ have the same sign, u is correctly classified by $f^-_{W_P}(u)$. However, when $y^{\geq}(u)$ and $f^-_{W_P}(u)$ have different signs, u is misclassified by $f^-_{W_P}(u)$. A real-valued classifier f and class indicator $y \in \{-1, 1\}$ for classification problems are usually used in statistical learning methods such as support vector machines or logistic regression.

Given parameters $\lambda^{\leq}_P, \lambda^{\geq}_{P^c} \geq 0$, for u and $W_P \in \mathcal{W}_P$, the hinge loss function is defined as follows:

$$L(y^{\geq}(u), f^-_{W_P}(u) | \lambda^{\leq}_P, \lambda^{\geq}_{P^c}) = \begin{cases} \lambda^{\leq}_P [-y^{\geq}(u) f^-_{W_P}(u)]_+ & y^{\geq}(u) = -1, \\ \lambda^{\geq}_{P^c} [-y^{\geq}(u) f^-_{W_P}(u)]_+ & y^{\geq}(u) = 1, \end{cases} \tag{20}$$

where the notation $[x]_+$ is $\max\{x, 0\}$. If u is correctly classified by $f^-_{W_P}(u)$, then $L(y^{\geq}(u), f^-_{W_P}(u) | \lambda^{\leq}_P, \lambda^{\geq}_{P^c}) = 0$. On the other hand, if u is misclassified by

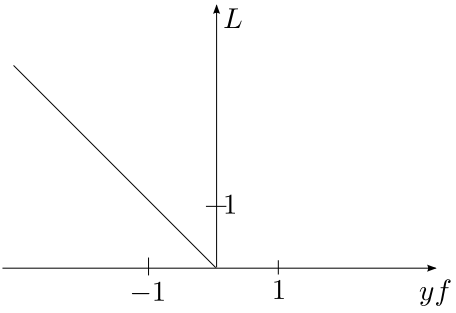


Fig. 1. A hinge loss function

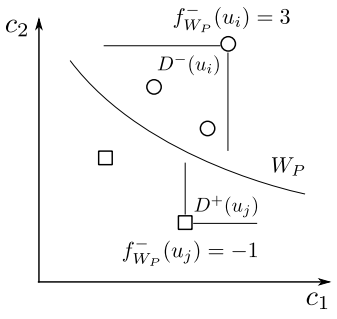


Fig. 2. W_P and $f^-_{W_P}$

$f^-_{W_P}(u)$, it becomes positive. Parameters $\lambda^{\leq}_{P}, \lambda^{\leq}_{\overline{P}^c}$ can be seen as costs for type 1 and type 2 errors, respectively. L is depicted in Figure 1 (when $\lambda_1 = \lambda_2 = 1$).

Value $f^-_{W_P}(u)$ shows the number of objects in W_P supporting $u \in W_P$. Namely, $f^-_{W_P}(u)$ indicates to what extent the membership of u to W_P is guaranteed. Figure 2 shows an illustration of W_P and $f^-_{W_P}$. In this example, there are 5 objects (3 circles and 2 squares), and set W_P is composed of object located above the line from top-left to bottom-right. $f^-_{W_P}$ of the top circle object u_i is 3 because there are 3 objects dominated by u_i and included in W_P. Similarly, $f^-_{W_P}$ of the bottom square object u_j is -1.

3.2 Empirical Risk Minimization for Upward Class $\boldsymbol{X}^{\geq}$

An empirical risk of $W_P \in \mathcal{W}_P$ is defined by the mean of hinge loss over U under given $\lambda^{\leq}_{P}$ and $\lambda^{\geq}_{\overline{P}^c}$, i.e.,

$$R(W_P|\lambda^{\leq}_{P}, \lambda^{\geq}_{\overline{P}^c}) = \frac{1}{n} \sum_{u \in U} L(y^{\geq}(u), f^-_{W_P}(u)|\lambda^{\leq}_{P}, \lambda^{\geq}_{\overline{P}^c}). \tag{21}$$

We can prove the following theorem.

Theorem 1. *Under given $\lambda^{\leq}_{P}, \lambda^{\geq}_{\overline{P}^c} \geq 0$, $W^*_P \in \mathcal{W}_P$ minimizes the empirical risk function $R(\cdot|\lambda^{\leq}_{P}, \lambda^{\geq}_{\overline{P}^c})$ if and only if W^*_P satisfies the following implications, for all $u \in U$*

$$\lambda^{\leq}_{P} \left|D^+(u) \cap X^{\leq}\right| < \lambda^{\geq}_{\overline{P}^c} \left|D^-(u) \cap X^{\geq}\right| \Rightarrow u \in W^*_P, \tag{22}$$

$$\lambda^{\leq}_{P} \left|D^+(u) \cap X^{\leq}\right| > \lambda^{\geq}_{\overline{P}^c} \left|D^-(u) \cap X^{\geq}\right| \Rightarrow u \notin W^*_P. \tag{23}$$

First, we prove the following lemma.

Lemma 1. *We have,*

$$R(W_P|\lambda^{\leq}_{P}, \lambda^{\geq}_{\overline{P}^c}) = \frac{1}{n} \left(\sum_{u' \in W_P} \lambda^{\leq}_{P} \left|D^+(u') \cap X^{\leq}\right| + \sum_{u' \in \neg W_P} \lambda^{\geq}_{\overline{P}^c} \left|D^-(u') \cap X^{\geq}\right| \right). \tag{24}$$

Proof.

$$
\begin{aligned}
&nR(W_P|\lambda^{\leq}_{P}, \lambda^{\geq}_{P^c}) \\
&\quad = \sum_{u \in W_P \cap X^{\leq}} \lambda^{\leq}_{P} \left|D^-(u) \cap W_P\right| + \sum_{u \in \neg W_P \cap X^{\geq}} \lambda^{\geq}_{P^c} \left|D^+(u) \cap \neg W_P\right|, \\
&\quad = \lambda^{\leq}_{P} \left|\left\{(u, u') \in U^2 \;\middle|\; u \in W_P \cap X^{\leq} \text{ and } u' \in D^-(u) \cap W_P\right\}\right| \\
&\qquad + \lambda^{\geq}_{P^c} \left|\left\{(u, u') \in U^2 \;\middle|\; u \in \neg W_P \cap X^{\geq} \text{ and } u' \in D^+(u) \cap \neg W_P\right\}\right|, \\
&\quad = \lambda^{\leq}_{P} \left|\left\{(u, u') \in U^2 \;\middle|\; u \in D^+(u') \cap W_P \cap X^{\leq} \text{ and } u' \in W_P\right\}\right| \\
&\qquad + \lambda^{\geq}_{P^c} \left|\left\{(u, u') \in U^2 \;\middle|\; u \in D^-(u') \cap \neg W_P \cap X^{\geq} \text{ and } u' \in \neg W_P\right\}\right|, \\
&\quad = \sum_{u' \in W_P} \lambda^{\leq}_{P} \left|D^+(u') \cap X^{\leq}\right| + \sum_{u' \in \neg W_P} \lambda^{\geq}_{P^c} \left|D^-(u') \cap X^{\geq}\right|.
\end{aligned}
$$

Proof (Theorem 1). We suppose that for $W_P \in \mathcal{W}_P$ there exists $u \in U$ such that $\lambda^{\leq}_{P} \left|D^+(u) \cap X^{\leq}\right| < \lambda^{\geq}_{P^c} \left|D^-(u) \cap X^{\geq}\right|$ and $u \notin W_P$. Then we show that W_P is not optimal. The statement that the optimality of W^*_P implies (23) is also proved in a similar way. Consider $W'_P = W_P \cup D^+(u)$. We can easily see that $W'_P \in \mathcal{W}_P$ by transitivity of D. The difference of empirical risks of W_P and W'_P is obtained as follows.

$$
\begin{aligned}
&n(R(W_P|\lambda^{\leq}_{P}, \lambda^{\geq}_{P^c}) - R(W'_P|\lambda^{\leq}_{P}, \lambda^{\geq}_{P^c})) \\
&\quad = \sum_{x' \in W_P} \lambda^{\leq}_{P} \left|D^+(x') \cap X^{\leq}\right| + \sum_{x' \in \neg W_P} \lambda^{\geq}_{P^c} \left|D^-(x') \cap X^{\geq}\right| \\
&\qquad - \left(\sum_{x' \in W_P \cup D^+(u)} \lambda^{\leq}_{P} \left|D^+(x') \cap X^{\leq}\right| + \sum_{x' \in \neg(W_P \cup D^+(u))} \lambda^{\geq}_{P^c} \left|D^-(x') \cap X^{\geq}\right| \right), \\
&\quad = - \sum_{x' \in \neg W_P \cap D^+(u)} \lambda^{\leq}_{P} \left|D^+(x') \cap X^{\leq}\right| + \sum_{x' \in \neg W_P \cap D^+(u)} \lambda^{\geq}_{P^c} \left|D^-(x') \cap X^{\geq}\right|, \\
&\quad \geq \sum_{x' \in \neg W_P \cap D^+(u)} \left(\lambda^{\geq}_{P^c} \left|D^-(u) \cap X^{\geq}\right| - \lambda^{\leq}_{P} \left|D^+(u) \cap X^{\leq}\right|\right) > 0.
\end{aligned}
$$

This leads to the conclusion leads that W_P is not optimal. Therefore, we have that the optimality of W^*_P implies (22).

Next, provide an arbitrary W^*_P which satisfies (22) and (23). Consider two sets,

$$
\begin{aligned}
W_1 &= \left\{u \in U \;\middle|\; \lambda^{\leq}_{P} \left|D^+(u) \cap X^{\leq}\right| < \lambda^{\geq}_{P^c} \left|D^-(u) \cap X^{\geq}\right|\right\}, \\
W_2 &= \left\{u \in U \;\middle|\; \lambda^{\leq}_{P} \left|D^+(u) \cap X^{\leq}\right| > \lambda^{\geq}_{P^c} \left|D^-(u) \cap X^{\geq}\right|\right\}.
\end{aligned}
$$

The empirical risk of W_P^* is obtained as follows.

$$\begin{aligned}
&nR(W_P^*|\lambda_P^{\leq}, \lambda_{P^c}^{\geq}) \\
&\quad = \sum_{u' \in W_1} \lambda_P^{\leq} \left| D^+(u') \cap X^{\leq} \right| + \sum_{u' \in W_P^* \cap \neg W_1 \cap \neg W_2} \lambda_P^{\leq} \left| D^+(u') \cap X^{\leq} \right| \\
&\qquad + \sum_{u' \in W_2} \lambda_{P^c}^{\geq} \left| D^-(u') \cap X^{\geq} \right| + \sum_{u' \in \neg W_P^* \cap \neg W_1 \cap \neg W_2} \lambda_{P^c}^{\geq} \left| D^-(u') \cap X^{\geq} \right|, \\
&\quad = \sum_{u' \in W_1} \lambda_P^{\leq} \left| D^+(u') \cap X^{\leq} \right| + \sum_{u' \in W_2} \lambda_{P^c}^{\geq} \left| D^-(u') \cap X^{\geq} \right| \\
&\qquad + \sum_{u' \in \neg W_1 \cap \neg W_2} \lambda_P^{\leq} \left| D^+(u') \cap X^{\leq} \right|.
\end{aligned}$$

The minimum empirical risk value of W_P^* depends only on W_1 and W_2. Additionally, from the first part of this proof, we know that an optimal W_P satisfies (22) and (23). Therefore, any W_P^* satisfying (22) and (23) is optimal.

Theorem 1 says that optimality of set W_P (definable with dominating sets D^+), in the sense of minimizing the empirical risk function, is determined by the implication rules (22) and (23), which is related to the conditions of the approximations in VP-DRSA. This leads to the following corollary.

Corollary 1. *Suppose* $\lambda_P^{\leq} > 0$. $\underline{X}^{\geq}(\frac{\lambda_P^{\leq}}{\lambda_P^{\leq} + \lambda_{P^c}^{\geq}})$ *minimizes* $R(\cdot|\lambda_P^{\leq}, \lambda_{P^c}^{\geq})$. *Moreover, if* $\lambda_P^{\leq} \left| D^+(u) \cap X^{\leq} \right| \neq \lambda_{P^c}^{\geq} \left| D^-(u) \cap X^{\geq} \right|$ *for every* $u \in U$ *then it is the unique optimal solution.*

Thus, the lower approximation of the upward class is characterized by an optimal solution of the empirical risk minimization.

3.3 Empirical Risk Minimization for Upward and Downward Classes, $X^{\geq}$ and $X^{\leq}$

Now, let us extend the above results to the case of classification into upward class $X^{\geq}$ or downward class $X^{\leq}$. Consider the following family of all downward definable sets,

$$\mathcal{W}_N = \left\{ W \subseteq U \;\middle|\; W = \bigcup_{u \in W} D^-(u) \right\}. \tag{25}$$

Let $W_P \in \mathcal{W}_P$ and $W_N \in \mathcal{W}_N$. We introduce additional nonnegative costs of type 1 and 2 errors for classifier W_N: $\lambda_N^{\geq}, \lambda_{N^c}^{\leq} \geq 0$. Then an empirical risk for $\lambda_P^{\leq}, \lambda_{P^c}^{\geq}, \lambda_N^{\geq}, \lambda_{N^c}^{\leq}$ and W_P, W_N is defined as follows:

$$\begin{aligned}
&\hat{R}(W_P, W_N|\lambda_P^{\leq}, \lambda_{P^c}^{\geq}, \lambda_N^{\geq}, \lambda_{N^c}^{\leq}) \\
&\quad = \frac{1}{n} \sum_{u \in U} \left(L(y^{\geq}(u), f_{W_P}^-(u)|\lambda_P^{\leq}, \lambda_{P^c}^{\geq}) + L(y^{\leq}(u), f_{W_N}^+(u)|\lambda_N^{\geq}, \lambda_{N^c}^{\leq}) \right),
\end{aligned} \tag{26}$$

where we define,

$$y^{\leq}(u) = -y^{\geq}(u), \tag{27}$$

$$f^{+}_{W_N}(u) = \begin{cases} |D^{+}(u) \cap W_N| & u \in W_N, \\ -|D^{-}(u) \cap (U \setminus W_N)| & u \notin W_N. \end{cases} \tag{28}$$

Similar to $f^{-}_{W_P}$, $f^{+}_{W_N}$ is a real-valued classifier for $X^{\leq}$, namely, if $f^{+}_{W_N}(u) > 0$ then u is classified to $X^{\leq}$, and if $f^{+}_{W_N}(u) < 0$ then u is classified to $X^{\geq}$.

Theorem 2. *Let $\lambda^{\leq}_{P}, \lambda^{\geq}_{P^c}, \lambda^{\geq}_{N}, \lambda^{\leq}_{N^c} \geq 0$ be given, and let $W^*_P \in \mathcal{W}_P$, $W^*_N \in \mathcal{W}_N$. Then (W^*_P, W^*_N) minimizes the empirical risk function $\hat{R}(\cdot,\cdot|\lambda^{\leq}_{P}, \lambda^{\geq}_{P^c}, \lambda^{\geq}_{N}, \lambda^{\leq}_{N^c})$ if and only if W^*_P and W^*_N satisfy for all $u \in U$ the following implications,*

$$\lambda^{\leq}_{P} \left|D^{+}(u) \cap X^{\leq}\right| < \lambda^{\geq}_{P^c} \left|D^{-}(u) \cap X^{\geq}\right| \Rightarrow u \in W^*_P, \tag{29}$$

$$\lambda^{\leq}_{P} \left|D^{+}(u) \cap X^{\leq}\right| > \lambda^{\geq}_{P^c} \left|D^{-}(u) \cap X^{\geq}\right| \Rightarrow u \notin W^*_P, \tag{30}$$

$$\lambda^{\geq}_{N} \left|D^{-}(u) \cap X^{\geq}\right| < \lambda^{\leq}_{N^c} \left|D^{+}(u) \cap X^{\leq}\right| \Rightarrow u \in W^*_N, \tag{31}$$

$$\lambda^{\geq}_{N} \left|D^{-}(u) \cap X^{\geq}\right| > \lambda^{\leq}_{N^c} \left|D^{+}(u) \cap X^{\leq}\right| \Rightarrow u \notin W^*_N. \tag{32}$$

Moreover, if $\lambda^{\leq}_{N^c}\lambda^{\geq}_{P^c} < \lambda^{\geq}_{N}\lambda^{\leq}_{P}$ holds, then any optimal solution (W^*_P, W^*_N) satisfies $W^*_P \cap W^*_N = \emptyset$.

Corollary 2. *Suppose $\lambda^{\leq}_{N^c}\lambda^{\geq}_{P^c} < \lambda^{\geq}_{N}\lambda^{\leq}_{P}$. The pair $(\underline{X}^{\geq}(\frac{\lambda^{\leq}_{P}}{\lambda^{\leq}_{P}+\lambda^{\geq}_{P^c}}), \underline{X}^{\leq}(\frac{\lambda^{\geq}_{N}}{\lambda^{\geq}_{N}+\lambda^{\leq}_{N^c}}))$ minimizes $\hat{R}(\cdot,\cdot|\lambda^{\leq}_{P}, \lambda^{\geq}_{P^c}, \lambda^{\geq}_{N}, \lambda^{\leq}_{N^c})$. Moreover, if all of objects satisfy,*

$$\lambda^{\leq}_{P} \left|D^{+}(u) \cap X^{\leq}\right| \neq \lambda^{\geq}_{P^c} \left|D^{-}(u) \cap X^{\geq}\right| \text{ and,}$$
$$\lambda^{\geq}_{N} \left|D^{-}(u) \cap X^{\geq}\right| \neq \lambda^{\leq}_{N^c} \left|D^{+}(u) \cap X^{\leq}\right|,$$

then it is the unique optimal pair of classifiers.

It should be noticed that when $\lambda^{\leq}_{N^c}\lambda^{\geq}_{P^c} < \lambda^{\geq}_{N}\lambda^{\leq}_{P}$, we have $\alpha = \frac{\lambda^{\leq}_{P}}{\lambda^{\leq}_{P}+\lambda^{\geq}_{P^c}} > 1 - \frac{\lambda^{\geq}_{N}}{\lambda^{\geq}_{N}+\lambda^{\leq}_{N^c}} = \beta$ and $\underline{X}^{\geq}(\frac{\lambda^{\leq}_{P}}{\lambda^{\leq}_{P}+\lambda^{\geq}_{P^c}}) \cap \underline{X}^{\leq}(\frac{\lambda^{\geq}_{N}}{\lambda^{\geq}_{N}+\lambda^{\leq}_{N^c}}) = \emptyset$.

3.4 Special Case When in the Set of Attributes There Is No Criterion

If all condition attributes are nominal, i.e., $AT_C = \emptyset$ and $AT = AT_N$, $D^{+}(u)$ and $D^{-}(u)$ boil down to an equivalence class $D(u) = D^{+}(u) = D^{-}(u)$. The definable sets $\mathcal{W}_P$, $\mathcal{W}_N$ and the classification functions $f^{-}_{W_P}$ and $f^{+}_{W_N}$ also boil down to the following definable set $\mathcal{W}$ and classification function f_W, respectively:

$$\mathcal{W} = \left\{ W \subseteq U \;\middle|\; W = \bigcup_{u \in W} D(u) \right\}, \tag{33}$$

$$f_W = \begin{cases} |D(u)| & u \in W, \\ -|D(u)| & u \notin W. \end{cases} \tag{34}$$

For convenience, let us rename $X^{\geq}$, $X^{\leq}$, $\lambda_P^{\leq}$, $\lambda_{N^c}^{\leq}$, $\lambda_N^{\geq}$, $\lambda_{P^c}^{\geq}$, $y^{\geq}$, and $y^{\leq}$ to X, X^c, λ_P^c, $\lambda_{N^c}^c$, λ_N, λ_{P^c}, y, and y^c, respectively. For $W_P, W_N \in \mathcal{W}$, the empirical risk function $\hat{R}$ is reformulated as:

$$
\begin{aligned}
&\hat{R}(W_P, W_N | \lambda_P^c, \lambda_{P^c}, \lambda_N, \lambda_{N^c}^c) \\
&= \frac{1}{n} \sum_{u \in U} (L(y(u), f_{W_P}(u) | \lambda_P^c, \lambda_{P^c}) + L(y(u), f_{W_N}(u) | \lambda_N, \lambda_{N^c}^c)), \\
&= \frac{1}{n} \sum_{E \in \{D(u) | u \in U\}} \left(\sum_{u' \in E \cap X^c \cap W_P} \lambda_P^c |E| + \sum_{u' \in E \cap X \cap \neg W_P} \lambda_{P^c} |E| \right. \\
&\qquad \left. + \sum_{u' \in E \cap X \cap W_N} \lambda_N |E| + \sum_{u' \in E \cap X^c \cap \neg W_N} \lambda_{N^c}^c |E| \right), \\
&= \frac{1}{n} \sum_{E \in \{D(u) | u \in U\}} (\lambda_P^c |E \cap X^c \cap W_P| + \lambda_{P^c} |E \cap X \cap \neg W_P| \\
&\qquad + \lambda_N |E \cap X \cap W_N| + \lambda_{N^c}^c |E \cap X^c \cap \neg W_N|) \, |E|.
\end{aligned}
$$

In order to see a relation to the Bayes risk, we introduce $W_P \cap W_N = \emptyset$.

$$
\begin{aligned}
&\hat{R}(W_P, W_N | \lambda_P^c, \lambda_{P^c}, \lambda_N, \lambda_{N^c}^c) \\
&= \frac{1}{n} \sum_{E \in \{D(u) | u \in U\}} \left(\sum_{E \subseteq W_P} (\lambda_P^c + \lambda_{N^c}^c) |E \cap X^c| + \sum_{E \subseteq W_N} (\lambda_N + \lambda_{P^c}) |E \cap X| \right. \\
&\qquad \left. + \sum_{E \subseteq \neg (W_P \cup W_N)} (\lambda_{P^c} |E \cap X| + \lambda_{N^c}^c |E \cap X^c|) \right) |E|, \\
&= \sum_{E \in \{D(u) | u \in U\}} \left(\sum_{E \subseteq W_P} (\lambda_P^c + \lambda_{N^c}^c) \frac{|E \cap X^c|}{|E|} + \sum_{E \subseteq W_N} (\lambda_N + \lambda_{P^c}) \frac{|E \cap X|}{|E|} \right. \\
&\qquad \left. + \sum_{E \subseteq \neg (W_P \cup W_N)} (\lambda_{P^c} \frac{|E \cap X|}{|E|} + \lambda_{N^c}^c \frac{|E \cap X^c|}{|E|}) \right) \left(\frac{|E|}{n} \right)^2 n.
\end{aligned}
$$

Representing $\frac{|E \cap X^c|}{|E|}$, $\frac{|E \cap X|}{|E|}$, and $\frac{|E|}{n}$ by $P(X^c|E)$, $P(X|E)$, and $P(E)$, respectively, we obtain

$$
\begin{aligned}
&\hat{R}(W_P, W_N | \lambda_P^c, \lambda_{P^c}, \lambda_N, \lambda_{N^c}^c)/n \\
&= \sum_{E \in \{D(u) | u \in U\}} \left(\sum_{E \subseteq W_P} (\lambda_P^c + \lambda_{N^c}^c) P(X^c|E) + \sum_{E \subseteq W_N} (\lambda_N + \lambda_{P^c}) P(X|E) \right. \\
&\qquad \left. + \sum_{E \subseteq \neg (W_P \cup W_N)} (\lambda_{P^c} P(X|E) + \lambda_{N^c}^c P(X^c|E)) \right) P(E)^2.
\end{aligned}
$$

Hence, $\hat{R}/n$ becomes similar to the Bayes risk which is defined by the last formula with replacement of multiplier $P(E)^2$ with $P(E)$. Because for each $E \in \{D(u)|u \in U\}$, we can decide which region among W_P, W_N and $\neg(W_P \cup W_N)$ includes E, individually, the implications characterizing optimal W_P^* and W_N^* becomes same as those obtained in DTRSM [8] when parameters λ_P and λ_N^c of DTRSM are zeros.

4 Concluding Remarks

In this paper, we have shown the connection between VP-DRSA and the empirical risk minimization. We have demonstrated that the empirical risk function with the proposed hinge loss function serves as a foundation for VP-DRSA.

Acknowledgment. The second and the fourth authors acknowledge financial support from the Polish National Science Center, grant no. 155534.

References

1. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: Monotonic Variable Consistency Rough Set Approaches. Internal Journal of Approximate Reasoning 50, 979–999 (2009)
2. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129, 1–47 (2001)
3. Greco, S., Słowiński, R., Yao, Y.: Bayesian decision theory for dominance-based rough set approach. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 134–141. Springer, Heidelberg (2007)
4. Inuiguchi, M., Yoshioka, Y., Kusunoki, Y.: Variable-precision dominance-based rough set approach and attribute reduction. International Journal of Approximation Reasoning 50, 1199–1214 (2009)
5. Pawlak, Z.: Rough sets. International Journal of Information and Computer Sciences 11, 341–356 (1982)
6. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177, 3–27 (2007)
7. Ślęzak, D., Ziarko, W.: The investigation of the Bayesian rough set model. International Journal of Approximate Reasoning 40, 81–91 (2005)
8. Yao, Y.: Probabilistic rough set approximations. International Journal of Approximation Reasoning 49, 255–271 (2008)
9. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46(1), 39–59 (1993)

Formulating Game Strategies in Game-Theoretic Rough Sets

Nouman Azam and JingTao Yao

Department of Computer Science, University of Regina,
Regina, Saskatchewan, Canada S4S 0A2
{azam200n,jtyao}@cs.uregina.ca

Abstract. The determination of thresholds (α, β) has been considered as a fundamental issue in probabilistic rough sets. The game-theoretic rough set (GTRS) model determines the required thresholds based on a formulated game between different properties related to rough sets approximations and classification. The game strategies in the GTRS model are generally based on an initial threshold configuration that corresponds to the Pawlak model. We study different approaches for formulating strategies by considering different initial conditions. An example game is shown for each case. The selection of a particular approach for a given problem may be based on the quality of data and computing resources at hand. The realization of these approaches in GTRS based methods may bring new insights into effective determination of probabilistic thresholds.

1 Introduction

The probabilistic rough set model has been recognized as a major extension, improvement and generalization of the Pawlak rough set model [10]. The model utilizes a pair of probabilistic (α, β) thresholds to determine the division between probabilistic positive, negative and boundary regions [10]. A fundamental issue in probabilistic rough sets is the computation or determination of the (α, β) threshold parameters [11]. Several attempts have been made recently in this regard including decision-theoretic, game-theoretic, information-theoretic, optimization based and risk based approaches [1, 3, 4, 5, 6, 7]. Despite these attempts, it might still be premature at this point of time to come up with a solution that is universally accepted and convince the majority (if not all) of the audience for its superiority. For now, the need for further research remains in order to obtain more interesting results.

The game-theoretic rough set (GTRS) model has recently provided an alternative way for determining the probabilistic thresholds [4]. It utilizes a game-theoretic environment in determining these thresholds by analyzing and directing towards the optimization of one or more characteristics of the rough set model. Particularly, the thresholds are computed based on a game between different properties related to rough sets based approximation, classification or decision making in order to reach a suitable tradeoff.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 145–153, 2013.

The strategies in GTRS are generally formulated based on an initial threshold configuration $(\alpha, \beta) = (1, 0)$ which corresponds to the Pawlak model [1, 2]. This only allows for the formulation of strategies in terms of decreasing levels for threshold α and increasing levels for threshold β (considering $0 \leq \beta < \alpha \leq 1$ in the probabilistic rough set model). It seems that the motivation or rationale behind this approach is to obtain a model that is at least better than the Pawlak model based on some considered performance criteria. The approach is useful for configuring the thresholds, it may not necessarily provide an overall better model. We propose various approaches for formulating strategies by considering different initial conditions. A game is implemented for each approach and the threshold modification trend based on a repetitive game is examined. It is hoped that these approaches may further improve and enhance the process of threshold modification and the quality of obtained thresholds.

2 Problem Statement

A main result of probabilistic rough sets is that the rules for determining the three regions are given by,

$$\begin{array}{ll} \text{Positive:} & \text{if } P(C|[x]) \geq \alpha, \\ \text{Negative:} & \text{if } P(C|[x]) \leq \beta, \text{ and} \\ \text{Boundary:} & \text{if } \beta < P(C|[x]) < \alpha. \end{array} \tag{1}$$

where $P(C[x])$ denotes the conditional probability of an object x to be in C given that the object is in $[x]$ and $0 \leq \beta < \alpha \leq 1$. The division between the three regions is based on the probabilistic thresholds (α, β) [9]. The determination and interpretation of thresholds are among the fundamental issues in probabilistic rough sets [11]. There are at least three approaches to determine the thresholds based on decision theory, game theory and information theory that lead us to decision-theoretic rough set (DTRS) [9], game-theoretic rough set (GTRS) [2, 4] and information theoretic rough set (ITRS) [3] models, respectively.

The GTRS model determines the threshold parameters based on a formulated game. A typical game consists of a tuple $\{P, S, u\}$, where:

- P is a finite set of n players, indexed by i,
- $S = S_1 \times ... \times S_n$, where S_i is a finite set of strategies available to player i.
- $u = (u_1, ..., u_n)$ where $u_i : S_i \longmapsto \Re$ is a real-valued utility or payoff function for player i.

The GTRS model considers the players in the form of multiple criteria. Each criterion represents a particular aspect of interest like accuracy or applicability of decision rules. Suitable measures are selected to evaluate these criteria in the context of rough sets based approximation and classification. Each criterion is affected by considering different (α, β) threshold configurations. The strategies are therefore formulated in terms of changes in probabilistic thresholds [1].

The payoff functions represent possible gains, benefits or performance levels achieved by considering different modification in threshold levels.

It is not generally suitable to look into the entire range of threshold values within a single GTRS based game. A repetitive or iterative game is generally used where at each iteration the game outcome is used in directing towards optimal threshold values. In existing GTRS based approaches, the initial (α, β) pair is considered as (1,0) that corresponds to the Pawlak model. We suggest and investigate additional approaches for formulating strategies by considering different initial conditions for determining effective threshold values.

3 Approaches for Formulating Strategies in GTRS

This section introduces four approaches for formulating strategies with GTRS. The game structure and threshold modification trend is discussed for each case.

3.1 The Two Ends Approach

The generally used approach for formulating strategies in GTRS is to consider suitable decreasing levels for threshold α and increasing levels for threshold β. Examples of this approach can be found in [1, 2, 4]. The strategies formulated in this way commonly consider an initial configuration of thresholds values, i.e. $(\alpha, \beta) = (1, 0)$ that corresponds to the Pawlak model. We call this approach as the two ends approach since the threshold values are being modified from the two extreme ends.

Table 1. Game for two ends approach

		P_2		
		$s_1 = \alpha_\downarrow$	$s_2 = \beta_\uparrow$	$s_3 = \alpha_\downarrow\beta_\uparrow$
P_1	$s_1 = \alpha_\downarrow$	....	.....	.....
	$s_2 = \beta_\uparrow$	....	.....	.....
	$s_3 = \alpha_\downarrow\beta_\uparrow$	....	.....	.....

An example game based on this approach is presented in the form of Table 1. Each player in this game considers three strategies, namely $s_1 = \alpha_\downarrow$ (decrease α), $s_2 = \beta_\uparrow$ (increase β), and $s_3 = \alpha_\downarrow\beta_\uparrow$ (decrease α and increase β). The increases or decreases may be set by the user or may be defined in terms of the utilities attained by the players. The outcome of this game may be used to repeat the game based on new values of the thresholds. As the game repeats, the threshold α is continuously decreased while threshold β is increased. The amount of an increase or decrease depends on the outcome of the implemented game.

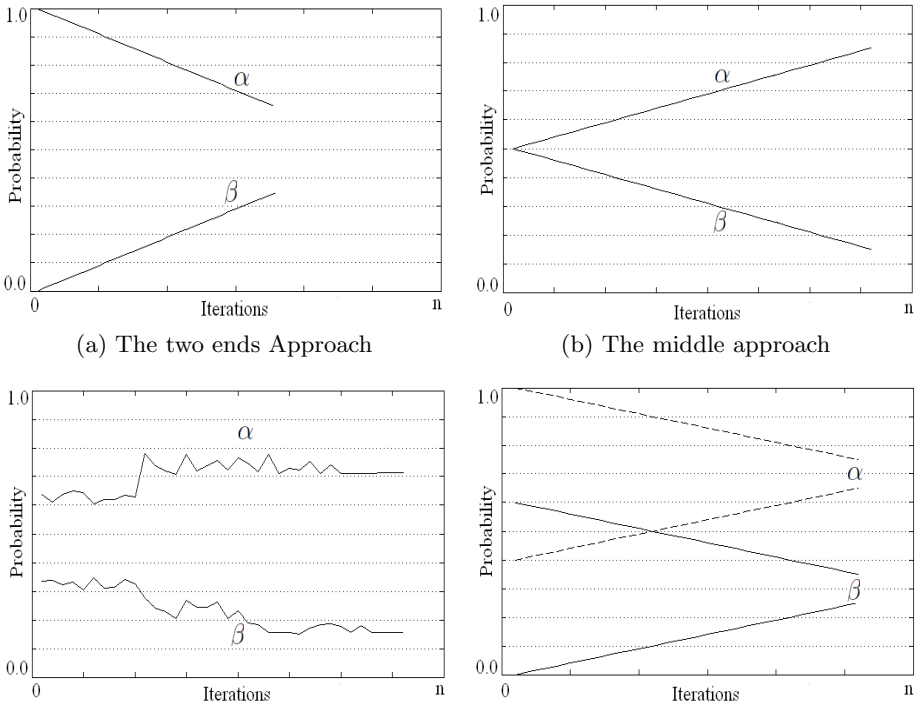


Fig. 1. The four approaches for threshold determination

Figure 1(a) shows the general threshold modification trend with this approach. The modifications in the threshold values are not necessarily linear with respect to iterations. The stop criteria with this approach should be defined to ensure that the process stops before the threshold α becomes less than or equal to β. This approach may be useful when the data are of high quality and the classes or concepts are well defined. A minimum size for the boundary region may be expected in this case. One can make certain decisions with high accuracy rate while keeping the value of α close to 1.0 and β close to 0.0. This means that an effective model may be obtained by considering some minor adjustments to threshold values $(\alpha, \beta) = (1, 0)$.

3.2 The Middle Approach

An alternative approach for formulating strategies is to consider the threshold modification from an initial threshold setting given by $\alpha = \beta$ that corresponds to the two-way decision model. Considering the constraint $\beta < \alpha$, a formulated game based on this approach should consider strategies for increasing α and decreasing β. In some sense this approach can provide an opposite mechanism for

Table 2. Game for middle start approach

		P_2		
		$s_1 = \alpha_\uparrow$	$s_2 = \beta_\downarrow$	$s_3 = \alpha_\uparrow\beta_\downarrow$
P_1	$s_1 = \alpha_\uparrow$	....	.....	....
	$s_2 = \beta_\downarrow$	....	.....	....
	$s_3 = \alpha_\uparrow\beta_\downarrow$	....	.....	....

Table 3. Game for random start

		P_2	
		$s_1 = \alpha_\uparrow$	$s_2 = \alpha_\downarrow$
P_1	$s_1 = \beta_\uparrow$	....	.....
	$s_2 = \beta_\downarrow$	....	.....

threshold configuration as compared to the two ends approach (where threshold α keeps decreasing while β keeps increasing). As the thresholds are being modified from a common or middle value, we name this approach as middle approach.

An example game for this approach may be implemented as shown in Table 2. The strategies may be interpreted as $s_1 = \alpha_\uparrow$ (increase α), $s_2 = \beta_\downarrow$ (decrease β), and $s_3 = \alpha_\uparrow\beta_\downarrow$ (increase α and decrease β). When this game is played repeatedly, the threshold α is expected to increase and β is expected to decrease. Figure 1(b) shows the expected development in the two threshold values based on the repeated game. The stop conditions in this approach should be carefully designed such that the iterative process stops before the Pawlak model is reached. This approach may be useful to compare the probabilistic two way decision model and the probabilistic three-way decision model. Particularly, it can provide further insights into the performance related issues associated with the two models.

This approach may be used when the data are of low quality and involve a high level of uncertainty. In such cases we expect many objects in the boundary region leading to its larger size. The number of available certain decisions are very limited. The objective in such situations is to reduce the boundary size to allow for some certain decisions at a cost of some decrease in the level of accuracy. The middle start which starts from zero sized boundary can provide useful configuration of thresholds under these conditions.

3.3 The Random Approach

We may consider a random point for starting the threshold configuration with GTRS. It is assumed that we do not have any knowledge about the modification direction that will provide effective threshold values. In other words, we are not sure wether to increase or decrease a particular threshold. The formulated strategies should therefore provide options for both increasing or decreasing a particular threshold. This means that the strategies will allow us to investigate effective threshold values in the neighborhood of the starting random point.

Table 3 presents an example game for this approach. Here the strategies for the two players are different. Players 1 has the strategies $s_1 = \beta_\uparrow$ (increase β)

and $s_2 = \beta_{\downarrow}$ (decrease β) and player 2 has the strategies $s_1 = \alpha_{\uparrow}$ (increase α) and $s_2 = \alpha_{\downarrow}$ (decrease α). Such a game may be realized when player 1 is considering some property of the negative region while player 2 is reflecting the same or some other property of the positive region. Figure 1(c) presents the general threshold modification trend. An implementation of this approach should provide a configuration that is at least better than the initial random point. However, an overall optimal configuration may be not be necessarily achieved. Finally, this approach may be suited to applications that are associated with an intermediate level of uncertainty where the effective threshold values can be located anywhere in the threshold space.

3.4 The Range Approach

The strategies may also be formulated by considering a possible range of values for the thresholds. It may not be feasible to evaluate and consider the entire set of values contained in the range within a single game, however, some selected values from the range may be represented as possible strategies. The game may start from a wider range which is iteratively reduced to a finer range based on a game outcome in a repeated game.

The game in Table 4 may be used to implement this approach. Considering an initial range for threshold α as $[0.5, 1.0]$, the strategies $s_1 = \alpha_1, s_2 = \alpha_2, ..., s_n = \alpha_n$ are representing different values in the considered range. Realizing an order among the strategies such as $\alpha_1 < \alpha_2 ... < \alpha_n$. The strategy α_1 may represent the lower value in the range, i.e. 0.5 and the α_n may represent the upper value in the range, i.e. 1.0. The other strategies may represent intermediate values taken at some specified intervals within the range. Similar interpretation may apply to strategies $s_1 = \beta_1, s_2 = \beta_2, ..., s_n = \beta_n$. The range may be reduced repeatedly by some specified factor, e.g. the range [0.5,1.0] for α may be reduced by a factor of 2 as $(1.0 - 0.5)/2 = 0.25$. The new range may be centered around the threshold values determined by the game outcome. Figure 1(d) presents the general trend in modifying thresholds with this approach. The approach may be useful when we are faced with tight computing constraint and quick convergence or determination of thresholds is desired.

Table 4. Game for range based approach

		P_2		
		$s_1 = \alpha_1$	.....	$s_n = \alpha_n$
P_1	$s_1 = \beta_1$	....	.....	....
	...	....	.....	....
	$s_n = \beta_n$	....	.....	....

4 Threshold Configuration with the Two Ends Approach

We provide an example for the two ends approach which can be used to construct examples for the other approaches. The example is similar to those discussed in [1, 2, 3]. Table 5 represents probabilistic information about a category or concept C based on a partition consisting of 18 equivalence classes. An equivalence class is represented as X_i, and its conditional probability with C as $P(C|X_i)$.

Table 5. Probabilistic information of a concept C

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9
$Pr(X_i)$	0.034	0.099	0.132	0.017	0.068	0.017	0.056	0.049	0.049
$Pr(C\|X_i)$	1.0	0.96	0.91	0.86	0.81	0.77	0.71	0.64	0.53
	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}
$Pr(X_i)$	0.115	0.072	0.01	0.119	0.019	0.042	0.009	0.047	0.046
$Pr(C\|X_i)$	0.49	0.43	0.38	0.31	0.27	0.22	0.15	0.09	0.02

Let us consider the game shown in Table 1 for implementing the two ends start approach. Considering the players in the game as the properties of accuracy and generality of the rough set model. For a group containing both positive and negative regions we may define these measures as [2],

$$Accuracy(\alpha,\beta) = \frac{\text{Correctly classified objects by POS}_{(\alpha,\beta)} \text{ and NEG}_{(\alpha,\beta)}}{\text{Total classified objects by POS}_{(\alpha,\beta)} \text{ and NEG}_{(\alpha,\beta)}}, \tag{2}$$

$$Generality(\alpha,\beta) = \frac{\text{Total classified objects by POS}_{(\alpha,\beta)} \text{ and NEG}_{(\alpha,\beta)}}{\text{Number of objects in } U}. \tag{3}$$

where $\text{POS}_{(\alpha,\beta)}$ and $\text{NEG}_{(\alpha,\beta)}$ are the probabilistic positive and negative regions. For each X_i, $X_i \subseteq \text{POS}_{(\alpha,\beta)}$ if $P(C|X_i) \geq \alpha$ and $X_i \subseteq \text{NEG}_{(\alpha,\beta)}$ if $P(C|X_i) \leq \beta$. This means that for $(\alpha,\beta) = (0.9, 0.1)$, we have, $\text{POS}_{(0.9,0.1)} = \bigcup\{X_1, X_2, X_3\}$ and $\text{NEG}_{(0.9,0.1)} = \bigcup\{X_{17}, X_{18}\}$.

Considering U as the total number of objects, number of objects classified by positive and negative regions can be calculated as [3],

$$\begin{aligned} \text{Classified objects by POS}_{(\alpha,\beta)} &= \sum_{P(C|X_i)\geq\alpha} P(X_i) \times U, \text{ and} \\ \text{Classified objects by NEG}_{(\alpha,\beta)} &= \sum_{P(C|X_i)\leq\beta} P(X_i) \times U. \end{aligned} \tag{4}$$

Moreover, the number of correctly classified objects can be determined as [3],

$$\begin{aligned} \text{Correctly classified by POS}_{(\alpha,\beta)} &= \sum_{P(C|X_i)\geq\alpha} P(C|X_i) \times P(X_i) \times U, \text{ and} \\ \text{Correctly classified by NEG}_{(\alpha,\beta)} &= \sum_{P(C|X_i)\leq\beta} (1 - P(C|X_i)) \times P(X_i) \times U. \end{aligned} \tag{5}$$

Table 6. The example game for the two ends approach

		Generality		
		$s_1 = \alpha_\downarrow$	$s_2 = \beta_\uparrow$	$s_3 = \alpha_\downarrow\beta_\uparrow$
Accuracy	$s_1 = \alpha_\downarrow$	(0.941,0.265)	(0.937,0.179)	(0.946,0.311)
	$s_2 = \beta_\uparrow$	(0.973,0.179)	(0.959,0.127)	**(0.959,0.226)**
	$s_3 = \alpha_\downarrow\beta_\uparrow$	(0.946,0.311)	(0.959,0.226)	(0.941,0.358)

For a threshold pair $(\alpha, \beta) = (0.9, 0.1)$, we can calculate the total number of classified objects by $\text{POS}_{(0.9,0.1)}$ as $(P(X_1) + P(X_2) + P(X_3)) \times U = 0.265 \times U$ and the number of classified objects by $\text{NEG}_{(0.9,0.1)} = (P(X_{17}) + P(X_{18})) \times U = 0.093 \times U$. Similarly, the number of correctly classified objects by $\text{POS}_{(0.9,0.1)} = (P(C|X_1) * P(X_1) + P(C|X_2) * P(X_2) + P(C|X_3) * (P(X_3)) \times U = 0.2492 \times U$ and the number of correctly classified objects by $\text{NEG}_{(0.9,0.1)} = ((1 - P(C|X_{17})) * P(X_{17}) + (1 - P(C|X_{18})) * P(X_{18})) \times U = 0.0879 \times U$. Putting these values in Equations (2) - (3), we have

$$\begin{aligned} \textit{Accuracy}(0.9, 0.1) &= \frac{(0.2492 + 0.0879) \times U}{(0.265 + 0.093) \times U} = \frac{0.3371}{0.358} = 0.941, \\ \textit{Generality}(0.9, 0.1) &= \frac{(0.265 + 0.093) \times U}{U} = 0.358. \end{aligned} \tag{6}$$

Focusing the game in Table 1, each player is allowed to choose from one of the following strategies namely $s_1 = \alpha_\downarrow$ (decrease α), $s_2 = \beta_\uparrow$ (increase β), and $s_3 = \alpha_\downarrow\beta_\uparrow$ (decrease α and increase β). Let us consider a decrease or increase of 5%. Each cell in the Table 1 corresponds to a strategy profile. A threshold pair corresponding to a strategy profile is calculated based on two rules, 1) If only one player plays a strategy of modifying a particular threshold, the value will be determined as an increase or decrease suggested by that player, 2) If both the players play the strategies of modifying a particular threshold, the value will be decided as the sum of the two changes.

Considering an initial threshold configuration of $(\alpha, \beta) = (1, 0)$, we may calculate the threshold pairs corresponding to different strategy profiles. For instance the profile $(s_1, s_1) = (\alpha_\downarrow, \alpha_\downarrow) = (0.9, 0.0)$. The corresponding values for the measures accuracy and generality can be calculated as mentioned above. Table 6 shows the resulting game. The pair of values inside a particular cell represents the utilities of the players. The cell with bold values represent the solution of the game determined by the Nash equilibrium [8]. The corresponding threshold values are given by $(\alpha, \beta) = (\beta_\uparrow, \alpha_\downarrow\beta_\uparrow) = (0.95, 0.1)$. The determined values may be used again to implement a game for the next round. Implementing an iterative game in this fashion will result in the modification sequence of $1.0 \rightarrow 0.95 \rightarrow 0.90 \rightarrow 0.85$ for threshold α and $0.0 \rightarrow 0.1 \rightarrow 0.15 \rightarrow 0.25$ for β. It is noted that these threshold modification trends are similar to those shown in Figure 1(a).

5 Conclusion

The game-theoretic rough set model has recently received some attention for determining effective probabilistic thresholds defining the three probabilistic rough set regions. The GTRS implements a game where the strategies are realized as different levels for modifying the thresholds. In this article, we examine additional approaches for formulating strategies based on different initial conditions. The implementation of these approaches is realized by considering example games corresponding to each approach. The iterative threshold modification with these approaches based on a repetitive game is also discussed. It is argued that some of these approaches may be more appropriate when different types of data and applications are considered. A demonstrative example is included to show the usability of the suggested approaches.

Acknowledgements. This work was partially supported by a Discovery Grant from NSERC Canada and the University of Regina FGSR Dean's Scholarship program.

References

[1] Azam, N., Yao, J.T.: Multiple criteria decision analysis with game-theoretic rough sets. In: Li, T., Nguyen, H.S., Wang, G., Grzymala-Busse, J.W., Janicki, R., Hassanien, A.E., Yu, H. (eds.) RSKT 2012. LNCS, vol. 7414, pp. 399–408. Springer, Heidelberg (2012)

[2] Azam, N., Yao, J.T.: Analyzing uncertainties of probabilistic rough set regions with game-theoretic rough sets. International Journal of Approximate Reasoning (2013), http://dx.doi.org/10.1016/j.ijar.2013.03.015

[3] Deng, X., Yao, Y.Y.: An information-theoretic interpretation of thresholds in probabilistic rough sets. In: Li, T., Nguyen, H.S., Wang, G., Grzymala-Busse, J.W., Janicki, R., Hassanien, A.E., Yu, H. (eds.) RSKT 2012. LNCS, vol. 7414, pp. 369–378. Springer, Heidelberg (2012)

[4] Herbert, J.P., Yao, J.T.: Game-theoretic rough sets. Fundamenta Informaticae 108(3-4), 267–286 (2011)

[5] Jia, X.Y., Tang, Z.M., Liao, W.L., Shang, L.: On an optimization representation of decision-theoretic rough set model. International Journal of Approximate Reasoning (2013), http://dx.doi.org/10.1016/j.ijar.2013.02.010

[6] Li, H.X., Zhou, X.Z.: Risk decision making based on decision-theoretic rough set: A three-way view decision model. International Journal of Computational Intelligence Systems 4(1), 1–11 (2011)

[7] Liu, D., Li, T., Ruan, D.: Probabilistic model criteria with decision-theoretic rough sets. Information Science 181(17), 3709–3722 (2011)

[8] von Neumann, J., Morgenstern, O., Kuhn, H., Rubinstein, A.: Theory of Games and Economic Behavior (Commemorative Edition). Princeton University Press (2007)

[9] Yao, Y.Y.: Decision-theoretic rough set models. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 1–12. Springer, Heidelberg (2007)

[10] Yao, Y.Y.: Probabilistic rough set approximations. International Journal of Approximate Reasoning 49(2), 255–271 (2008)

[11] Yao, Y.Y.: Two semantic issues in a probabilistic rough set model. Fundamenta Informaticae 108(3-4), 249–265 (2011)

Sequential Optimization of Approximate Inhibitory Rules Relative to the Length, Coverage and Number of Misclassifications

Fawaz Alsolami[1,2], Igor Chikalov[1], and Mikhail Moshkov[1]

[1] Computer, Electrical and Mathematical Sciences and Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
[2] Computer Science Department, King Abdulaziz University, Saudi Arabia

Abstract. This paper is devoted to the study of algorithms for sequential optimization of approximate inhibitory rules relative to the length, coverage and number of misclassifications. Theses algorithms are based on extensions of dynamic programming approach. The results of experiments for decision tables from UCI Machine Learning Repository are discussed.

Keywords: inhibitory rules, length, coverage, number of misclassifications, dynamic programming.

1 Introduction

In this paper, we present algorithms for optimization of approximate inhibitory rules based on a dynamic programming approach. Inhibitory rules have in the consequent part a relation "attribute $\neq$ value" whereas decision (deterministic) rules have "attribute = value". In [1, 2] it was shown that, for some information systems, decision rules cannot describe the whole information contained in the system. However, inhibitory rules describe the whole information for every information system [3]. Moreover, classifiers based on inhibitory rules have often better accuracy than classifiers based on decision rules [4–6].

In [3] greedy algorithms for inhibitory rules construction were studied. In [7, 8] we presented a dynamic programming approach for construction and optimization of exact inhibitory rules relative to the length and coverage. We considered also sequential optimization of exact inhibitory rules relative to the length and coverage, and presented some comparison of the length and coverage of inhibitory rules constructed by the greedy algorithm and dynamic programming. Similar approaches were used in [9] for sequential optimization of decision (deterministic) rules.

In the present paper, we study algorithms for sequential optimization of inhibitory rules relative to the length, coverage and number of misclassifications. We compare rules constructed by these algorithms with the rules constructed by a greedy algorithm for decision tables from UCI Machine Learning Repository [10].

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 154–165, 2013.

This paper consists of eight sections. Section 2 contains definitions of main notions. In Section 3, we study a directed acyclic graph which allows to describe the whole set of nonredundant γ-inhibitory rules. The procedures of optimization of nonredundant γ-inhibitory rules relative to the length, coverage and number of misclassifications are presented in Section 4 and sequential optimizations in Section 5. Section 7 contains results of experiments with decision tables from UCI Machine Learning Repository and finally Section 8 contains conclusions.

2 Main Notions

A *decision table* T is a rectangular table with n columns labeled with conditional attributes $f_1, \ldots, f_n$. Rows of this table are filled with nonnegative integers which are interpreted as values of conditional attributes. Rows of T are pairwise different and each row is labeled with a nonnegative integer (decision) which is interpreted as a value of the decision attribute d. We denote by $D(T)$ the set of distinct decisions for the table T. We denote by $N(T)$ the number of rows in the table T.

The least common decision for T is a decision from the set $D(T)$ attached to the minimum number of rows in T. If we have a number of such decisions then we choose the minimum one. By $N_{lcd}(T)$ we denote the number of rows in the table T labeled with the least common decision for T.

Let T be nonempty, $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$ and $v_1, \ldots, v_m$ be nonnegative integers. By $T(f_{i_1}, v_1) \ldots (f_{i_m}, v_m)$ we denote a subtable of the table T which contains only rows that have values $v_1, \ldots, v_m$ at the intersection with columns $f_{i_1}, \ldots, f_{i_m}$. Such nonempty subtables (including the table T) are called *separable subtables* of T.

We denote by $E(T)$ the set of attributes from $\{f_1, \ldots, f_n\}$ which are not constant on T. For any $f_i \in E(T)$, we denote by $E(T, f_i)$ the set of values of the attribute f_i in T.

The expression

$$f_{i_1} = v_1 \wedge \ldots \wedge f_{i_m} = v_m \rightarrow d \neq k \tag{1}$$

is called an *inhibitory rule over* T if $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, $v_1, \ldots v_m$ are nonnegative integers, and $k \in D(T)$. It is not impossible that $m = 0$. In this case (1) is equal to the rule

$$\rightarrow d \neq k. \tag{2}$$

Let Θ be a subtable of T and $r = (b_1, \ldots, b_n)$ be a row of Θ. We say that the rule (1) is *realizable for* r, if $v_1 = b_{i_1}, \ldots, v_m = b_{i_m}$. The rule (2) is realizable for any row from Θ.

Let γ be a nonnegative real number. We say that the rule (1) is γ-*true for* Θ if k is the least common decision for $\Theta' = \Theta(f_{i_1}, v_1) \ldots (f_{i_m}, v_m)$ and $N_{lcd}(\Theta') \leq \gamma$. If $m = 0$ then the rule (2) is γ-true for Θ if k is the least common decision for Θ and $N_{lcd}(\Theta) \leq \gamma$.

If the rule (1) is an inhibitory rule over T which is γ-true for Θ and realizable for r, we say that (1) is a *γ-inhibitory rule for Θ and r over T*.

We say that the rule (1) with $m > 0$ is a *nonredundant* γ-inhibitory rule for Θ and r over T if (1) is a γ-inhibitory rule for Θ and r over T and the following conditions hold:

(i) $f_{i_1} \in E(\Theta)$, and if $m > 1$ then $f_{i_j} \in E(\Theta(f_{i_1}, v_1) \dots (f_{i_{j-1}}, v_{j-1}))$ for $j = 2, \dots, m$;
(ii) $N_{lcd}(\Theta) > \gamma$, and if $m > 1$ then $N_{lcd}(\Theta(f_{i_1}, v_1) \dots (f_{i_j}, v_j)) > \gamma$ for $j = 1, \dots, m-1$.

If $m = 0$ then the rule (2) is a *nonredundant* γ-inhibitory rule for Θ and r over T if (2) is a γ-inhibitory rule for Θ and r over T, i.e., if k is the least common decision for Θ and $N_{lcd}(\Theta) \leq \gamma$.

Let Θ be a subtable of T, τ be a rule over T and τ be equal to (1). The *number of misclassifications* of τ relative to Θ is the number of rows in Θ for which τ is realizable and which are labeled with the decision k. We denote it by $\mu(\tau, \Theta)$. The number of misclassifications of the rule (2) relative to Θ is equal to the number of rows in Θ which are labeled with the decision k.

The number m of conditions on the left-hand side of τ is called the *length* of this rule and is denoted by $l(\tau)$. The length of inhibitory rule (2) is equal to 0.

The *coverage* of τ relative to Θ is the number of rows in Θ for which τ is realizable and which are labeled with decisions other than k. We denote it by $c(\tau, \Theta)$. The coverage of inhibitory rule (2) relative to Θ is equal to the number of rows in Θ which are labeled with decisions other than k.

3 Directed Acyclic Graph $\Lambda_\gamma(T)$

We consider an algorithm that constructs a directed acyclic graph $\Lambda_\gamma(T)$ which will be used to describe the set of nonredundant γ-inhibitory rules for T and for each row r of T over T. Nodes of the graph are separable subtables of the table T. During each step, the algorithm processes one node and marks it with the symbol *. At the first step, the algorithm constructs a graph containing a single node T which is not marked with the symbol *.

Let us assume that the algorithm has already performed p steps. We describe now the step $(p+1)$. If all nodes are marked with the symbol * as processed, the algorithm finishes its work and presents the resulting graph as $\Lambda_\gamma(T)$. Otherwise, choose a node (table) Θ, which has not been processed yet.

Let k be the least common decision for Θ. If $N_{lcd}(\Theta) \leq \gamma$ label the considered node with the decision k, mark it with the symbol * and proceed to the step $(p+2)$. If $N_{lcd}(\Theta) > \gamma$, for each $f_i \in E(\Theta)$, draw a bundle of edges from the node Θ. Let $E(\Theta, f_i) = \{b_1, \dots, b_t\}$. Then draw t edges from Θ and label these edges with pairs $(f_i, b_1), \dots, (f_i, b_t)$ respectively. These edges enter to nodes $\Theta(f_i, b_1), \dots, \Theta(f_i, b_t)$. If some of nodes $\Theta(f_i, b_1), \dots, \Theta(f_i, b_t)$ are absent in the graph then add these nodes to the graph. We label each row r of Θ with the set of attributes $E_{\Lambda_\gamma(T)}(\Theta, r) = E(\Theta)$. Mark the node Θ with the symbol * and proceed to the step $(p+2)$.

The graph $\Lambda_\gamma(T)$ is a directed acyclic graph. A node of this graph will be called *terminal* if there are no edges leaving this node. Note that a node Θ of $\Lambda_\gamma(T)$ is terminal if and only if $N_{lcd}(\Theta) \leq \gamma$.

Later, we describe the procedures of optimization of the graph $\Lambda_\gamma(T)$. As a result we obtain a graph G with the same sets of nodes and edges as in $\Lambda_\gamma(T)$. The only difference is that any row r of each nonterminal node Θ of G is labeled with a nonempty set of attributes $E_G(\Theta, r) \subseteq E(\Theta)$.

For each node Θ of G and for each row r of Θ, we describe a set of γ-inhibitory rules $Rul_G(\Theta, r)$ over T. We move from terminal nodes of G to the node T.

Let Θ be a terminal node of G and k be the least common decision for Θ. Then

$$Rul_G(\Theta, r) = \{\rightarrow d \neq k\}.$$

Let now Θ be a nonterminal node of G such that for each child Θ' of Θ and for each row r' of Θ', a set of rules $Rul_G(\Theta', r')$ is already defined. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. For any $f_i \in E_G(\Theta, r)$, we define the set of rules $Rul_G(\Theta, r, f_i)$ as follows:

$$Rul_G(\Theta, r, f_i) = \{f_i = b_i \wedge \sigma \rightarrow d \neq s : \sigma \rightarrow d \neq s \in Rul_G(\Theta(f_i, b_i), r)\}.$$

Then

$$Rul_G(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r)} Rul_G(\Theta, r, f_i).$$

Theorem 1. *For each node Θ of $\Lambda_\gamma(T)$ and for each row r of Θ, the set $Rul_{\Lambda_\gamma(T)}(\Theta, r)$ is equal to the set of all nonredundant γ-inhibitory rules for Θ and r over T.*

Example 1. To illustrate the algorithm presented above, we consider an example based on decision table T_0 (see Fig.1). In the example we set $\gamma = 1$, so during the construction of the graph $\Lambda_1(T_0)$ we stop the partitioning of a subtable Θ of T_0 when $N_{lcd}(\Theta) \leq 1$ (see Fig.1). We denote $G = \Lambda_1(T_0)$.

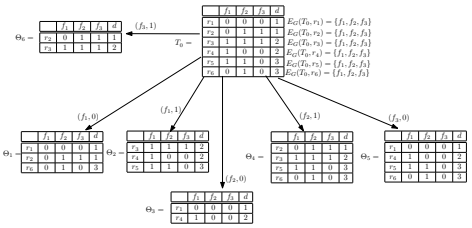


Fig. 1. Graph $G = \Lambda_1(T_0)$

For each node Θ of the graph G and for each row r of Θ we describe a set $Rul_G(\Theta, r)$. We move from terminal nodes of G to the node T_0. Terminal nodes of the graph G are Θ_1, Θ_2, Θ_3, Θ_4, Θ_5, Θ_6. For these nodes we have:

$Rul_G(\Theta_1, r_1) = Rul_G(\Theta_1, r_2) = Rul_G(\Theta_1, r_6) = \{\rightarrow d \neq 2\}$,
$Rul_G(\Theta_2, r_3) = Rul_G(\Theta_2, r_4) = Rul_G(\Theta_2, r_5) = \{\rightarrow d \neq 1\}$,
$Rul_G(\Theta_3, r_1) = Rul_G(\Theta_3, r_4) = \{\rightarrow d \neq 3\}$,
$Rul_G(\Theta_4, r_2) = Rul_G(\Theta_4, r_3) = Rul_G(\Theta_4, r_5) = Rul_G(\Theta_4, r_6) = \{\rightarrow d \neq 1\}$,
$Rul_G(\Theta_5, r_1) = Rul_G(\Theta_5, r_4) = Rul_G(\Theta_5, r_5) = Rul_G(\Theta_5, r_6) = \{\rightarrow d \neq 1\}$,
$Rul_G(\Theta_6, r_2) = Rul_G(\Theta_6, r_3) = \{\rightarrow d \neq 3\}$.

Now we can describe the sets of rules corresponding to rows of T_0. This is a nonterminal node of G for which all children Θ_1, Θ_2, Θ_3, Θ_4, Θ_5 and Θ_6 are already treated. We have:

$Rul_G(T_0, r_1) = \{f_1 = 0 \rightarrow d \neq 2, f_2 = 0 \rightarrow d \neq 3, f_3 = 0 \rightarrow d \neq 1\}$,
$Rul_G(T_0, r_2) = \{f_1 = 0 \rightarrow d \neq 2, f_2 = 1 \rightarrow d \neq 1, f_3 = 1 \rightarrow d \neq 3\}$,
$Rul_G(T_0, r_3) = \{f_1 = 1 \rightarrow d \neq 1, f_2 = 1 \rightarrow d \neq 1, f_3 = 1 \rightarrow d \neq 3\}$,
$Rul_G(T_0, r_4) = \{f_1 = 1 \rightarrow d \neq 1, f_2 = 0 \rightarrow d \neq 3, f_3 = 0 \rightarrow d \neq 1\}$,
$Rul_G(T_0, r_5) = \{f_1 = 1 \rightarrow d \neq 1, f_2 = 1 \rightarrow d \neq 1, f_3 = 0 \rightarrow d \neq 1\}$,
$Rul_G(T_0, r_6) = \{f_1 = 0 \rightarrow d \neq 2, f_2 = 1 \rightarrow d \neq 1, f_3 = 0 \rightarrow d \neq 1\}$.

4 Procedures of Optimization Relative to Length, Coverage and Number of Misclassifications

We start describing the procedure of optimization of the graph G relative to the length l. For each node Θ in the graph G, this procedure corresponds to each row r of Θ the set $Rul^l_G(\Theta, r)$ of γ-inhibitory rules with the minimum length from $Rul_G(\Theta, r)$ and the number $Opt^l_G(\Theta, r)$ – the minimum length of a γ-inhibitory rule from $Rul_G(\Theta, r)$.

We traverse from the terminal nodes of the graph G to the node T. Then, we assign to each row r of each table Θ the number $Opt^l_G(\Theta, r)$ and we change the set $E_G(\Theta, r)$ attached to the row r in Θ if Θ is a nonterminal node of G. We denote the obtained graph by $G(l)$.

Let Θ be a terminal node of G. Then we correspond the number

$$Opt^l_G(\Theta, r) = 0$$

to each row r of Θ.

Let Θ be a nonterminal node of G and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We correspond the number

$$Opt^l_G(\Theta, r) = \min\{Opt^l_G(\Theta(f_i, b_i), r) + 1 : f_i \in E_G(\Theta, r)\}$$

to the row r in the table Θ and we set

$$E_{G(l)}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^l_G(\Theta(f_i, b_i), r) + 1 = Opt^l_G(\Theta, r)\}.$$

Theorem 2. *For each node Θ of the graph $G(l)$ and for each row r of Θ the set $Rul_{G(l)}(\Theta, r)$ is equal to the set $Rul^l_G(\Theta, r)$ of all γ-inhibitory rules with the minimum length from the set $Rul_G(\Theta, r)$.*

We consider now the procedure of optimization of the graph G relative to the coverage c. For each node Θ in the graph G, this procedure corresponds to each row r of Θ the set $Rul^c_G(\Theta, r)$ of γ-inhibitory rules with maximum coverage from $Rul_G(\Theta, r)$ and the number $Opt^c_G(\Theta, r)$ – the maximum coverage of a γ-inhibitory rule from $Rul_G(\Theta, r)$.

We move from the terminal nodes of the graph G to the node T. We assign to each row r of each table Θ the number $Opt^c_G(\Theta, r)$ which is the maximum coverage of a γ-inhibitory rule from $Rul_G(\Theta, r)$ and change the set $E_G(\Theta, r)$ attached to the row r in Θ if Θ is a nonterminal node G. We denote the obtained graph by $G\,(c)$.

Let Θ be a terminal node of G. Then we assign the number

$$Opt^c_G(\Theta, r) = N(\Theta) - N_{lcd}(\Theta)$$

to each row r of Θ.

Let Θ be a nonterminal node of G and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We assign the number

$$Opt^c_G(\Theta, r) = \min\{Opt^c_G(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$$

to the row r in the table Θ and we set

$$E_{G(c)}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^c_G(\Theta(f_i, b_i), r) = Opt^c_G(\Theta, r)\}.$$

Theorem 3. *For each node Θ of the graph $G(c)$ and for each row r of Θ the set $Rul_{G(c)}(\Theta, r)$ is equal to the set $Rul^c_G(\Theta, r)$ of all γ-inhibitory rules with the maximum coverage from the set $Rul_G(\Theta, r)$.*

We consider now the procedure of optimization of the graph G relative to the number of misclassifications μ. For each node Θ in the graph G, this procedure corresponds to each row r of Θ the set $Rul^\mu_G(\Theta, r)$ of γ-inhibitory rules with the minimum number of misclassifications from $Rul_G(\Theta, r)$ and the number $Opt^\mu_G(\Theta, r)$ – the minimum number of misclassifications of a γ-inhibitory rule from $Rul_G(\Theta, r)$.

We move from the terminal nodes of the graph G to the node T. We will correspond to each row r of each table Θ the number $Opt^\mu_G(\Theta, r)$ which is the minimum number of misclassifications of a γ-inhibitory rule from $Rul_G(\Theta, r)$ and we will change the set $E_G(\Theta, r)$ attached to the row r in Θ if Θ is a nonterminal node of G. We denote the obtained graph by $G(\mu)$.

Let Θ be a terminal node of G. Then we correspond to each row r of Θ the number $Opt^\mu_G(\Theta, r)$ which is equal to $N_{lcd}(\Theta)$.

Let Θ be a nonterminal node of G and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We correspond the number

$$Opt^\mu_G(\Theta, r) = \min\{Opt^\mu_G(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$$

to the row r in the table Θ, and we set

$$E_{G(\mu)}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt_G(\mu)(\Theta(f_i, b_i), r) = Opt_G(\mu)(\Theta, r)\}.$$

Theorem 4. *For each node Θ of the graph $G(\mu)$ and for each row r of Θ, the set $Rul_{G(\mu)}(\Theta, r)$ is equal to the set $Rul_G(\mu)(\Theta, r)$ of all γ-inhibitory rules with the minimum number of misclassifications from the set $Rul_G(\Theta, r)$.*

Example 2. Figure 2 presents the directed acyclic graph $G(\mu)$ obtained from the graph G (see Fig. 1) by the procedure of optimization relative to the number of misclassifications. Using the graph $G(\mu)$ we can describe for each row r_i, $i = 1, \ldots, 6$, of the table T_0 the set $Rul^{\mu}_G(T_0, r_i)$ of all nonredundant 1-inhibitory rules for T_0 and r_i over T_0 with the minimum number of misclassifications. We give also the value $Opt^{\mu}_G(T_0, r_i)$ which is equal to the minimum number of misclassifications of a nonredundant 1-inhibitory rule for T_0 and r_i over T_0. This value was obtained during the procedure of optimization of the graph G relative to the number of misclassifications. We have:

$Rul_{G^{\mu}}(T_0, r_1) = \{f_1 = 0 \to \neq 2, f_2 = 0 \to d \neq 3\}$, $Opt^{\mu}_G(T_0, r_1) = 0$,
$Rul_{G^{\mu}}(T_0, r_2) = \{f_1 = 0 \to \neq 2, f_3 = 1 \to d \neq 3\}$, $Opt^{\mu}_G(T_0, r_2) = 0$,
$Rul_{G^{\mu}}(T_0, r_3) = \{f_1 = 1 \to \neq 1, f_3 = 1 \to d \neq 3\}$, $Opt^{\mu}_G(T_0, r_3) = 0$,
$Rul_{G^{\mu}}(T_0, r_4) = \{f_1 = 1 \to \neq 1, f_2 = 0 \to d \neq 3\}$, $Opt^{\mu}_G(T_0, r_4) = 0$,
$Rul_{G^{\mu}}(T_0, r_5) = \{f_1 = 1 \to d \neq 1\}$, $Opt^{\mu}_G(T_0, r_5) = 0$,
$Rul_{G^{\mu}}(T_0, r_6) = \{f_1 = 0 \to d \neq 2\}$, $Opt^{\mu}_G(T_0, r_6) = 0$.

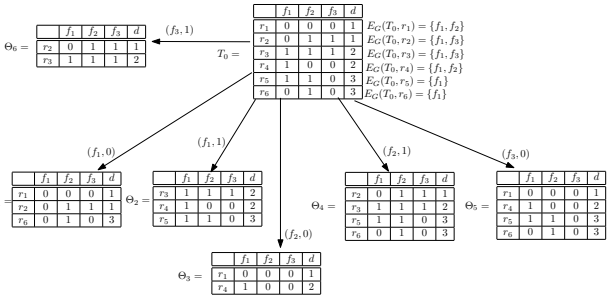


Fig. 2. Graph $G(\mu)$

5 Sequential Optimization

Theorems 2-4 show that we can make sequential optimization relative to the length, coverage and number of misclassifications. We can find all nonredundant γ-inhibitory rules with maximum coverage and after that among these rules find all rules with minimum length. We can continue and find among the obtained rules all rules with minimum number of misclassifications. The order of optimization can be changed.

Sequential optimization relative to the three cost functions allows us to discover the existence of so-called *totally optimal* nonredundant γ-inhibitory rules for a given table T and its row r. A nonredundant γ-inhibitory rule τ for T and r

is called totally optimal if τ has simultaneously the minimum length, the maximum coverage and the minimum number of misclassifications among all possible nonredundant γ-inhibitory rules for T and r. Note that the results of sequential optimization of rules for T and r relative to length, coverage and number of misclassifications do not depend on the order of optimization if and only if there is a totally optimal nonredundant γ-inhibitory rule for T and r.

Example 3. Figure 3 shows the result of the work of sequential optimization procedures relative to the number of misclassifications, length and coverage respectivley. The graph $G(\mu lc)$ is obtained from the graph $G(\mu)$ when sequential procedures relative to length and coverage are applied on the graph G^{μ}. Based

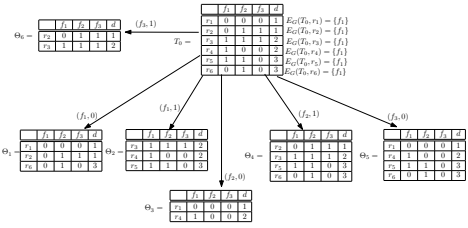


Fig. 3. Graph $G(\mu lc)$

on the graph $G(\mu lc)$, we correspond to row r_i, $i = 1, \ldots, 6$, of the table T_0 the set $Rul_{G^{\mu lc}}(T_0, r_i)$ of all nonredundant 1-inhibitory rules for T_0 and r_i which have the maximum coverage among all nonredundant 1-inhibitory rules for T_0 and r_i with the minimum length. So, we have:

$Rul_{G^{\mu lc}}(T_0, r_1) = \{f_1 = 0 \to d \neq 2\}$,
$Rul_{G^{\mu lc}}(T_0, r_2) = \{f_1 = 0 \to d \neq 2\}$,
$Rul_{G^{\mu lc}}(T_0, r_3) = \{f_1 = 1 \to d \neq 1\}$,
$Rul_{G^{\mu lc}}(T_0, r_4) = \{f_1 = 1 \to d \neq 1\}$,
$Rul_{G^{\mu lc}}(T_0, r_5) = \{f_1 = 1 \to d \neq 1\}$,
$Rul_{G^{\mu lc}}(T_0, r_6) = \{f_1 = 0 \to d \neq 2\}$.

As a result, we have for each row r_i, $i = 1, \ldots, 6$, a totally optimal nonredundant 1-inhibitory rule relative to number of misclassification, the length and coverage.

6 Greedy Algorithm

Let T be a decision table with n columns labeled with conditional attributes $f_1, \ldots, f_n$ and a decision attribute d, and $r = (b_1, \ldots, b_n)$ be a row of T. We now describe a greedy algorithm which constructs a γ-inhibitory rule for T and r.

If $N_{lcd}(T) \leq \gamma$ then the output of the algorithm is the rule $\to d \neq k$ where k is the least common decision for T. Let now $N_{lcd}(T) > \gamma$ and let us assume that the

greedy algorithm has already chosen the attributes $f_{i_1}, \ldots, f_{i_m}$. If $N_{lcd}(T') \leq \gamma$ where $T' = T\left(f_{i_1}, b_{i_1}\right) \ldots \left(f_{i_m}, b_{i_m}\right)$ then the output of the algorithm is the rule

$$f_{i_1} = b_{i_1} \wedge \ldots \wedge f_{i_m} = b_{i_m} \rightarrow d \neq k$$

where k is the least common decision for T'. Otherwise, the algorithm will choose an attribute f_{i_m} for which the value of $N_{lcd}\left(T'\left(f_{i_{m+1}}, b_{i_{m+1}}\right)\right)$ is minimum, etc.

7 Experimental Results

For experiments we use decision tables from the UCI Machine Learning Repository [10]. We preprocess the decision tables by eliminating attributes which, each row, take unique value such as ID number, merging identical rows into a single row with the most common decision for the group of identical rows, and imputing missing values with the most common value of the corresponding attribute.

Let T be one of these decision tables. We consider for this table the value of $N_{lcd}(T)$ and values of γ from the set $\Gamma(T) = \{\lfloor N_{lcd}(T) \times 0.2 \rfloor, \lfloor N_{lcd}(T) \times 0.3 \rfloor, \lfloor N_{lcd}(T) \times 0.5 \rfloor\}$. These parameters can be found in Table 1, where (i) column "Rows" contains the number of rows, (ii) column "Attributes" contains the number of conditional attributes, (iii) column "$N_{lcd}(T)$" contains the number of rows with the least common decision for T, and (iv) column "$\gamma \in \Gamma(T)$" contains values from $\Gamma(T)$. Table 2 allows us to compare the number of rows in

Table 1. Parameters of decision tables and values of γ

Decision table	Rows	Attributes	$N_{lcd}(T)$	$\gamma \in \Gamma(T)$		
				$\lfloor N_{lcd}(T) \times 0.2 \rfloor$	$\lfloor N_{lcd}(T) \times 0.3 \rfloor$	$\lfloor N_{lcd}(T) \times 0.5 \rfloor$
Balance-scale	625	4	49	9	14	24
Breast-cancer	266	9	76	15	22	38
Cars	1728	6	65	13	19	32
Hayes-roth	69	4	18	3	5	9
Shuttle-landing	15	6	6	1	1	3
Soybean-small	47	35	10	2	3	5
Zoo	59	16	4	0	1	2
Tic-tac-toe	959	9	332	66	99	166

decision table T (column "Rows") with the number of rows with totally optimal nonredundant γ-inhibitory rules (columns "t-o-rows"). Also this table contains information about minimum, average and maximum number of totally optimal nonredundant γ-inhibitory rules for rows. It is interesting to note that, with the growth of γ, the number of rows with totally optimal nonredundant γ-inhibitory rules can (i) decrease, (ii) increase, (iii) fluctuate, and (iv) be stable.

Tables 3, 4 and 5 shows the behaviour of complexity parameters (average length, coverage, and number of misclassifications of rules) depending on the order of optimization. When there is at least one totally optimal rule for each row of a given decision table, the outputs of procedures of optimization are the same for different orders.

Table 2. Number of rows with totally optimal nonredundant γ-inhibitory rules

Decision table	Rows	$\lfloor N_{lcd} \times 0.2 \rfloor$				$\lfloor N_{lcd} \times 0.3 \rfloor$				$\lfloor N_{lcd} \times 0.5 \rfloor$			
		t-o-rows	t-opt rules			t-o-rows	t-opt rules			t-o-rows	t-opt rules		
			min	avg	max		min	avg	max		min	avg	max
Balance-scale	625	97	0	0.61	8	625	1	1.95	4	625	1	1.95	4
Breast-cancer	266	0	0	0	0	0	0	0	0	0	0	0	0
Cars	1728	1688	0	1.51	3	1664	0	1.48	3	1472	0	1.33	3
Hayes-roth	69	37	0	0.72	2	0	0	0	0	33	0	0.65	2
Shuttle-landing	15	7	0	0.6	2	7	0	0.6	2	0	0	0	0
Soybean-small	47	47	1	2.36	3	47	1	2.36	3	47	1	2.36	3
Zoo	59	59	1	1.05	2	59	1	1.05	2	59	1	1.05	2
Tic-tac-toe	958	0	0	0	0	0	0	0	0	50	0	0.05	1

Table 3. Sequential optimization for $\gamma = \lfloor N_{lcd} \times 0.2 \rfloor$

Decision table	$\gamma \in \Gamma(T)$								
	$l + c + \mu$			$c + l + \mu$			$\mu + l + c$		
	l	c	μ	c	l	μ	μ	l	c
Balance-scale	1.13	104.08	7.96	104.08	1.13	7.96	0.92	1.97	26.64
Breast-cancer	1.07	33.77	10.45	70.37	2.29	14.47	0.71	2.45	9.83
Cars	1.03	547.78	0.22	547.78	1.03	0.22	0.0	1.05	543.70
Hayes-roth	1.57	9.97	0.61	9.97	1.57	0.61	0.10	1.57	8.19
Shuttle-landing	1.13	2.0	0.27	3.0	2.40	0.60	0.07	1.33	1.93
Soybean-small	1.0	37.0	0.0	37.0	1.0	0.0	0.0	1.0	37.0
Zoo	1.0	50.46	0.0	50.46	1.0	0.0	0.0	1.0	50.46
Tic-tac-toe	1.29	138.80	54.58	140.78	1.60	46.30	10.52	2.61	34.73

Table 4. Sequential optimization for $\gamma = \lfloor N_{lcd} \times 0.3 \rfloor$

Decision table	$\gamma \in \Gamma(T)$								
	$l + c + \mu$			$c + l + \mu$			$\mu + l + c$		
	l	c	μ	c	l	μ	μ	l	c
Balance-scale	1	115.87	9.13	115.87	1	9.13	9.13	1	115.87
Breast-cancer	1.0	45.83	15.67	79.62	1.97	19.49	1.31	2.03	10.74
Cars	1.0	554.10	0.56	554.10	1.0	0.56	0.0	1.05	543.70
Hayes-roth	1.0	17.74	4.26	17.74	1.0	4.26	0.54	1.57	8.88
Shuttle-landing	1.13	2.0	0.27	3.0	2.40	0.60	0.07	1.33	1.93
Soybean-small	1.0	37.0	0.0	37.0	1.0	0.0	0.0	1.0	37.0
Zoo	1.0	50.46	0.0	50.46	1.0	0.0	0.0	1.0	50.46
Tic-tac-toe	1.03	259.62	81.86	259.90	1.04	81.92	16.59	2.0	64.55

Table 5. Sequential optimization for $\gamma = \lfloor N_{lcd} \times 0.5 \rfloor$

Decision table	$\gamma \in \Gamma(T)$								
	$l + c + \mu$			$c + l + \mu$			$\mu + l + c$		
	l	c	μ	c	l	μ	μ	l	c
Balance-scale	1	115.87	9.13	115.87	1	9.13	9.13	1	115.87
Breast-cancer	1.0	138.30	35.38	140.39	1.05	35.40	3.20	1.92	13.87
Cars	1.0	572.44	3.56	572.44	1.0	3.56	0.37	1.03	545.96
Hayes-roth	1.0	17.74	4.26	17.74	1.0	4.26	2.26	1.0	14.61
Shuttle-landing	1.07	2.20	0.40	6.67	2.67	3.0	0.20	1.20	1.93
Soybean-small	1.0	37.0	0.0	37.0	1.0	0.0	0.0	1.0	37.0
Zoo	1.0	50.46	0.0	50.46	1.0	0.0	0.0	1.0	50.46
Tic-tac-toe	1.0	327.79	108.46	327.79	1.0	108.46	66.28	1.0	157.55

Table 6 shows the average results for the system of rules constructed by the greedy algorithm, for $\gamma \in \Gamma(T)$, in terms of length, coverage and number of misclassifications, where the algorithm corresponds a γ-inhibitory rule for each row of a given decision table.

We compare the results obtained by the greedy algorithm with optimal results by computing relative differences $((l_{\text{greedy}} - l_{\text{opt}})/l_{\text{opt}}, (c_{\text{opt}} - c_{\text{greedy}})/c_{\text{opt}}, (\mu_{\text{greedy}} - \mu_{\text{opt}})/\mu_{\text{opt}}))$ as shown in Table 7. If $\mu_{\text{opt}} = 0$ then we write "$\mu_{\text{greedy}}/0$".

Table 6. Results of greedy algorithm work

Decision Table	$\gamma \in \Gamma(T)$								
	$\lfloor N_{lcd}(T) \times 0.2 \rfloor$			$\lfloor N_{lcd}(T) \times 0.3 \rfloor$			$\lfloor N_{lcd}(T) \times 0.5 \rfloor$		
	l	c	u	l	c	u	l	c	u
Balance-scale	1.13	104.05	7.99	1.12	115.87	9.13	1.12	115.87	9.13
Breast-cancer	1.07	20.44	6.87	1.0	22.02	7.71	1.0	22.19	7.83
Cars	1.03	453.83	0.25	1.46	462.10	0.56	1.46	462.10	0.56
Hayes-roth	1.57	7.93	0.25	1	13.83	2.26	1	13.83	2.26
Shuttle-landing	1.13	2	0.27	1.33	2	0.27	1.07	2	0.33
Soybean-small	1	9.93	0	1	9.93	0	1	9.93	0
Zoo	1	30.63	0	1	30.63	0	1	30.63	0
Tic-tac-toe	1.29	119.44	48.04	1.03	152.56	63.86	1	157.55	66.28

Table 7. Relative difference between results of greedy algorithm and optimal results

Decision Table	$\gamma \in \Gamma(T)$								
	$\lfloor N_{lcd}(T) \times 0.2 \rfloor$			$\lfloor N_{lcd}(T) \times 0.3 \rfloor$			$\lfloor N_{lcd}(T) \times 0.5 \rfloor$		
	l	c	u	l	c	u	l	c	u
Balance-scale	0	0.0	7.68	0.12	0	0	0.12	0	0
Breast-cancer	0	0.71	8.68	0	0.72	4.89	0	0.84	1.45
Cars	0	0.17	**0.25/0**	0.46	0.17	**0.56/0**	0.46	0.19	0.51
Hayes-roth-data	0	0.21	1.5	0	0.22	3.19	0	0.22	0
Shuttle-landing	0	0.33	2.86	0.178	0.33	2.86	0	0.7	0.65
Soybean-small	0	0.73	**0/0**	0	0.73	**0/0**	0	0.73	**0/0**
Zoo	0	0.39	**0/0**	0	0.39	**0/0**	0	0.39	**0/0**
Tic-tac-toe	0	0.15	3.57	0	0.41	2.85	0	0.52	0

The obtained results show that the greedy algorithm is more suitable for the minimization of length than for maximization of coverage or minimization of the number of misclassifications.

8 Conclusions

The paper considered (from theoretical and experimental points of view) sequential optimization of approximate inhibitory rules relative to three cost functions. It included also a comparison of dynamic programming algorithms with a greedy algorithm. The proposed algorithms can be useful for knowledge extraction and representation.

References

1. Skowron, A., Suraj, Z.: Rough sets and concurrency. Bulletin of the Polish Academy of Sciences 41(3), 237–254 (1993)
2. Suraj, Z.: Some remarks on extensions and restrictions of information systems. In: Ziarko, W.P., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 204–211. Springer, Heidelberg (2001)
3. Delimata, P., Moshkov, M., Skowron, A., Suraj, Z.: Inhibitory Rules in Data Analysis: A Rough Set Approach. SCI, vol. 163. Springer, Heidelberg (2009)
4. Delimata, P., Moshkov, M., Skowron, A., Suraj, Z.: Two families of classification algorithms. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 297–304. Springer, Heidelberg (2007)
5. Delimata, P., Moshkov, M., Skowron, A., Suraj, Z.: Lazy classification algorithms based on deterministic and inhibitory decision rules. In: Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 1773–1778 (2008)
6. Delimata, P., Moshkov, M., Skowron, A., Suraj, Z.: Comparison of lazy classification algorithms based on deterministic and inhibitory decision rules. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 55–62. Springer, Heidelberg (2008)
7. Alsolami, F., Chikalov, I., Moshkov, M., Zielosko, B.: Optimization of inhibitory decision rules relative to length and coverage. 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. 149–154. Springer, Heidelberg (2012)
8. Alsolami, F., Chikalov, I., Moshkov, M., Zielosko, B.M.: Length and coverage of inhibitory decision rules. In: Nguyen, N.-T., Hoang, K., Jędrzejowicz, P. (eds.) ICCCI 2012, Part II. LNCS, vol. 7654, pp. 325–334. Springer, Heidelberg (2012)
9. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Optimization of approximate decision rules relative to number of misclassifications: Comparison of greedy and dynamic programming approaches. In: Graña, M., Toro, C., Howlett, R.J., Jain, L.C. (eds.) KES 2012. LNCS, vol. 7828, pp. 41–50. Springer, Heidelberg (2013)
10. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007), `http://www.ics.uci.edu/~mlearn/`

Robustness Measure of Decision Rules

Motoyuki Ohki and Masahiro Inuiguchi

Graduate School of Engineering Science, Osaka University
Toyonaka, Osaka, 560-8531, Japan

Abstract. Rough set approaches provide useful tools to find minimal decision rules. The obtained minimal decision rules are used to classify unseen objects. On the other hand, the condition parts of the minimal decision rules are sometimes used to design new objects which will be classified into the target decision class. While we are interested in the goodness of the set of obtained minimal decision rules in the former case, we are interested in the goodness of an individual minimal decision rule in the latter case. In this paper, we propose robustness measure as a new type of evaluation index for decision rules. The measure evaluates to what extent the decision rule maintains the goodness of classification against the partially-matched data. Numerical experiments are conducted to examine the effectiveness of robustness measure.

Keywords: rough set, decision rule, interestingness measure, robustness.

1 Introduction

Rough set approaches originated by Pawlak [1] provide useful tools to find minimal decision rules and applied to various fields such as medicine, engineering, management, economy and so on. The obtained minimal decision rules are used to classify unseen objects. On the other hand, the conditions of the minimal decision rules are sometimes used to design new objects which will be classified into the target decision class. For example, in Kansei engineering, the minimal decision rules inferring popular items are regarded as the design knowledge and new items satisfying their conditions are designed to attract more customers and to enhance customer satisfaction [2]. While we are interested in the goodness of the set of obtained minimal decision rules with which we build a classifier system in the former case, we are interested in the goodness of individual minimal decision rule useful as design knowledge in the latter case. A good classifier could be produced from a set of good decision rules. In this sense, the study of good decision rules may serve as a foundation for building a good classifier as well as for obtaining good design knowledge.

In this paper, we investigate the evaluation of decision rules rather than the evaluation of the set of decision rules. With regard to this issue, various quantitative measures of rule interestingness (attractiveness), called *interestingness measures*, have been proposed [3–6]. An interestingness measure is useful as an index evaluating the usefulness and effectiveness of decision rules which are not

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 166–177, 2013.

always 100% confident even for the given decision table. Interestingness measures evaluate the strongness of relationship between the whole body of conditions and the conclusion of a decision rule and never evaluate the strongness of relationship between a part of conditions and the conclusion. Considering the possibility of partially-matched data in unseen objects, we could give a good evaluation to the decision rule maintaining the confidence of its conclusion against the partially-matched data. Such a decision rule would be also good for design knowledge because the new designed items may fail to satisfy a few conditions of the decision rule despite the designer's intention. From this point of view, we propose robustness measures as a new type of evaluation index for decision rules. The robustness measures of a decision rule evaluate to what extent the interestingness is preserved from the removal of a part of condition.

This paper is organized as follows. A brief survey of the interestingness measure is given in next section. After an example illustrating the necessity of robustness measure is given, several concepts of robustness are described in Section 3. Then the robustness measures are defined also in Section 3. In Section 4, we conduct three numerical experiments. By those experiments, we confirm the usefulness and effectiveness of decision rules with high robusteness scores as the design knowledge, in the classification of unseen objects and in the classification of objects with missing values.

2 Interestingness and the Idea of Robustness

Interestingness measures have been developed to build a good classification system composed of association rules [3–6]. The support and confidence are the most used interestingness measures. Let $supp(E)$ be the support of statement E, i.e., the number of objects in the data-set for which E is true. Then the support of rule $E \to H$ is defined by $supp(E \to H) = supp(E \wedge H)$. On the other hand, the confidence of rule $E \to H$ is defined by $conf(E \to H) = supp(E \wedge H)/supp(E)$. As a counterpart of the confidence, the recall of rule $E \to H$ is defined by $rec(E \to H) = supp(E \wedge H)/supp(H)$. Other than those, there are a lot of interestingness measures have been proposed in the literature [3–6]. Those measures are different in their characteristics such as generality, conciseness, reliability, novelty, surprisingness, utility, actionability, and so on (see [4]).

In rough set approaches to rule induction, except variable-precision and variable-consistency models, only 100% confident rules $E \to H$ with minimal condition E called decision rules are induced. Namely, for such a rule $E \to H$, we have $conf(E \to H) = 1$ and we cannot find any other rule $E' \to H$ such that $conf(E' \to H) = 1$ and $conf(E' \to E) = 1$. Therefore, the confidence does not work well and the support $supp(E \to H)$ and the recall $rec(E \to H)$ are often used for evaluation of rules. Here we note that, unlike association rules, a decision rule has its premise E described by the minimal condition attributes and its conclusion H described by decision attributes. In rough set approaches, the confidence and the recall are called the accuracy and the coverage, respectively.

The condition E of a rule $E \to H$ is composed of several elementary conditions e_i, $i = 1, 2, \ldots, p$. Namely, it is expressed as the conjunction of

elementary conditions e_i, $i = 1, 2, \ldots, p$, i.e., $E = e_1 \wedge e_2 \wedge \cdots \wedge e_p$. For the sake of simplicity, $E = e_1 \wedge e_2 \wedge \cdots \wedge e_p$ is written as $E = \{e_1, e_2, \ldots, e_p\}$. Then $B = \{b_1, b_2, \ldots, b_q\} \subseteq E$ means a relaxed condition $B = b_1 \wedge b_2 \wedge \cdots \wedge b_q$ of E and rule $B \to H$ becomes a coarser rule of $E \to H$. For convenience, we define $supp(\emptyset) = |U|$, where U is the set of objects in the given data-set.

When we evaluate a rule $E \to H$ by an interestingness measure, we do not take into consideration the evaluations of any coarser rules $B \to H$ with relaxed conditions $B \subset E$. Under static, noise-free and error-free environment, this evaluation would work very well. However, when decision rules may change with situation and when observed data may include noise and error, the evaluation of decision rules without considerations of its coarser rules would not be always sufficient.

For example, when an induced rule $E \to H$ shows the condition (E) of popular goods (H), a designer may design a new product satisfying the condition $E = e_1 \wedge e_2 \wedge \cdots \wedge e_p$. If the designer's understanding of an elementary condition e_i of E is different from the customers' understandings, he/she may fail to sell the new product by this mismatch. Moreover, when a given data-set is not sufficient to express all cases, the induced decision rules are not perfect. Then we may come across partially matched new data to the induced decision rules. It would be useful if the matched part B of condition E may work sufficiently to conclude H even with some errors.

From this point of view, we propose robustness measures to evaluate rules $E \to H$ induced by rough set approaches. In the robustness measures, we take into consideration the evaluation of coarser rules to evaluate the induced decision rules. In the references [7, 8], another kind of robustness measure is defined for association rules. While this robustness measure evaluates the fragility of an association rule against noise in the given data-set, the proposing robustness measure evaluates the usefulness of a decision rule against partially matched data.

3 Robustness Measure

We propose robustness measures to evaluate decision rules induced by rough set approaches including variable-precision and variable-consistency models. To illustrate the concept of the proposing robustness measure, we consider a decision table shown in Table 1. In this decision table, the frequency (fr) of each pattern (pat) is shown in the last column. The lower approximations of decision classes in the classical rough set approach are small and thus the application of variable-precision rough set approach would be adequate. By variable-precision rough set approach, we may induce rules r_1: $(a_1 = 1) \wedge (a_2 = 1) \wedge (a_3 = 1) \to (d = 1)$ with confidence 0.863636 and r_2: $(a_2 = 1) \wedge (a_3 = 1) \wedge (a_5 = 1) \to (d = 1)$ with confidence 0.785714. For those rules, let us consider the confidence of their coarser rules with more relaxed conditions. The values of the confidence and support of those coarser rules are shown in Table 2. Comparing confidence values of coarser rules of r_1 and r_2 having the same number of dropped elementary conditions in

Table 1. An illustrative decision table

pat	a_1	a_2	a_3	a_4	a_5	d	fr	pat	a_1	a_2	a_3	a_4	a_5	d	fr	pat	a_1	a_2	a_3	a_4	a_5	d	fr
p_1	1	1	1	0	0	1	32	p_7	0	1	1	0	1	1	22	p_{12}	1	1	1	1	0	1	6
p_2	1	1	1	0	0	0	2	p_8	0	1	1	0	1	0	5	p_{13}	1	1	0	0	1	0	4
p_3	1	1	0	1	0	0	20	p_9	0	1	1	1	1	1	8	p_{14}	1	1	1	1	0	0	2
p_4	1	0	1	0	0	0	18	p_{10}	0	1	1	1	1	0	4	p_{15}	1	0	1	0	1	1	2
p_5	0	1	1	1	0	0	10	p_{11}	0	1	1	1	1	1	3	p_{16}	1	0	1	0	1	0	2
p_6	1	1	1	1	0	0	2																

Table 2. The values of confidence of coarser rules

name	rule	confidence	support
r_1	$(a_1 = 1) \wedge (a_2 = 1) \wedge (a_3 = 1) \rightarrow (d = 1)$	0.863636	38
$r_1(\overline{a_3})$	$(a_1 = 1) \wedge (a_2 = 1) \rightarrow (d = 1)$	0.558824	38
$r_1(\overline{a_2})$	$(a_1 = 1) \wedge (a_3 = 1) \rightarrow (d = 1)$	0.606061	40
$r_1(\overline{a_1})$	$(a_2 = 1) \wedge (a_3 = 1) \rightarrow (d = 1)$	0.739583	71
$r_1(\overline{a_2 a_3})$	$(a_1 = 1) \rightarrow (d = 1)$	0.444444	40
$r_1(\overline{a_1 a_3})$	$(a_2 = 1) \rightarrow (d = 1)$	0.591667	71
$r_1(\overline{a_1 a_2})$	$(a_3 = 1) \rightarrow (d = 1)$	0.618644	73
r_2	$(a_2 = 1) \wedge (a_3 = 1) \wedge (a_5 = 1) \rightarrow (d = 1)$	0.785714	33
$r_2(\overline{a_5})$	$(a_2 = 1) \wedge (a_3 = 1) \rightarrow (d = 1)$	0.739583	71
$r_2(\overline{a_3})$	$(a_2 = 1) \wedge (a_5 = 1) \rightarrow (d = 1)$	0.717391	33
$r_2(\overline{a_2})$	$(a_3 = 1) \wedge (a_5 = 1) \rightarrow (d = 1)$	0.760870	35
$r_2(\overline{a_3 a_5})$	$(a_2 = 1) \rightarrow (d = 1)$	0.591667	71
$r_2(\overline{a_2 a_5})$	$(a_3 = 1) \rightarrow (d = 1)$	0.618644	73
$r_2(\overline{a_2 a_3})$	$(a_5 = 1) \rightarrow (d = 1)$	0.7	35

Table 2, we observe that the coarser rules of r_2 take higher confidence values than those of r_1 while r_2 takes lower confidence value than r_1. When we consider only confidence values of rules, we evaluate r_1 is more interesting than r_2. However, considering that the induced rule may be used for partially-matched data, r_2 would be more interesting. As shown in Table 2, we cannot find such an advantage of r_2 over r_1 even if we add the evaluation by support.

From this point of view, we propose robustness measures. As is seen in the previous example, robustness measures are defined by using some interestingness measure $f(E \rightarrow H)$. To formulate robustness measures, we assume the following two settings:

S1. The larger $f(B \rightarrow H)$, the more interesting rule $B \rightarrow H$. (Gain property)
S2. $f(B \rightarrow H) < f(E \rightarrow H),\ \forall B \subset E.$ (minimality of E)

S1 means that f shows the favorability rather than inadvisability. S2 means that the interestingness decreases by the strict relaxation of the condition of rule $E \rightarrow H$. This implies that E is a minimal description such that the interestingness is not less than $f(E \rightarrow H)$. Interestingness measures such as support, recall, relative risk and so on cannot satisfy S2. On the contrary, interestingness measures such as confidence, lift ($lift(E \rightarrow H) = |U| conf(E \rightarrow H)/supp(H)$), and so on may satisfy S2. Moreover, S2 implies that only such minimal rule

$E \to H$ is considered. For a while we assume this strong setting S2 but once the robustness measures are formulated, they are useful for any rule.

Using an adequate interestingness measure f, the preserved f-value from the lack of all conditional attributes in L under rule $E \to H$ is defined by

$$pres_f(E \to H; L) = f(E_{-L} \to H), \tag{1}$$

where $L \subseteq C$ and C is the set of all condition attributes. E_{-L} is the subset of elementary conditions of E which does not specify the value of condition attribute in L. When $E_{-L} = E$, $pres_f$ takes its maximum ($= f(E \to H)$) under setting S2.

We note that when $E_{-L} = \emptyset$, $pres_f$ does not always take the minimum. Moreover, for some L such that $E_{-L} \neq E$, we may have $pres_f(E \to H; L) = f(E_{-L} \to H) < f(\emptyset \to H)$. This inequality implies that rule $E_{-L} \to H$ is less interesting than rule saying "everything satisfies H" in the sense of interestingness measure of f. If this inequality holds for some L, rule $E_{-L} \to H$ may deteriorate the reasoning. For example, in Table 1, confidence of $r_1(\overline{a_2 a_3})$, 0.444444 is less than the probability of $(d = 1)$, 0.514085.

Then we define the following concepts of robustness of rule $E \to H$:

Genuine Robustness: rule $E \to H$ is said to be *robust* with respect to f iff

$$pres_f(E \to H; L) \geq f(\emptyset \to H), \ \forall L \subseteq C. \tag{2}$$

Especially, the rule is said to be *strongly robust* with respect to f iff (2) is satisfied with strong inequality.

ε-Robustness: rule $E \to H$ is said to be *ε-robust* iff

$$pres_f(E \to H; L) \geq f(\emptyset \to H) - \varepsilon, \ \forall L \subseteq C. \tag{3}$$

kth-order Robustness: rule $E \to H$ is said to be *kth-order robust* iff

$$pres_f(E \to H; L) \geq f(\emptyset \to H), \ \forall L \subseteq C \text{ such that } |L| \leq k. \tag{4}$$

kth-order ε-Robustness: rule $E \to H$ is said to be *kth-order ε-robust* iff

$$pres_f(E \to H; L) \geq f(\emptyset \to H) - \varepsilon, \ \forall L \subseteq C \text{ such that } |L| \leq k. \tag{5}$$

Expectation-based Robustness: rule $E \to H$ is said to be *expectantly robust* with respect to f iff

$$Ex(pres_f(E \to H; L)) = \sum_{L \subseteq C} P(L) \cdot pres_f(E \to H; L) \geq f(\emptyset \to H), \tag{6}$$

where $Ex(\cdot)$ is an expectation operator and $P(L) \geq 0$ is the probability that values of all condition attributes in L are missing in the given unseen object. We assume $\sum_{L \subseteq C} P(L) = 1$.

Median-based Robustness: rule $E \to H$ is said to be *50%-robust* with respect to f iff

$$Median(pres_f(E \to H; L)) \geq f(\emptyset \to H), \tag{7}$$

where $Median(\cdot)$ is the median operator with respect to $P(L)$ described above.
α-Quantile-based Robustness: rule $E \to H$ is said to be $100\alpha\%$-*robust* with respect to f iff

$$\alpha\text{-}Quantile(pres_f(E \to H; L)) \geq f(\emptyset \to H), \tag{8}$$

where $\alpha \in (0, 1]$ and $\alpha\text{-}Quantile(\cdot)$ is the α-quantile operator with respect to $P(L)$ described above.

We note that the median-based robustness is the special case of the α-quantile-based robustness, i.e., it is 0.5-quantile based robustness. The concepts of ε-robustness and kth-order robustness can be applied to the expectation-based, median-based and α-quantile-based robustness. For example, kth-order expectation-based ε-robust rule is a rule $E \to H$ satisfying

$$Ex(pres_f(E \to H; L)) = \sum_{L \subseteq C} P(L||L| \leq k) \cdot pres_f(E \to H; L) \geq f(\emptyset \to H) - \varepsilon, \tag{9}$$

where $P(L||L| \leq k) \geq 0$ is the conditional probability when $|L|$ is at most k and it satisfies $\sum_{L \subseteq C:|L| \leq k} P(L||L| \leq k) = 1$.

The estimation of $P(L)$ is not an easy task. However, we may specify it in a way with some appropriate assumption. Even if this estimation is not very correct, the proposing robustness measures can work to evaluate the maintenance of the rule interestingness against the partially matched data.

Corresponding to each concept of robustness, we define robustness measure with respect to an interestingness measure f satisfying S1 and S2 for some E. The robustness measures corresponding to the concepts above are shown as follows:

Genuine Robustness Measure:

$$rbst(E \to H) = \min_{L \subseteq C} pres_f(E \to H; L), \tag{10}$$

kth-order Robustness Measure:

$$rbst_k(E \to H) = \min_{L \subseteq C:\ |L| \leq k} pres_f(E \to H; L), \tag{11}$$

Expectation-based Robustness Measure:

$$Ex\text{-}rbst(E \to H) = \sum_{L \subseteq C} P(L) \cdot pres_f(E \to H; L), \tag{12}$$

α-Quantile-based Robustness Measure:

$$\alpha\text{-}rbst(E \to H) = \alpha\text{-}Quantile(pres_f(E \to H; L)), \tag{13}$$

We can also define the kth-order expectation-based robustness measure and the kth-order α-quantile-based robustness measure corresponding to the kth-order expectation-based robustness and the kth-order α-quantile-based robustness, respectively. Comparing the above robustness measures with $f(\emptyset \to H)$, we

Table 3. Eight data-sets

Data-set	$\|U\|$	$\|C\|$	$\|V_d\|$	Data-set	$\|U\|$	$\|C\|$	$\|V_d\|$	Data-set	$\|U\|$	$\|C\|$	$\|V_d\|$
car	1728	6	4	iris	150	4	3	wine	178	13	3
ecoli	336	7	8	nursery	12960	8	5	zoo	101	16	7
glass	214	9	7	soybean	562	35	15				

find the robustness and ε-robustness defined above. Namely, we prefer a rule with larger robustness measure to a rule with smaller robustness measure because it is safer against the information loss.

When E is a minimal condition satisfying S2, those measures take values smaller than $f(E \to H)$. However, we can use those measures for any rules $E \to H$ including non-minimal rules. Therefore, those measures may take larger values than $f(E \to H)$. For example, when $E \to H$ is a minimal rule with 100% confidence, a 100% confident rule $E \cup K \to H$ never takes a smaller value of robustness measure than $E \to H$ but may take a greater value, where K is a set of elementary conditions. In this sense, the robustness criterion is contrasting with the minimal description criterion and also with the recall criterion.

4 Numerical Experiments

4.1 Common Settings of Experiments

We conduct three experiments with different purposes. By the first experiment, we confirm the usefulness of the decision rules with high robustness scores as the design knowledge. By the second experiment, we examine whether the classifier based on decision rules with higher robustness scores classifies unseen objects more accurately. Finally, by the third experiment, we examine whether the classifier based on decision rules with higher robustness scores performs better in the classification of objects with missing values.

The eight data-sets listed in Table 3 are used for these experiments. They are obtained from UCI Machine Learning Repository [10]. In Table 3, $|U|$, $|C|$ and $|V_d|$ show the number of objects, the number of condition attributes and the number of decision classes, respectively.

We run the 10 fold cross validation 10 times. At each validation stage, we first induce a set of rules by MLEM2 algorithm [9] from training data, which is a minimal set of minimal rules with 100% confidence. Let N_{rule} be the number of induced rules by MLEM2 algorithm and $\beta \in (0, 1)$. Then we select βN_{rule} rules from those induced rules according to robustness measure $Ex\text{-}rbst(E \to H)$. Namely, we obtain two subsets of induced rules: one called *Top* is composed of βN_{rule} rules with top βN_{rule} robustness scores and the other called *Bottom* is composed of βN_{rule} rules with bottom βN_{rule} robustness scores. For comparison, we prepare the third subset *Random* of induced rules composed of randomly chosen βN_{rule} rules. The performances of those three subsets of induced rules in checking data are compared. Here βN_{rule} is assumed to be rounded to the nearest integer. To define the expectation-based robustness measure, assuming that we

Table 4. The average robustness scores of sampled rules (continue)

β	car (Tr: $\bar{N}_{\text{rule}} = 57.12$)			car (Ch: $\bar{N}_{\text{match}} = 49.60$)		
	Top	$Bottom$	$Random$	Top'	$Bottom'$	$Random'$
50%	$0.842_{\pm 0.110}$	$0.605_{\pm 0.074}$	$0.724_{\pm 0.153}$	$0.833_{\pm 0.137}$	$0.639_{\pm 0.178}$	$0.739_{\pm 0.185}$
30%	$0.919_{\pm 0.069}$	$0.560_{\pm 0.057}$	$0.724_{\pm 0.151}$	$0.907_{\pm 0.105}$	$0.607_{\pm 0.187}$	$0.738_{\pm 0.191}$
10%	$0.955_{\pm 0.003}$	$0.503_{\pm 0.054}$	$0.724_{\pm 0.156}$	$0.952_{\pm 0.025}$	$0.585_{\pm 0.238}$	$0.737_{\pm 0.198}$
β	ecoli (Tr: $\bar{N}_{\text{rule}} = 34.38$)			ecoli(Ch: $\bar{N}_{\text{match}} = 21.34$)		
	Top	$Bottom$	$Random$	Top'	$Bottom'$	$Random'$
50%	$0.883_{\pm 0.043}$	$0.752_{\pm 0.069}$	$0.817_{\pm 0.089}$	$0.790_{\pm 0.227}$	$0.462_{\pm 0.367}$	$0.625_{\pm 0.349}$
30%	$0.909_{\pm 0.032}$	$0.719_{\pm 0.067}$	$0.817_{\pm 0.086}$	$0.841_{\pm 0.179}$	$0.401_{\pm 0.370}$	$0.635_{\pm 0.331}$
10%	$0.939_{\pm 0.016}$	$0.663_{\pm 0.068}$	$0.821_{\pm 0.083}$	$0.907_{\pm 0.105}$	$0.291_{\pm 0.360}$	$0.629_{\pm 0.342}$
β	glass (Tr: $\bar{N}_{\text{rule}} = 21.96$)			glass (Ch: $\bar{N}_{\text{match}} = 15.13$)		
	Top	$Bottom$	$Random$	Top'	$Bottom'$	$Random'$
50%	$0.888_{\pm 0.026}$	$0.798_{\pm 0.044}$	$0.840_{\pm 0.058}$	$0.668_{\pm 0.325}$	$0.514_{\pm 0.376}$	$0.601_{\pm 0.356}$
30%	$0.902_{\pm 0.018}$	$0.776_{\pm 0.038}$	$0.843_{\pm 0.058}$	$0.687_{\pm 0.318}$	$0.483_{\pm 0.385}$	$0.588_{\pm 0.364}$
10%	$0.917_{\pm 0.012}$	$0.745_{\pm 0.035}$	$0.834_{\pm 0.061}$	$0.725_{\pm 0.293}$	$0.424_{\pm 0.391}$	$0.544_{\pm 0.372}$
β	iris (Tr: $\bar{N}_{\text{rule}} = 7.6$)			iris (Ch: $\bar{N}_{\text{match}} = 5.10$)		
	Top	$Bottom$	$Random$	Top'	$Bottom'$	$Random'$
50%	$0.912_{\pm 0.049}$	$0.763_{\pm 0.127}$	$0.841_{\pm 0.124}$	$0.866_{\pm 0.164}$	$0.620_{\pm 0.333}$	$0.755_{\pm 0.289}$
30%	$0.930_{\pm 0.042}$	$0.735_{\pm 0.135}$	$0.830_{\pm 0.125}$	$0.889_{\pm 0.141}$	$0.574_{\pm 0.349}$	$0.727_{\pm 0.310}$
10%	$0.965_{\pm 0.024}$	$0.666_{\pm 0.153}$	$0.832_{\pm 0.132}$	$0.952_{\pm 0.087}$	$0.468_{\pm 0.380}$	$0.705_{\pm 0.328}$

know one of condition attribute is missing but we do not know which condition attribute is missing, we use the following probability distribution: $P(L) = 1/|C|$ if $|L| = 1$ and $P(L) = 0$ otherwise.

4.2 The Usefulness as Design Knowledge

To confirm the usefulness of the decision rules with high robustness scores as the design knowledge, we calculate the average robustness scores in the checking data for subsets of induced rules. In this calculation, we use rule sets Top', $Bottom'$ and $Random'$ instead of rule sets Top, $Bottom$ and $Random$, respectively. We define rule sets Top', $Bottom'$ and $Random'$ by replacing N_{rule} in the definitions of rule sets Top, $Bottom$ and $Random$ with N_{match} showing the number of rules whose conditions are satisfied by at least one of checking data. This replacement is caused by the fact that robustness scores cannot be obtained properly for unfired rules. The results for $\beta = 50\%$, 30% and 10% are shown in the right half of Tables 4 and 5. The average robustness scores in the training data are also shown in the left half of those tables. $\bar{N}_{\text{rule}}$ and $\bar{N}_{\text{match}}$ are the average values of N_{rule} and N_{match}, respectively.

As shown in those tables, the average values are gradually decreases (resp. increases) as β increases in Top' (resp. $Bottom'$). This implies that the order of robustness scores in the training data is mostly preserved in the checking data. Moreover, the average robustness scores of Top' are larger than those of $Bottom'$ and $Random'$ when β is sufficiently small. From those results, we conclude that

Table 5. The average robustness scores of sampled rules (continuation)

β	nursery (Tr: $N_{\text{rule}} = 533.43$)			nursey (Ch: $N_{\text{match}} = 389.46$)		
	Top	$Bottom$	$Random$	Top'	$Bottom'$	$Random'$
50%	$0.877_{\pm 0.022}$	$0.760_{\pm 0.093}$	$0.818_{\pm 0.090}$	$0.875_{\pm 0.044}$	$0.773_{\pm 0.197}$	$0.826_{\pm 0.152}$
30%	$0.890_{\pm 0.018}$	$0.712_{\pm 0.094}$	$0.818_{\pm 0.090}$	$0.887_{\pm 0.038}$	$0.731_{\pm 0.238}$	$0.821_{\pm 0.159}$
10%	$0.909_{\pm 0.015}$	$0.599_{\pm 0.066}$	$0.818_{\pm 0.090}$	$0.906_{\pm 0.030}$	$0.602_{\pm 0.329}$	$0.832_{\pm 0.152}$
β	soybean (Tr: $\bar{N}_{\text{rule}} = 55.74$)			soybean (Ch: $\bar{N}_{\text{match}} = 33.33$)		
	Top	$Bottom$	$Random$	Top'	$Bottom'$	$Random'$
50%	$0.969_{\pm 0.007}$	$0.935_{\pm 0.020}$	$0.952_{\pm 0.022}$	$0.928_{\pm 0.157}$	$0.601_{\pm 0.424}$	$0.761_{\pm 0.362}$
30%	$0.973_{\pm 0.003}$	$0.924_{\pm 0.017}$	$0.951_{\pm 0.023}$	$0.949_{\pm 0.112}$	$0.494_{\pm 0.435}$	$0.763_{\pm 0.361}$
10%	$0.976_{\pm 0.003}$	$0.907_{\pm 0.014}$	$0.952_{\pm 0.022}$	$0.959_{\pm 0.086}$	$0.426_{\pm 0.439}$	$0.768_{\pm 0.361}$
β	wine (Tr: $\bar{N}_{\text{rule}} = 4.58$)			wine (Ch: $\bar{N}_{\text{match}} = 4.26$)		
	Top	$Bottom$	$Random$	Top'	$Bottom'$	$Random'$
50%	$0.963_{\pm 0.010}$	$0.940_{\pm 0.018}$	$0.951_{\pm 0.018}$	$0.893_{\pm 0.135}$	$0.871_{\pm 0.158}$	$0.887_{\pm 0.147}$
30%	$0.965_{\pm 0.010}$	$0.939_{\pm 0.019}$	$0.951_{\pm 0.020}$	$0.891_{\pm 0.129}$	$0.866_{\pm 0.156}$	$0.886_{\pm 0.123}$
10%	$0.972_{\pm 0.006}$	$0.933_{\pm 0.025}$	$0.951_{\pm 0.023}$	$0.911_{\pm 0.113}$	$0.859_{\pm 0.183}$	$0.884_{\pm 0.155}$
β	zoo (Tr: $\bar{N}_{\text{rule}} = 9.63$)			zoo (Ch: $\bar{N}_{\text{match}} = 5.04$)		
	Top	$Bottom$	$Random$	Top'	$Bottom'$	$Random'$
50%	$0.949_{\pm 0.013}$	$0.898_{\pm 0.051}$	$0.924_{\pm 0.046}$	$0.947_{\pm 0.070}$	$0.863_{\pm 0.254}$	$0.899_{\pm 0.206}$
30%	$0.955_{\pm 0.009}$	$0.885_{\pm 0.054}$	$0.922_{\pm 0.048}$	$0.953_{\pm 0.047}$	$0.834_{\pm 0.287}$	$0.894_{\pm 0.217}$
10%	$0.963_{\pm 0.001}$	$0.861_{\pm 0.058}$	$0.933_{\pm 0.038}$	$0.963_{\pm 0.009}$	$0.783_{\pm 0.335}$	$0.930_{\pm 0.135}$

the rules with high robustness scores in the training data can take high robustness scores in a set of unseen objects. Thus, the rules with high robustness scores in the training data could be useful as design knowledge.

4.3 The Usefulness in Classification of Unseen Objects

Unseen objects will not match to the rules totally but partially. From this point of view, rules with high robustness can contribute well to the classification of unseen objects. To examine this conjecture, we compare classification accuracies of classifiers based on Top, $Bottom$ and $Random$ by 10 times run of 10 fold cross validation.

For the classification of unseen objects, we apply the classification system of LERS [9] described below. Let $\mathcal{R}$ be a subset of induced decision rules and u an object to be classified. We define $R(D_i)$ as a set of all decision rules in $\mathcal{R}$ inferring the belongingness to a class D_i, $Mat(u, \mathcal{R})$ as a set of decision rules in $\mathcal{R}$ whose conditions are satisfied with object u, and $PM(u, \mathcal{R})$ as a set of decision rules in $\mathcal{R}$ some of whose conditions for condition attributes are satisfied with object u. When conditions of decision rules in $\mathcal{R}$ are satisfied with object u, i.e., $Mat(u, \mathcal{R}) \neq \emptyset$, the following measure $Supp(D_i)$ is calculated:

$$Supp(D_i) = \sum_{r \in R(D_i) \cap Mat(u, \mathcal{R})} Strength(r) \times Specificity(r), \tag{14}$$

where r is a decision rule, $Strength(r)$ is the total number of objects in given decision table correctly classified by rule r and $Specificity(r)$ is the total

Table 6. The result in data without missing values

β	car			ecoli		
	Top	*Bottom*	*Random*	*Top*	*Bottom*	*Random*
90	$0.974_{\pm 0.012}^{**}$	$0.961_{\pm 0.050}$	$0.965_{\pm 0.032}$	$0.782_{\pm 0.067}^{**}$	$0.675_{\pm 0.126}$	$0.763_{\pm 0.076}$
80	$0.956_{\pm 0.015}^{**}$	$0.440_{\pm 0.077}$	$0.937_{\pm 0.047}$	$0.779_{\pm 0.065}^{**}$	$0.438_{\pm 0.117}$	$0.738_{\pm 0.080}$
70	$0.932_{\pm 0.020}^{**}$	$0.292_{\pm 0.033}$	$0.907_{\pm 0.057}$	$0.771_{\pm 0.068}^{**}$	$0.378_{\pm 0.084}$	$0.712_{\pm 0.090}$
60	$0.901_{\pm 0.023}^{**}$	$0.292_{\pm 0.033}$	$0.860_{\pm 0.095}$	$0.753_{\pm 0.074}^{**}$	$0.323_{\pm 0.085}$	$0.672_{\pm 0.104}$
50	$0.863_{\pm 0.027}^{**}$	$0.261_{\pm 0.046}$	$0.833_{\pm 0.089}$	$0.733_{\pm 0.078}^{**}$	$0.254_{\pm 0.086}$	$0.640_{\pm 0.106}$
β	glass			iris		
	Top	*Bottom*	*Random*	*Top*	*Bottom*	*Random*
90	$0.668_{\pm 0.109}^{**}$	$0.635_{\pm 0.122}$	$0.648_{\pm 0.123}$	$0.930_{\pm 0.066}^{**}$	$0.930_{\pm 0.066}$	$0.930_{\pm 0.066}$
80	$0.644_{\pm 0.115}^{**}$	$0.597_{\pm 0.111}$	$0.621_{\pm 0.131}$	$0.930_{\pm 0.066}^{**}$	$0.825_{\pm 0.140}$	$0.848_{\pm 0.151}$
70	$0.614_{\pm 0.121}^{**}$	$0.564_{\pm 0.114}$	$0.588_{\pm 0.111}$	$0.929_{\pm 0.066}^{**}$	$0.715_{\pm 0.124}$	$0.798_{\pm 0.181}$
60	$0.592_{\pm 0.119}^{**}$	$0.494_{\pm 0.110}$	$0.560_{\pm 0.130}$	$0.925_{\pm 0.074}^{**}$	$0.603_{\pm 0.185}$	$0.705_{\pm 0.197}$
50	$0.569_{\pm 0.128}^{**}$	$0.426_{\pm 0.129}$	$0.514_{\pm 0.127}$	$0.867_{\pm 0.159}^{**}$	$0.409_{\pm 0.192}$	$0.619_{\pm 0.217}$
β	nursery			soybean		
	Top	*Bottom*	*Random*	*Top*	*Bottom*	*Random*
90	$0.977_{\pm 0.004}^{**}$	$0.642_{\pm 0.014}$	$0.948_{\pm 0.082}$	$0.871_{\pm 0.043}^{**}$	$0.700_{\pm 0.067}$	$0.815_{\pm 0.065}$
80	$0.965_{\pm 0.005}^{**}$	$0.508_{\pm 0.015}$	$0.886_{\pm 0.128}$	$0.869_{\pm 0.042}^{**}$	$0.536_{\pm 0.078}$	$0.745_{\pm 0.085}$
70	$0.946_{\pm 0.006}^{**}$	$0.435_{\pm 0.015}$	$0.820_{\pm 0.158}$	$0.862_{\pm 0.042}^{**}$	$0.467_{\pm 0.075}$	$0.684_{\pm 0.089}$
60	$0.924_{\pm 0.011}^{**}$	$0.398_{\pm 0.014}$	$0.760_{\pm 0.163}$	$0.857_{\pm 0.044}^{**}$	$0.338_{\pm 0.078}$	$0.618_{\pm 0.094}$
50	$0.879_{\pm 0.011}^{**}$	$0.386_{\pm 0.013}$	$0.740_{\pm 0.164}$	$0.833_{\pm 0.048}^{**}$	$0.254_{\pm 0.062}$	$0.532_{\pm 0.102}$
β	wine			zoo		
	Top	*Bottom*	*Random*	*Top*	*Bottom*	*Random*
90	$0.925_{\pm 0.064}$	$0.925_{\pm 0.064}$	$0.925_{\pm 0.064}$	$0.960_{\pm 0.068}^{**}$	$0.676_{\pm 0.191}$	$0.906_{\pm 0.119}$
80	$0.780_{\pm 0.172}^{**}$	$0.858_{\pm 0.106}$	$0.842_{\pm 0.140}$	$0.918_{\pm 0.079}^{**}$	$0.422_{\pm 0.157}$	$0.810_{\pm 0.166}$
70	$0.700_{\pm 0.147}^{**}$	$0.812_{\pm 0.123}$	$0.783_{\pm 0.148}$	$0.908_{\pm 0.089}^{**}$	$0.317_{\pm 0.137}$	$0.692_{\pm 0.205}$
60	$0.563_{\pm 0.221}^{**}$	$0.739_{\pm 0.152}$	$0.694_{\pm 0.157}$	$0.893_{\pm 0.098}^{**}$	$0.243_{\pm 0.132}$	$0.561_{\pm 0.233}$
50	$0.455_{\pm 0.167}^{**}$	$0.628_{\pm 0.147}$	$0.583_{\pm 0.144}$	$0.854_{\pm 0.130}^{**}$	$0.156_{\pm 0.125}$	$0.483_{\pm 0.235}$

number of condition attributes in the condition of rule r. For convenience, when $Mat(u, \mathcal{R}) = \emptyset$, we define $Supp(D_i) = 0$. When no conditions of rules in $\mathcal{R}$ are satisfied with object u, i.e., when $Mat(u, \mathcal{R}) = \emptyset$, the following measure $M(D_i)$ is calculated:

$$M(D_i) = \sum_{r \in R(D_i) \cap PM(u,\mathcal{R})} Match\_factor(r) \times Strength(r) \times Specificity(r), \tag{15}$$

where $Match\_factor(r)$ is the ratio of the number of matched conditions for condition attributes of rule r to the total number of condition attributes used in rule r. The classification is performed as follows: if there exists D_j such that $Supp(D_j) > 0$, the class D_i with the largest $Supp(D_i)$ is selected. Otherwise, the class D_i with the largest $M(D_i)$ is selected.

The results of this experiment for $\beta = 50\%$, 60%, 70%, 80% and 90% are shown in Table 6. In Table 6, marks $*$ and $**$ mean the average classification accuracy of top βR rules is significantly different from that of random βR rules by the paired t-test with significance level $\alpha = 0.05$ and 0.01. As shown in

Table 7. The results in data with missing values

β	car			ecoli		
	Top	*Bottom*	*Random*	*Top*	*Bottom*	*Random*
90%	$0.796_{\pm 0.020}^{**}$	$0.785_{\pm 0.036}$	$0.788_{\pm 0.027}$	$0.684_{\pm 0.065}^{**}$	$0.595_{\pm 0.114}$	$0.677_{\pm 0.064}$
80%	$0.797_{\pm 0.021}^{**}$	$0.331_{\pm 0.050}$	$0.782_{\pm 0.029}$	$0.681_{\pm 0.064}^{**}$	$0.373_{\pm 0.098}$	$0.652_{\pm 0.077}$
70%	$0.794_{\pm 0.023}^{**}$	$0.224_{\pm 0.027}$	$0.759_{\pm 0.039}$	$0.674_{\pm 0.065}^{**}$	$0.323_{\pm 0.073}$	$0.627_{\pm 0.098}$
60%	$0.788_{\pm 0.024}^{**}$	$0.224_{\pm 0.027}$	$0.741_{\pm 0.056}$	$0.661_{\pm 0.067}^{**}$	$0.284_{\pm 0.072}$	$0.611_{\pm 0.089}$
50%	$0.773_{\pm 0.026}^{**}$	$0.202_{\pm 0.034}$	$0.729_{\pm 0.072}$	$0.646_{\pm 0.072}^{**}$	$0.226_{\pm 0.074}$	$0.554_{\pm 0.122}$
β	glass			iris		
	Top	*Bottom*	*Random*	*Top*	*Bottom*	*Random*
90%	$0.582_{\pm 0.082}^{*}$	$0.550_{\pm 0.118}$	$0.560_{\pm 0.088}$	$0.764_{\pm 0.066}$	$0.764_{\pm 0.139}$	$0.764_{\pm 0.066}$
80%	$0.565_{\pm 0.089}^{**}$	$0.525_{\pm 0.107}$	$0.536_{\pm 0.091}$	$0.765_{\pm 0.066}^{**}$	$0.708_{\pm 0.152}$	$0.702_{\pm 0.112}$
70%	$0.547_{\pm 0.096}^{**}$	$0.498_{\pm 0.103}$	$0.517_{\pm 0.097}$	$0.758_{\pm 0.068}^{**}$	$0.625_{\pm 0.127}$	$0.653_{\pm 0.129}$
60%	$0.531_{\pm 0.098}^{**}$	$0.442_{\pm 0.110}$	$0.491_{\pm 0.098}$	$0.752_{\pm 0.075}^{**}$	$0.553_{\pm 0.130}$	$0.586_{\pm 0.153}$
50%	$0.512_{\pm 0.106}^{**}$	$0.384_{\pm 0.106}$	$0.452_{\pm 0.102}$	$0.707_{\pm 0.118}^{**}$	$0.398_{\pm 0.118}$	$0.568_{\pm 0.155}$
β	nursery			soybean		
	Top	*Bottom*	*Random*	*Top*	*Bottom*	*Random*
90%	$0.816_{\pm 0.006}^{**}$	$0.542_{\pm 0.011}$	$0.779_{\pm 0.084}$	$0.834_{\pm 0.038}^{**}$	$0.667_{\pm 0.062}$	$0.778_{\pm 0.063}$
80%	$0.813_{\pm 0.007}^{**}$	$0.436_{\pm 0.014}$	$0.737_{\pm 0.116}$	$0.831_{\pm 0.039}^{**}$	$0.505_{\pm 0.073}$	$0.723_{\pm 0.075}$
70%	$0.800_{\pm 0.008}^{**}$	$0.375_{\pm 0.012}$	$0.694_{\pm 0.138}$	$0.823_{\pm 0.039}^{**}$	$0.438_{\pm 0.069}$	$0.655_{\pm 0.084}$
60%	$0.783_{\pm 0.011}^{**}$	$0.356_{\pm 0.013}$	$0.619_{\pm 0.148}$	$0.817_{\pm 0.041}^{**}$	$0.321_{\pm 0.072}$	$0.595_{\pm 0.095}$
50%	$0.747_{\pm 0.011}^{**}$	$0.353_{\pm 0.012}$	$0.607_{\pm 0.153}$	$0.794_{\pm 0.045}^{**}$	$0.246_{\pm 0.059}$	$0.521_{\pm 0.086}$
β	wine			zoo		
	Top	*Bottom*	*Random*	*Top*	*Bottom*	*Random*
90%	$0.784_{\pm 0.063}$	$0.784_{\pm 0.063}$	$0.784_{\pm 0.063}$	$0.866_{\pm 0.064}^{**}$	$0.610_{\pm 0.179}$	$0.823_{\pm 0.128}$
80%	$0.690_{\pm 0.141}^{**}$	$0.744_{\pm 0.093}$	$0.737_{\pm 0.118}$	$0.854_{\pm 0.072}^{**}$	$0.373_{\pm 0.141}$	$0.761_{\pm 0.157}$
70%	$0.644_{\pm 0.130}^{**}$	$0.715_{\pm 0.112}$	$0.667_{\pm 0.117}$	$0.847_{\pm 0.082}^{**}$	$0.277_{\pm 0.121}$	$0.641_{\pm 0.244}$
60%	$0.537_{\pm 0.198}^{**}$	$0.651_{\pm 0.139}$	$0.656_{\pm 0.142}$	$0.835_{\pm 0.090}^{**}$	$0.212_{\pm 0.116}$	$0.559_{\pm 0.206}$
50%	$0.446_{\pm 0.155}^{**}$	$0.550_{\pm 0.125}$	$0.579_{\pm 0.134}$	$0.803_{\pm 0.118}^{**}$	$0.135_{\pm 0.110}$	$0.437_{\pm 0.233}$

Table 6, *Top* is better than *Random* and *Bottom* except for data-set "wine". Then we observe that the classifier based on rules with high robustness scores performs well in the classification of unseen objects. In data-set "wine", rules with high recall scores take low robustness scores and the variation of the robustness scores is rather small as shown in Table 5. The 100% confident rules with high recall scores often perform well in the classification. This explains why *Bottom* becomes better than *Top* in data-set "wine".

4.4 The Usefulness in Classification of Objects with Missing Values

In the previous experiments, we observed that rules with high robustness scores in the training data take high scores in a set of unseen objects and that the classifier based on such rules performs well in the classification of unseen objects. From those observations, we guess that the classifier based on rules with high robustness scores can stably perform well in the classification against loss of attribute data.

To confirm this by a numerical experiment, we delete all data of a condition attribute from the checking data at validation stage of 10 fold cross validation. There are $|C|$ condition attributes. The deletion is applied to each condition attribute so that we obtain $|C|Ch$ checking data with missing values, where Ch is the number of checking data. Then the average classification accuracy is calculated at each validation stage.

The results of this experiment for $\beta = 50\%$, 60%, 70%, 80% and 90% are shown in Table 7. In Table 7, marks $*$ and $**$ mean the average classification accuracy of top βR rules is significantly different from that of random βR rules by the paired t-test with significance level $\alpha = 0.05$ and 0.01. As shown in Table 7, *Top* is better than *Random* and *Bottom* except for data-set "wine". We observe that the classifier based on rule with high robustness scores maintains the classification accuracy against the missing value of a condition attribute. The reason why similar result cannot be obtained for "wine" data is the same as the previous experiment.

By the experiments, we observed the usefulness of a robustness measure. Because the robustness measure reflects a different aspect from the recall, we will investigate the relation between the robustness measure and the recall in our future work. Moreover, we will study the application of robustness measures to rule induction algorithms.

Ackowledgement. The authors express their gratitude to Mr. Eiji Sekiya (Osaka University) for his great help in the experiments.

References

1. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
2. Mori, N., Tanaka, H., Inoue, K. (eds.): Rough Sets and Kansei. Kaibundo, Tokyo (2006) (in Japanese)
3. Geng, L., Hamilton, H.J.: Interestingness Measures for Data Mining: A Survey. ACM Computing Surveys 38(9), 1–32 (2006)
4. Greco, S., Słowiński, R., Szczech, I.: Properties of Rule Interestingness Measures and Alternative Approaches to Normalization of Measures. Information Sciences 216, 1–16 (2012)
5. Lenca, P., Meyer, P., Vaillant, B., Lallich, S.: On Selecting Interestingness Measures for Association Rules: User Oriented Description and Multiple Criteria Decision Aid. European Journal of Operation Research 184, 610–626 (2008)
6. Mcgarry, K.: A Survey of Interestingness Measure for Knowledge Discovery. The Knowledge Engineering Review, 1–24 (2005)
7. Lenca, P., Vaillant, B., Lallich, S.: On the Robustness of Association Rules. In: Proceedings of 2006 IEEE Conference on Cybernetics and Intelligent Systems, pp. 596–601 (2006)
8. Le Bras, Y., Meyer, P., Lenca, P., Lallich, S.: A Robustness Measure of Association Rules. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part II. LNCS, vol. 6322, pp. 227–242. Springer, Heidelberg (2010)
9. Grzymala-Busse, J.W.: MLEM2 - Discretization During Rule Induction. In: Proceedings of the IIPWM 2003, pp. 499–508 (2003)
10. UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/`

Exploring Margin for Dynamic Ensemble Selection

Leijun Li, Qinghua Hu, Xiangqian Wu, and Daren Yu

Biometric Computing Research Centre, School of Computer Science and Technology,
Harbin Institute of Technology, Harbin 150001, P.R. China
lileijun1985@163.com,{huqinghua,xqwu,yudaren}@hit.edu.cn

Abstract. How to effectively combine the outputs of base classifiers is one of the key issues in ensemble learning. A new dynamic ensemble selection algorithm is proposed in this paper. In order to predict a sample, the base classifiers whose classification confidences on this sample are greater than or equal to specified threshold value are selected. Since margin is an important factor to the generalization performance of voting classifiers, thus the threshold value is estimated via the minimization of margin loss. We analyze the proposed algorithm in detail and compare it with some other multiple classifiers fusion algorithms. The experimental results validate the effectiveness of our algorithm.

Keywords: dynamic ensemble selection, threshold value, classification confidence, margin.

1 Introduction

Ensemble learning is an effective method to develop accurate classification systems [1,23]. Typically, there are two steps to construct an ensemble system: learning a set of base classifiers and combining them with a certain strategy. For learning strategies, the key is to obtain both diverse and accurate base classifiers. So far various algorithms have been invented and they can be roughly categorized into two schemes. One is to learn the base classifiers in parallel, such as, Bagging [1], Rotation Forest [13]. The other is sequential, that is, the base classifiers are trained one by one, including AdaBoost [4], LogitBoost [5] and so on.

As to fusion strategies, there are also two main schemes. One uses the fixed set of base classifiers which can be all the base classifiers or only a subset of them. It requires large memory store and takes much computation time for prediction when using all the base classifiers [10]. In order to alleviate these drawbacks, ensemble pruning was proposed and it selects a fraction of base classifiers for fusion. If there are L base classifiers, we have 2^L-1 nonempty sub-ensembles. Therefore, it is intractable to search the optimal solution via exhaustive search for a moderate ensemble size. In order to alleviate this difficulty, several strategies have been utilized in ensemble pruning to obtain sub-optimal subset, such as, genetic algorithm [22], ordered aggregation technique [10,11] and so on. It

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 178–187, 2013.

has been demonstrated that ensemble pruning is more effective in terms of classification performance than using all the base classifiers.

However there is a drawback with ensemble pruning. That is, the selected base classifiers on the validation set may not be well adapted for predicting the test set due to the differences among samples. Intuitively, the classification performance of adopting different base classifiers for different samples may be better than that of using the fixed base classifiers. Thus the dynamic scheme was proposed, including dynamic classifier selection [6,8,18] and dynamic ensemble selection [9,15,21]. For dynamic classifier selection, it selects one base classifier to predict a sample every time and the selected classifier for the sample is thought to most likely classify it correctly. These algorithms include dynamic classifier selection based on classifier's local accuracy [18], dynamic classifier selection based on multiple classifier behaviour [8] and so on. Since only one base classifier is selected, thus if the selected base classifier is not able to classify the sample correctly, there is no way to avoid the misclassification [16].

In order to overcome this drawback, dynamic ensemble selection was introduced. Rather than selecting a single base classifier, it selects one subset of base classifiers to predict a sample every time. In [15], DCS-based DCES method was proposed. It considers both accuracy and diversity and contains two versions: cluster and select version, and K-NN and selection version. Then in [9], dynamic classifier ensemble selection by K-nearest-oracles was proposed. Given a test sample, it selects a subset of base classifiers which can correctly classify those K neighbors on the validation set. Recently, GDES-AD was proposed and it is robust to noise [21]. In this paper, a new dynamic ensemble selection algorithm is proposed. In order to predict a sample, the base classifiers whose classification confidences on this sample are greater than or equal to specified threshold value are selected and the threshold value is estimated via the minimization of margin loss.

The rest of the paper is organized as follows. The new algorithm is introduced in Section 2. Then we analyze it in detail and compare it with other fusion methods on some UCI classification tasks in Section 3. Finally, Section 4 offers the conclusions and future work.

2 The Proposed Algorithm

In the framework of dynamic ensemble selection, we are generally given a set of base classifiers $\{h_1, \cdots, h_L\}$ and the main goal is to dynamically select a subset for prediction. Intuitively, the higher the classification confidence provided by the classifier, the higher the probability that the classifier has correctly classified this sample. Thus, for the proposed method, a sample is classified by a subset of base classifiers whose classification confidences on this sample are greater than or equal to specified threshold value. Naturally, how to estimate an appropriate threshold value becomes the key issue.

Margin is an important factor to the generalization performance of voting classifier [14,20]. It has been reported that if the voting classifier can generate

good margin distribution, then its generalization error will be small. Motivated by this observation, the threshold value is estimated via the minimization of margin loss in this paper.

Classification confidence is utilized in the new algorithm, thus every classifier h_j assigns classification confidence r_{ij} for its classification decision h_{ij} on sample x_i. For example, consider a linear real valued classifier $h(x) = \mathbf{u} \cdot x - b$, a sample x is given classification decision 1 if $h(x) \geq 0$ and -1 otherwise. Then the value $|h(x)|$ can be seen as its classification confidence. In [17] the bound on generalization errors of $h(x)$ is given and it shows that classification confidence is an important factor to its classification performance. On the other hand, in [14], the margin of sample $x_i \in X$ is defined as the difference between the number of correct votes and the maximum number of votes received by any wrong label. It only considers the classification decision. Inspired by the conclusion in [17], classification confidence is added into margin as follows.

Definition 1. *For $x_i \in X(i = 1, 2 \cdots, n)$, let $\omega = \{\omega_1, \cdots, \omega_c\}$ be class labels set, $H = \{h_{ij} | h_{ij} \in \omega\}$ and $R = \{r_{ij} | r_{ij} \in [0, 1]\}$ be classification decision and classification confidence of x_i by the classifier $h_j(j = 1, 2 \cdots, L)$, respectively. The margin of sample x_i based on classification confidence is denoted by*

$$M(x_i) = S(\omega_i) - max\{S(\omega_j) | i \neq j\} \tag{1}$$

where $S(\omega_i)$ means the sum of classification confidences in R whose corresponding classification decision is ω_i which is the true label of x_i.

The detail process of the proposed method is given as Algorithm 1. Here, for sample x_i, least squares loss function and logistic loss function are respectively utilized to compute its margin loss.

Definition 2. *For $x_i \in X$, the margin loss of x_i based on two different loss functions are respectively denoted as*

$$l_1(x_i) = [1 - M(x_i)]^2 \tag{2}$$

$$l_2(x_i) = \log(1 + \exp(-M(x_i))) \tag{3}$$

Algorithm 1. (DES-Margin)
Input:
•$X = \{(x_i, y_i), i = 1, 2, \cdots, n\}$: the validation set;
•x: the test sample;
•$h_j(j = 1, 2, \cdots, L)$: the base classifiers
Output: the label of x;
1. Apply $h_j(j = 1, 2, \cdots, L)$ on $x_i \in X(i = 1, 2, \cdots, n)$ to get classification decision h_{ij} and corresponding classification confidence r_{ij}
2. Compute the difference between the maximum classification confidence and the minimum classification confidence on sample x_i and denote it by $d(i) = \max\{r_{ij} | j = 1, 2, \cdots, L\} - \min\{r_{ij} | j = 1, 2, \cdots, L\}$

3. For $t = 1, 2, \cdots, L + 1$

4. The base classifiers whose classification confidences on x_i are greater than or equal to $\min\{r_{ij}|j = 1, 2, \cdots, L\}+(t-1)*d(i)/L$ are selected to compute its margin M_{it} and corresponding margin loss l_{it} as Definitions 1 and 2.

5. The sum of margin loss $l_{it}(i = 1, \cdots, n)$ on X is denoted by $S(t)$

6.End for

7. Estimate T as $S(T) = \min\{S(t)|t = 1, 2, \cdots, L + 1\}$

8. Apply $h_j(j = 1, 2, \cdots, L)$ on test sample x to get its classification decision h_{xj} and corresponding classification confidence r_{xj}

9. The base classifiers whose classification confidences on x are greater than or equal to $\min\{r_{xj}|j = 1, 2, \cdots, L\} + (T - 1) * d(x)/L$ are selected to classify x with weighted voting and the weight is corresponding classification confidence.

It should be noted that, since the classification confidences of base classifiers on different samples are usually different, the selected base classifiers subsets for different samples are usually different. Besides, for a test sample, the threshold value is determined by its minimum classification confidence, the value of T and the difference between its maximum classification confidence and its minimum classification confidence, thus the threshold values for different samples can also be different.

In what follows, the base classifiers learning strategy is given for the completeness of these experiments conducted in this paper. Here the nearest-neighbor algorithm is utilized to learn the base classifiers and the classification confidence of sample $x \in X$ is computed as $|f(x_2, x) - f(x_1, x)|/2$ where f is Euclidean distance function, x_1 is the nearest sample of x in X and x_2 is the nearest sample of x in X out of the class of x_1 [7]. If there are mixed numerical and categorical features, Heterogeneous Euclidean-Overlap Metric function can be introduced [19]. Besides, inspired by the idea of multimodal perturbation [23], bootstrap sampling and random feature selection are combined to perturb the training set.

3 Algorithm Analysis and Experimental Evaluation

From Algorithm 1, it can be seen that the base classifiers with larger classification confidence tend to be selected. Then whether larger classification confidence means better classification performance? Some experiments were conducted on UCI data sets [2] to answer this question. Table 1 describes the 15 data sets used in this work. In these experiments, the ratio of bootstrap sampling and random feature selection was set as 0.75 and the base classifiers number L was 100. The relationship between classification accuracy and classification confidence was shown as Figure 1. Specifically, on the x-axis, "1" means every test sample is classified by the classification decision with the minimum classification confidence and "100" means every test sample is classified by the classification decision with the maximal classification confidence.

From Figure 1, we can see the trend that the higher the classification confidence, the better the classification performance. It empirically interprets the reason of selecting the base classifiers with large classification confidence for fusion.

Table 1. Description of 15 data sets used in this study

Data set	Instances	Features	Classes
australian	690	14	2
bupa	345	6	2
crx	690	15	2
german	1000	20	2
hepatitis	155	19	2
liver	345	6	2
lymphography	148	18	4
movement	360	90	15
pima	768	8	2
rice	104	5	2
spectf	269	44	2
wdbc	569	30	2
wpbc	198	33	2
vehicl	846	18	4
yeast	1484	7	2

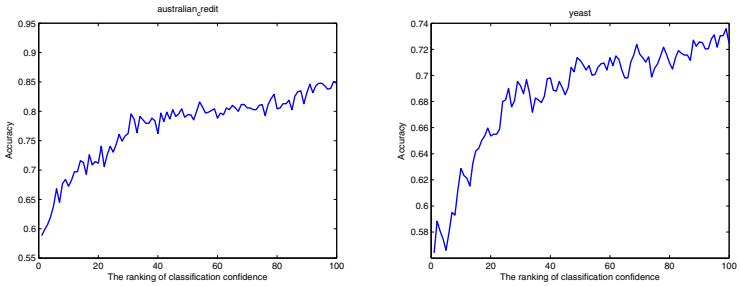


Fig. 1. Variation of classification accuracies with the ranking of classification confidence

Then whether the best classification performance can be obtained when only the base classifiers with the largest classification confidence are selected? In what follows, we explore this question and show the relationship between the classification accuracies and different threshold values as Figure 2. The experimental settings were given as the above experiment. The threshold values for sample x is computed as $\min\{r_{xj}|j = 1, 2, \cdots, 100\} + (t-1) * d(x)/100$ and $1 \leq t \leq 101$. Here different t correspond to different threshold values. On the x-axis, "1" means the threshold value for sample x is $\min\{r_{xj}|j = 1, 2, \cdots, L\} + (1-1) * d(x)/100 = \min\{r_{xj}|j = 1, 2, \cdots, L\}$ and all the base classifiers are selected to predict it with weighted voting; "101" means the threshold value for sample x is $\min\{r_{xj}|j = 1, 2, \cdots, L\} + (101-1) * d(x)/100 = \max\{r_{xj}|j = 1, 2, \cdots, L\}$ and only the base classifiers with the largest classification confidence on sample x are selected to classify this sample.

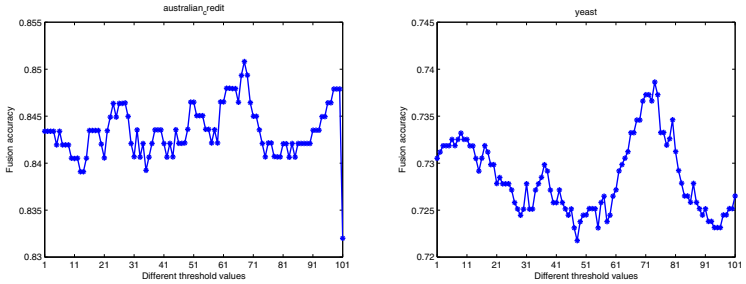


Fig. 2. Variation of classification accuracies with different threshold values

From Figure 2, it can be seen that selecting the base classifiers with the largest classification confidence does not mean the best classification performance and an appropriate threshold value is needed. The reason lies in the number of base classifiers with the largest classification confidence is small. In fact, a tradeoff between the threshold value and the number of selected base classifiers should be made. For DES-Margin, the threshold value is estimated based on the minimization of margin loss.

In Algorithm 1, least squares loss function and logistic loss function are respectively utilized to compute margin loss. Then which loss function can obtain better classification performance? In the experiments, each data set is split into 10 folds: 8 folds were used for training 100 base classifiers, 1 fold to estimate the threshold value and 1 fold for evaluating the classification performance. Table 2 shows the classification results of DES-Margin with the two loss functions. The bold accuracy is the the highest one. From Table 2, it can be seen that the classification performances of DES-Margin with the two loss functions are similar. Thus, in what follows, we only use least squares loss function in DES-Margin and compare it with other methods, including: the simple voting using all the classifiers, the single classifier, Reduce-Error Pruning (RE) [11], DCS-LA [18] and KNORA-UNION [9].

Reduce-Error Pruning is an ensemble pruning algorithm based on ordered aggregation technique. The base classifier with the lowest classification error on validation set is firstly selected into an empty initial sub-ensemble and then the remaining classifiers are sequentially added into the sub-ensemble to make classification error of the new sub-ensemble as low as possible. Finally the sub-ensemble with the best performance is selected as the pruned ensemble.

For DCS-LA, it is a dynamic classifier selection method based on local accuracy. In order to classify a test sample, the local accuracy of each classifier is estimated in a local region which is defined as K-nearest neighbors of the test sample. Then the classifier with the best classification performance in this local region is selected to classify the test sample.

As to dynamic ensemble selection algorithm KNORA-UNION, for each test sample, its K-nearest neighbors are estimated and the base classifiers which can correctly classify any of the K-nearest neighbors are selected to classify the test

Table 2. Classification performance of DES-Margin with different loss functions

Data set	least squares loss	logistic loss
australian	84.20 ±4.47	**84.78 ±4.27**
bupa	62.29±8.81	**63.44± 7.90**
crx	**83.03±15.49**	82.45 ±15.94
german	**73.50 ±3.14**	73.40± 4.43
hepatitis	85.50±9.10	**86.17± 8.89**
liver	**61.56± 9.00**	60.30± 7.22
lymphography	78.05±11.51	78.05±11.51
movement	**78.78± 18.99**	77.44± 18.97
pima	73.05±5.89	**73.82 ±5.06**
rice	82.18±9.90	82.18 ± 9.90
spectf	74.69±5.65	**75.44± 3.65**
vehicl	**72.56±2.78**	72.45± 3.71
wdbc	97.03±2.46	**97.37± 2.06**
wpbc	75.21±5.72	75.21± 5.72
yeast	**73.12 ±5.23**	72.31 ±5.52

Table 3. Classification performance of DES-Margin and other fusion methods

Data set	DES-Margin	SV	NN	RE	DCS-LA	KNORA-UNION
australian	84.20 ±4.47	82.74 ±3.00	78.85 ±4.69	83.05 ±3.40	80.29 ±4.28	83.04 ±3.38
bupa	62.29±8.81	59.95 ±9.46	60.24 ±6.24	60.85 ±9.38	62.55 ±12.97	58.50 ±11.24
crx	83.03±15.49	81.30 ±13.28	78.98 ±11.72	81.28 ±11.93	80.72 ±12.97	82.16 ±13.78
german	73.50 ±3.14	73.10 ±3.70	68.10 ±3.87	72.90 ±2.47	70.00 ±3.09	73.20 ±3.99
hepatitis	85.50±9.10	84.67 ±6.32	80.50 ±8.32	83.50 ±8.26	84.33 ±7.71	84.67 ±6.32
liver	61.56± 9.00	60.50 ±9.89	60.24 ±6.24	59.65 ±10.74	61.38 ±11.24	59.07 ±9.51
lymphography	78.05±11.51	77.34 ±11.20	70.91 ±12.12	71.36 ±10.70	70.19 ±11.83	76.62 ±10.83
movement	78.78± 18.99	77.89 ±18.88	77.67 ±18.46	77.89 ±18.88	76.56 ±19.63	77.89 ±18.88
pima	73.05±5.89	71.48 ±5.45	69.53 ±3.78	70.31 ±2.72	68.23 ±2.00	71.74 ±5.08
rice	82.18±9.90	78.25 ±9.30	76.23 ±10.40	83.87 ±10.73	86.80 ±7.70	78.25 ±9.30
spectf	74.69±5.65	72.37 ±6.94	70.93 ±8.48	70.89 ±12.09	74.28 ±7.47	72.37 ±6.94
vehicl	72.56±2.78	71.74 ±2.55	69.61 ±2.93	73.63 ±3.48	71.98 ±2.86	71.74 ±2.45
wdbc	97.03±2.46	96.15 ±2.83	95.09 ±3.05	96.13 ±2.96	94.56 ±2.54	96.15 ±2.83
wpbc	75.21±5.72	72.21 ±5.94	68.08 ±7.68	70.71 ±7.43	72.74 ±9.47	72.21 ±5.94
yeast	73.12 ±5.23	72.38 ±5.09	70.36 ±5.97	71.57 ±4.34	69.88 ±2.52	72.65 ±5.03

sample with simple voting. Here a base classifier can have more than one vote if it correctly classifies more than one neighbor, that is, the more neighbors a classifier classifies correctly, the more votes this classifier will have for the test sample.

Table 3 shows the classification results for each algorithm. A rank sum test called Nemenyi test [12] is performed to compare DES-Margin with other methods from the statistical viewpoint and the significance level α was set as 0.05. In Nemenyi test, the critical difference [3] for 6 algorithms and 15 data sets at significance level $\alpha = 0.05$ is

Table 4. Classification performance using different selection and voting strategies

Data set	SV	SDES-Margin	WV	DES-Margin
australian	82.74 ±3.00	82.46 ±2.94	84.34±4.45	84.20 ±4.47
bupa	59.95 ±9.46	62.58 ±7.11	61.71±9.11	62.29±8.81
crx	81.30 ±13.28	82.01 ±14.49	83.61±14.67	83.03±15.49
german	73.10 ±3.70	73.10 ±3.70	73.50±3.14	73.50 ±3.14
hepatitis	84.67 ±6.32	85.50 ±9.10	84.33±8.90	85.50±9.10
liver	60.50 ±9.89	59.93 ±11.06	60.55±9.12	61.56± 9.00
lymphography	77.34 ±11.20	77.34 ±11.20	78.77±11.28	78.05±11.51
movement	77.89 ±18.88	78.44 ±18.80	78.44±19.61	78.78± 18.99
pima	71.48 ±5.45	71.74 ±5.37	72.53±5.84	73.05±5.89
rice	78.25 ±9.30	81.18 ±9.53	78.25±9.30	82.18±9.90
spectf	72.37 ±6.94	73.54 ±7.70	74.28±5.43	74.69±5.65
vehicl	71.74 ±2.55	72.57 ±2.07	72.44±2.84	72.56±2.78
wdbc	96.15 ±2.83	96.85 ±2.43	96.67±2.24	97.03±2.46
wpbc	72.21 ±5.94	75.21 ±5.72	74.71±6.40	75.21±5.72
yeast	72.38 ±5.09	72.85 ±4.47	73.05±5.33	73.12 ±5.23

$$CD = q_{0.05}\sqrt{\frac{k(k+1)}{6N}} = 2.850 \times \sqrt{\frac{6 \times (6+1)}{6 \times 15}} = 1.947 \tag{4}$$

where $q_{0.05}$ is the critical values, k is the number of algorithms and N is the number of data sets.

The average ranks for DES-Margin, SV, NN, RE, DCS-LA and KNORA-UNION are (1.267, 3.333, 5.333, 3.733, 4.000, 3.333). The average rank differences between DES-Margin and the other methods are $(3.333 - 1.267 = 2.066 > 1.947, 5.333 - 1.267 = 4.066 > 1.947, 3.733 - 1.267 = 2.466 > 1.947, 4.000 - 1.267 = 2.733 > 1.947, 3.333 - 1.267 = 2.066 > 1.947)$, thus DES-Margin performs significantly better than SV, NN, RE, DCS-LA and KNORA-UNION. These experiments validate the effectiveness of DES-Margin.

Besides, it can be seen that there are mainly two parts in DES-Margin: dynamic ensemble selection for different test samples and weighted voting based on classification confidence. Then whether they are helpful for improving fusion performance? Here four solutions are considered: simple voting using all the classifiers (SV), simple voting based on dynamic ensemble selection (SDES-Margin), weighted voting using all the classifiers (WV) and weighted voting based on dynamic ensemble selection (DES-Margin). The experimental results in Table 4 indicate that dynamic ensemble selection and weighted voting based on classification confidence are necessary for improving classification performance.

4 Conclusions and Future Work

In this paper, a new dynamic ensemble selection algorithm DES-Margin is proposed. In order to predict a sample, the base classifiers whose classification confidences on this sample are greater than or equal to specified threshold value are

selected and the threshold value is estimated via the minimization of margin loss. This algorithm is analyzed systematically and the experimental results validate its effectiveness. Future works include, but are not limited to:

1) Exploring the internal relationship between generalization performance of voting classifier and margin based on classification confidence.

2) In this work, the threshold value is estimated via margin loss minimization. In the future, we will consider other estimation methods and apply them in dynamic ensemble selection.

Acknowledgments. This work is supported by National Natural Science Foundation of China under Grant 61170107,10978011, 60873140, 61073125, 61071179 and 11078010, National Science Fund for Distinguished Young Scholars under Grant 50925625 and the Program for New Century Excellent Talents in University (No. NCET-08-0155), the Research Fund for the Doctoral Program of Higher Education of China (No. 20101303110004), the Fundamental Research Funds for the Central Universities (No. HIT. NSRIF. 2013091) and the Fok Ying Tong Education Foundation (No. 122035).

References

1. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
2. Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine Learning Databases. Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, `http://archive.ics.uci.edu/ml/`
3. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139 (1997)
5. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: A Statistical View of Boosting. Annals of Statistics 28, 337–407 (2000)
6. Fagundes, D., Canuto, A.: Applying weights in the functioning of the dynamic classifier selection method. In: Proceedings of the Ninth Brazilian Symposium on Neural Networks, pp. 23–27 (2006)
7. Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin based feature selection-theory and algorithms. In: Proceedings of the Twenty-First International Conference on Machine Learning, pp. 43–50. ACM (2004)
8. Giacinto, G., Roli, F.: Dynamic classifier selection based on multiple classifier behaviour. Pattern Recognition 34, 1879–1881 (2001)
9. Ko, A.H.R., Sabourin, R., Britto Jr., A.S.B.: From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition 41, 1735–1748 (2008)
10. Martínez-Muñoz, G., Hernandez-Lobato, D., Suarez, A.: An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 245–259 (2009)
11. Margineantu, D.D., Dietterich, T.G.: Pruning Adaptive Boosting. In: Proceedings of the 14th International Conference on Machine Learning, pp. 211–218 (1997)
12. Nemenyi, P.B.: Distribution-free multiple comparisons. PhD thesis, Princeton University (1963)

13. Rodríguez, J.J., Kuncheva, L.I.: Rotation Forest: A New Classifier Ensemble Method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1619–1630 (2006)
14. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. Annals of Statistics 26, 1651–1686 (1998)
15. Santana, A., Soares, R.G.F., Canuto, A.M.P., Souto, M.C.P.: A dynamic classifier selection method to build ensembles using accuracy and diversity. In: Proceedings of the Ninth Brazilian Symposium on Neural Networks (SBRN), pp. 36–41 (2006)
16. Shin, H.W., Sohn, S.Y.: Selected tree classifier combination based on both accuracy and error diversity. Pattern Recognition 38, 191–197 (2005)
17. Shawe-Taylor, J., Cristianini, N.: Margin Distribution Bounds on Generalization. In: Fischer, P., Simon, H.U. (eds.) EuroCOLT 1999. LNCS (LNAI), vol. 1572, pp. 263–273. Springer, Heidelberg (1999)
18. Woods, K., Kegelmeyer, W.P., Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 405–410 (1997)
19. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)
20. Wang, L.W., Sugiyama, M., Jing, Z.X., Yang, C., Zhou, Z.H., Feng, J.F.: A Refined Margin Analysis for Boosting Algorithms via Equilibrium Margin. Journal of Machine Learning Research 12, 1835–1863 (2011)
21. Xiao, J., He, C.Z., Jiang, X.Y., Liu, D.H.: A dynamic classifier ensemble selection approach for noise data. Information Sciences 180, 3402–3421 (2010)
22. Zhou, Z.H., Wu, J.X., Tang, W.: Ensembling neural networks: many could be better than all. Artificial Intelligence 137, 239–263 (2002)
23. Zhou, Z.H., Yu, Y.: Ensembling Local Learners Through Multimodal Perturbation. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 35, 725–735 (2005)

Evaluation of Incremental Change of Set-Based Indices

Shusaku Tsumoto and Shoji Hirano

Department of Medical Informatics, School of Medicine, Faculty of Medicine
Shimane University
89-1 Enya-cho Izumo 693-8501 Japan
{tsumoto,hirano}@med.shimane-u.ac.jp

Abstract. This paper proposes a new framework for evaluation of set-based indices based on incremental sampling. Since these indices are defined by the relations between conditional attributes (R) and decision attribute(D), incremental sampling gives four possible cases according to the increment of sets for R or D. Using this classification, the behavior of indices can be evaluated for four cases. We applied this technique to several set-based indices. The results show that the evaluation framework gives a powerful tool for evaluation of set-based indices. Especially, it is found that the behavior of indices can be determined by a firstly given dataset..

Keywords: incremental rule induction, incremental sampling scheme, subrule layer, rule induction index, bayesian confirmation measure.

1 Introduction

There have been proposed several symbolic inductive learning methods, such as induction of decision trees [1–3], and AQ family [4, 5]. These methods are applied to discover meaningful knowledge from large databases, and their usefulness is in some aspects ensured. However, most of the approaches induces rules from all the data in databases, and cannot induce incrementally when new samples are derived. Thus, we have to apply rule induction methods again to the databases when such new samples are given, which causes the computational complexity to be expensive even if the complexity is n^2.

Thus, it is important to develop incremental learning systems in order to manage large databases [6, 7]. However, most of the previously introduced learning systems have the following two problems: first, those systems do not outperform ordinary learning systems, such as AQ15 [5], C4.5 [3] and CN2 [4]. Secondly, those incremental learning systems mainly induce deterministic rules. Therefore, it is indispensable to develop incremental learning systems which induce probabilistic rules to solve the above two problems.

Tsumoto and Hirano proposed a incremental rule induction method by using a new framework of incremental sampling [8]. Since accuracy and coverage [9] are defined by the relations between conditional attributes (R) and decision

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 188–199, 2013.

attribute(D), incremental sampling gives four possible cases according to the update of accuracy coverage as shown in Table 1. Using this classification, the behavior of these indices can be evaluated for four cases. Furthermore, they conducted experimental evaluation of rule induction method based on this framework, which gave comparable results compared with conventional approaches.

Table 1. Incremental Sampling Scheme

	R	D	$R \wedge D$
1.	0	0	0
2.	0	+1	0
3.	+1	0	0
4.	+1	+1	+1

In this paper, we extend this scheme by including the negations of R and D to evaluate the set-based indices where the negated terms are used. The results show that the evaluation framework gives a powerful tool for evaluation of set-based indices. Especially, it is found that the behavior of indices can be determined by a firstly given dataset.

The paper is organized as follows: Section 2 makes a brief description about rough set theory and the definition of probabilistic rules based on this theory. Section 3 discusses a former framework for incremental rule induction methods. Then, Section 4 extends the existing framework and applies it to evaluation of other indices. Finally, Section 5 concludes this paper.

2 Rough Sets and Probabilistic Rules

2.1 Rough Set Theory

Rough set theory clarifies set-theoretic characteristics of the classes over combinatorial patterns of the attributes, which are precisely discussed by Pawlak [10, 11]. This theory can be used to acquire some sets of attributes for classification and can also evaluate how precisely the attributes of database are able to classify data. One of the main features of rough set theory is to evaluate the relationship between the conditional attributes and the decision attributes by using the hidden set-based relations. Let a conditional attribute or conjunctive formula of attributes a decision attribute be denoted by R and D. Then, a relation between R and D can be evaluated by each supporting sets ($[x]_R$ and $[x]_D$) and their overlapped region denoted by $R \wedge D$ ($[x]_R \cap [x]_D$). If $[x]_R \subset [x]_D$, then a proposition $R \rightarrow D$ will hold and R will be a part of lower approximation of D. Dually, D can be called a upper approximation of R. In this way, we can define the characteristics of classification in the set-theoretic framework. Let n_R, n_D and n_{RD} denote the cardinality of $[x]_R$, $[x]_D$ and $[x]_R \cap [x]_D$, respectively. Accuracy (true predictive value) and coverage (true positive rate) can be defined as:

$$\alpha_R(D) = \frac{n_{RD}}{n_R} \qquad and \qquad \kappa_R(D) = \frac{n_{RD}}{n_D}, \tag{1}$$

It is notable that $\alpha_R(D)$ measures the degree of the sufficiency of a proposition, $R \rightarrow D$, and that $\kappa_R(D)$ measures the degree of its necessity. For example, if $\alpha_R(D)$ is equal to 1.0, then $R \rightarrow D$ is true. On the other hand, if $\kappa_R(D)$ is equal to 1.0, then $D \rightarrow R$ is true. Thus, if both measures are 1.0, then $R \leftrightarrow D$.

For further information on rough set theory, readers could refer to [9–11].

2.2 Probabilistic Rules

The simplest probabilistic model is that which only uses classification rules which have high accuracy and high coverage. [1] This model is applicable when rules of high accuracy can be derived. Such rules can be defined as:

$$R \stackrel{\alpha,\kappa}{\rightarrow} d \text{ s.t.} \quad \begin{array}{l} R = \vee_i R_i = \vee \wedge_j [a_j = v_k], \\ \alpha_{R_i}(D) > \delta_\alpha \text{ and } \kappa_{R_i}(D) > \delta_\kappa, \end{array}$$

where δ_α and δ_κ denote given thresholds for accuracy and coverage, respectively. where $|A|$ denotes the cardinality of a set A, $\alpha_R(D)$ denotes an accuracy of R as to classification of D, and $\kappa_R(D)$ denotes a coverage, or a true positive rate of R to D, respectively. We call these two inequalities *rule selection inequalities*.

3 Incremental Rule Induction

From the definition of accuracy and coverage, Equations(1) accuracy and coverage may nonmonotonically change. Since the above classification gives four additional patterns, we will consider accuracy and coverage for each case as shown in Table 2. in which $|[x]_R(t)|$, $|D(t)|$ and $|[x]_R \cap D(t)|$ are denoted by n_R, n_D andn_{RD}. As shown in [8], Table 3 gives the classification of four cases

Table 2. Four patterns for an additional Example

t:	$[x]_R(t)$	$D(t)$	$[x]_R \cap D(t)$
	n_R	n_D	n_{RD}

t+1	$[x]_R(t+1)$	$D(t+1)$	$[x]_R \cap D(t+1)$
	$n_R + 1$	$n_D + 1$	$n_{RD} + 1$
	$n_R + 1$	n_D	n_{RD}
	n_R	$n_D + 1$	n_{RD}
	n_R	n_D	n_{RD}

of an additional example. These updates can be visualized in a simplified form as shown in Table 4, where $\rightarrow$, $\uparrow$ and $\downarrow$ denotes stable, increase and decrese in sample or indices. It is notable that updates of accuracy and coverage are complementary: that is, each pattern of change of values of accuracy and coverage corresponds to each pattern of four possiblities for incremental sampling.

[1] In this model, we assume that accuracy is dominant over coverage.

Table 3. Summary of Change of Accuracy and Coverage

			$\alpha(t+1)$	$\kappa(t+1)$	
1.	n_R	n_D	n_{RD}	$\alpha(t)$	$\kappa(t)$
2.	n_R	n_D+1	n_{RD}	$\alpha(t)$	$\frac{\kappa(t)n_D}{n_D+1}$
3.	n_R+1	n_D	n_{RD}	$\frac{\alpha(t)n_R}{n_R+1}$	$\kappa(t)$
4.	n_R+1	n_D+1	$n_{RD}+1$	$\frac{\alpha(t)n_R+1}{n_R+1}$	$\frac{\kappa(t)n_D+1}{n_D+1}$

Table 4. Incremental Sampling Scheme for Accuracy and Coverage

	R	D	$R \wedge D$	$\alpha_R(D)$	$\kappa_R(D)$
1.	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$
2.	$\rightarrow$	$\uparrow$	$\rightarrow$	$\downarrow$	$\rightarrow$
3.	$\uparrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\downarrow$
4.	$\uparrow$	$\uparrow$	$\rightarrow$	$\uparrow$	$\uparrow$

3.1 Updates of Accuracy and Coverage

From Table 3, updates of Accuracy and Coverage can be calculated from the original datasets for each possible case. Since rules is defined as a probabilistic proposition with two inequalities, supporting sets should satisfy the following constraints:

$$\alpha(t+1) > \delta_\alpha \quad , \quad \kappa(t+1) > \delta\kappa \tag{2}$$

Then, the conditions for updating can be calculated from the original datasets: when accuracy or coverage does not satisfy the constraint, the corresponding formula should be removed from the candidates. On the other hand, both accuracy and coverage satisfy both constraints, the formula should be included into the candidates. Thus, the following inequalities are important for inclusion of R into the conditions of rules for D:

$$\alpha(t+1) = \frac{\alpha(t)n_R + 1}{n_R + 1} > \delta_\alpha, \quad , \quad \kappa(t+1) = \frac{\kappa(t)n_D + 1}{n_D + 1} > \delta\kappa.$$

For its exclusion, the following inequalities are important:

$$\alpha(t+1) = \frac{\alpha(t)n_R}{n_R + 1} < \delta_\alpha,$$
$$\kappa(t+1) = \frac{\kappa(t)n_D}{n_D + 1} < \delta\kappa.$$

Thus, the following inequalities are obtained for accuracy and coverage.

Theorem 1. *If accuracy and coverage of a formula R to d satisfies one of the following inequalities, then R may include into the candidates of formulae for probabilistic rules.*

$$\frac{\delta_\alpha(n_R+1) - 1}{n_R} < \alpha_R(D)(t) \leq \delta_\alpha, \tag{3}$$

$$\frac{\delta_\kappa(n_D+1)-1}{n_D} < \kappa_R(D)(t) \leq \delta_\kappa. \tag{4}$$

A set of R which satisfies the above two constraints is called **in subrule layer**.

Theorem 2. *If accuracy and coverage of a formula R to d satisfies one of the following inequalities, then R may exclude from the candidates of formulae for probabilistic rules.*

$$\delta_\alpha < \alpha_R(D)(t) < \frac{\delta_\alpha(n_R+1)}{n_R}, \tag{5}$$

$$\delta_\kappa < \kappa_R(D)(t) < \frac{\delta_\kappa(n_D+1)}{n_D}. \tag{6}$$

A set of R which satisfies the above two constraints is called **out subrule layer**.

It is notable that the lower and upper bounds can be calculated from the original datasets.

Select all the formulae whose accuracy and coverage satisfy the above inequalities They will be a candidate for updates. A set of formulae which satisfies the rule selection inequalities for probabilistic rules is called a *rule layer* and a set of formulae which satisfies Eqn (3) and (4) is called a *subrule layer (in)*. For more detail, please refer to [8].

3.2 Lift

Next, let us take a $lift$ measure, denoted by $l_R(D)$, which is defined as:

$$l_R(D) = \frac{n_{RD}}{n_R n_D},$$

which can be viewed as an index for degree of statistical independence. By using the definition of accuracy and coverage, the lift can be reformulate as:

$$l_R(D) = \frac{\alpha_R(D)}{n_D} = \frac{\kappa_R(D)}{n_R}$$

Then, updates of $l_R(D)$ can be illustrated as in Table 5. An interesting case is the fourth class where accuracy and coverage will increase. Since n_D and n_R also increase, the updates of lift will be dependent on the value of n_D and n_R. When these numbers are small, the lift may decrease, but when these numbers are sufficiently large, the value will increase, but the degree of increase will be smaller.

The similar technique in the above section can be applied, and summary of change of lift can be derived as in Table 6.

For both R and d positive, more direct calculation can be obtained as follows.

Table 5. Incremental Sampling Scheme for Lift

	R	D	$R \wedge D$	$\alpha_R(D)$	$\kappa_R(D)$	$l_R(D)$
1.	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$
2.	$\rightarrow$	$\uparrow$	$\rightarrow$	$\downarrow$	$\rightarrow$	$\downarrow$
3.	$\uparrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\downarrow$	$\downarrow$
4.	$\uparrow$	$\uparrow$	$\rightarrow$	$\uparrow$	$\uparrow$	?

Table 6. Summary of Change of Lift

			$\alpha(t+1)$	$l(t+1)$
n_R	n_D	n_{RD}	$\alpha(t)$	$l(t)$
n_R+1	n_D	n_{RD}	$\frac{\alpha(t)n_R}{n_R+1}$	$\frac{\alpha(t)n_R}{n_D(n_R+1)}$
n_R	n_D+1	n_{RD}	$\alpha(t)$	$\frac{\alpha(t)}{n_D+1}$
n_R+1	n_D+1	$n_{RD}+1$	$\frac{\alpha(t)n_R+1}{n_R+1}$	$\frac{\alpha(t)n_R+1}{(n_D+1)(n_R+1)}$

3.3 *R* and *d*: Positive

Finally, the fourthcase is when an additional example satisfies R and belongs to d.

$$\begin{aligned}\Delta_4 l_R(D)(t+1) &= l_R(D)(t+1) - l_R(D)(t) \\ &= \frac{n_{RD}+1}{(n_D+1)(n_R+1)} - \frac{n_{RD}}{n_R n_D} \\ &= \frac{n_D n_R - n_{RD}(n_R+n_D+1)}{n_D n_R(n_R+1)(n_D+1)} \\ &= \frac{1 - l(t)(n_R+n_D+1)}{(n_R+1)(n_D+1)}\end{aligned}$$

Thus, when

$$l_R(D)(t) < \frac{1}{n_R+n_D+1},$$

the difference $\Delta l_R(D)(t+1)$ will be positive. Thus, the table will be shown in Table 7 if the number of sample is sufficiently large.

Usually, lift is used in the context where the value is larger than 1, the difference is negative. Therefore, the change of the lift measure is monotonically negative, whose behavior is very different from accuracy and coverage. This shows that we do not have to consider the subrule layer for *in* when we use only lift for rule selection inequality.

3.4 Threshold for Lift

Since the lift will be monotonically decreasing, the update scheme is very different from the pair of accuracy and coverage. If the inequality of lift is given as $l_R(D) >$

Table 7. Incremental Sampling Scheme for Lift

	R	D	$R \wedge D$	$\alpha_R(D)$	$\kappa_R(D)$	$l_R(D)$
1.	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$
2.	$\rightarrow$	$\uparrow$	$\rightarrow$	$\downarrow$	$\rightarrow$	$\downarrow$
3.	$\uparrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\downarrow$	$\downarrow$
4.	$\uparrow$	$\uparrow$	$\rightarrow$	$\uparrow$	$\uparrow$	$\downarrow$

δ_l, where δ_l denotes the threshold of l, then four cases will give the following constraints:

$$l(t) - \frac{n_{RD}}{(n_R+1)n_R n_D} > \delta_l$$
$$l(t) - \frac{n_{RD}}{(n_D+1)n_R n_D} > \delta_l$$
$$l(t) - \frac{1 - l(t)(n_R+n_D+1)}{(n_R+1)(n_D+1)} > \delta_l$$

Therefore, the value of lift measure should satisfy the following constraints:

$$l(t) > \frac{\delta_l(n_R+1)n_D}{n_R n_D},$$
$$l(t) > \frac{\delta_l(n_D+1)n_R}{n_R n_D},$$
$$l(t) > \frac{\delta_l(n_D+1)(n_R+1)-1}{n_R n_D}$$

which are already sufficiently complex. However, these inequality shows that the update of $l(t)$ and its constraints are determined by the values given in an original dataset.

Thus,

Theorem 3. *If accuracy and coverage of a formula R to d satisfies one of the following inequalities, then R may exclude from the candidates of formulae for probabilistic rules.*

$$l(t) > \max\{\frac{\delta_l(n_R+1)n_D}{n_R n_D}, \frac{\delta_l(n_D+1)n_R}{n_R n_D}, \frac{\delta_l(n_D+1)(n_R+1)-1}{n_R n_D}\}$$

A set of R which satisfies the above two constraints is called **out subrule layer**. □

From the viewpoint of classification of four possibilities, the usage of lift is not enough and another indice should be added. For example, coverage can be used for this purpose as shown in Table 7

4 Extension of Incremental Sampling Scheme

For definition of a set-based index, the negation of R and D, denoted by $\neg R$ and $\neg D$ may be needed. For example, specificity, $spec_R(D)$ is defined as:

$$spec_R(D) = \frac{n_{\neg R \neg D}}{n_{\neg D}} = \kappa_{\neg R}(\neg D)$$

This value is used with sensitivity, denoted by $sen_R(D)$, whose definition is equal to coverage in the context of decision theory:

$$sen_R(D) = \kappa_R(D).$$

In this case, a table of incremental sampling scheme need to be extended. Table 8 shows the extension of incremental sampling scheme. From this table, we can

Table 8. Extended Incremental Sampling Scheme

	R	D	$\neg R$	$\neg D$	$R \wedge D$	$\neg R \wedge D$	$R \wedge \neg D$	$\neg R \wedge \neg D$
1.	0	0	+1	+1	0	0	0	+1
2.	0	+1	+1	0	0	+1	0	0
3.	+1	0	0	+1	0	0	+1	0
4.	+1	+1	0	0	+1	0	0	0

construct extended sampling scheme for accuracy and coverage as shown in Table 9. Table 9 gives the incremental update of sensitvitiy and specifity as shown

Table 9. Extended Incremental Sampling Scheme for Accuracy and Coverage

	$\alpha_R(D)$	$\kappa_R(D)$	$\alpha_R(\neg D)$	$\kappa_R(\neg D)$	$\alpha_{\neg R}(D)$	$\kappa_{\neg R}(D)$	$\alpha_{\neg R}(\neg D)$	$\kappa_{\neg R}(\neg D)$
1.	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\downarrow$	$\downarrow$	$\rightarrow$	$\uparrow$	$\uparrow$
2.	$\rightarrow$	$\downarrow$	$\rightarrow$	$\rightarrow$	$\uparrow$	$\uparrow$	$\downarrow$	$\rightarrow$
3.	$\downarrow$	$\rightarrow$	$\uparrow$	$\uparrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\downarrow$
4.	$\uparrow$	$\uparrow$	$\downarrow$	$\rightarrow$	$\rightarrow$	$\downarrow$	$\rightarrow$	$\rightarrow$

in Table 10, which shows the behavior of one index is complementary to that of the other index.

Although we have uncertainty in $d_R(D)$ in general, the value will increase if the number of sample will be sufficiently large.

Table 10. Incremental Sampling Scheme for Sensitivity and Specificity

	R	D	$sens_R(D)$	$spec_R(D)$
1.	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\uparrow$
2.	$\rightarrow$	$\uparrow$	$\rightarrow$	$\downarrow$
3.	$\uparrow$	$\rightarrow$	$\downarrow$	$\rightarrow$
4.	$\uparrow$	$\uparrow$	$\uparrow$	$\rightarrow$

4.1 Bayesian Confirmation Measure

Greco, Matarazzo and Slowinski proposed two bayesian confirmation measure for rule induction [12, 13]:

$$
\begin{aligned}
d_R(D) &= P(D|R) - P(D) \\
r_R(D) &= \log \frac{P(D|R)}{P(D)} \\
l2_R(D) &= \log \frac{P(R|D)}{P(R|\neg D)} \\
f_R(D) &= \frac{P(R|D) - P(R|\neg D)}{P(R|D) + P(R|\neg D)} \\
s_R(D) &= P(D|R) - P(D|\neg R). \\
b_R(D) &= P(R, D) - P(R)P(D)
\end{aligned}
$$

Originally, $l2_R(D)$ is defined as l in [13] . However, we have already used $l_R(D)$ for lift, so we denote it by $l2_R(D)$. In our notation, these values can be reformulated as follows.

$$
\begin{aligned}
d_R(D) &= \alpha_R(D) - \frac{n_D}{U} \\
r_D(D) &= \log \alpha_R(D) n_D \\
l2_R(D) &= \log \frac{\kappa_R(D)}{\kappa_R(\neg D)} \\
f_R(D) &= \frac{\kappa_R(D) - \kappa_R(\neg D)}{\kappa_R(D) + \kappa_R(\neg D)} \\
s_R(D) &= \alpha_R(D) - \alpha_{\neg R}(D) \\
b_R(D) &= n_R n_D (l_R(D) - 1)
\end{aligned}
$$

For example, using Table 9, the behavior of f and s is given as Table 11. Since f-measure is defined as a ratio, simple qualitative estimation cannot give their behavior, we have to calculate the difference by using ordinary calculation.

Table 11. Incremental Sampling Scheme for Bayesian Confirmation Measures f and s

	R	D	$s_R(D)$	$f_R(D)$
1.	$\rightarrow$	$\rightarrow$	$\uparrow$	$\uparrow$
2.	$\rightarrow$	$\uparrow$	$\downarrow$	$\downarrow$
3.	$\uparrow$	$\rightarrow$	$\downarrow$	?
4.	$\uparrow$	$\uparrow$	$\uparrow$	?

***R*:Positive.** Since $\kappa_R(D)(t+1) = \kappa_R(D)(t)$ and $\kappa_R(\neg D)(t+1) = \kappa_R(\neg D)(t) + \Delta_3\kappa_R(\neg D))(t)$,

$$f_3(t+1) = \frac{\kappa_R(D)(t) - (\kappa_R(\neg D)(t) + \Delta_3\kappa_R(\neg D))(t))}{\kappa_R(D)(t) + (\kappa_R(\neg D)(t) + \Delta_3\kappa_R(\neg D))(t))}.$$

Thus,

$$\begin{aligned}\Delta_3 f(t+1) &= f(t+1) - f(t) \\ &= \frac{-2\Delta_3\kappa_R(\neg D)(t)\kappa_R(D)(t)}{Denf_3(t+1)f(t)} < 0,\end{aligned} \tag{7}$$

where $Denf(t+1)f(t)$ denotes the denominator of $f(t+1)f(t)$.

***R* and *d* :Positive.** Since $\kappa_R(\neg D)(t+1) = \kappa_R(\neg D)(t)$ and $\kappa_R(D)(t+1) = \kappa_R(D)(t) + \Delta_4\kappa_R(D))(t)$,

$$f_4(t+1) = \frac{\kappa_R(D)(t) + \Delta_4\kappa_R(D)(t) - \kappa_R(\neg D)(t)}{\kappa_R(D)(t) + \Delta_4\kappa_R(D)(t) + \kappa_R(D)(t).}$$

Thus,

$$\begin{aligned}\Delta_4 f(t+1) &= f(t+1) - f(t) \\ &= \frac{+2\Delta_4\kappa_R(D)(t)\kappa_R(\neg D)(t)}{Denf_4(t+1)f(t)} > 0,\end{aligned} \tag{8}$$

where $Denf(t+1)f(t)$ denotes the denominator of $f(t+1)f(t)$. Thus, Table 11 is obtained as shown in Table 12.

In the same way, the qualitative behavior of other measures is obtained as shown in Table 13. Thus, qualitative behavior of these confirmation measures is the same as the sum of sensitivity and specificity, which shows that qualitative behavior of sentitivity and one of specificity are components of Bayesian confirmation measures.

5 Conclusion

This paper proposes a new framework for evaluation of set-based indices based on incremental sampling. Since these indices are defined by the relations between

Table 12. Incremental Sampling Scheme for Bayesian Confirmation Measures (2)

	R	D	$s_R(D)$	$f_R(D)$
1.	$\rightarrow$	$\rightarrow$	$\uparrow$	$\uparrow$
2.	$\rightarrow$	$\uparrow$	$\downarrow$	$\downarrow$
3.	$\uparrow$	$\rightarrow$	$\downarrow$	$\downarrow$
4.	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$

Table 13. Incremental Sampling Scheme for Other Bayesian Measures

	R	D	$d_R(D)$	$r_R(D)$	$l2_R(D)$	$b_R(D)$
1.	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\uparrow$	$\rightarrow$
2.	$\rightarrow$	$\uparrow$	$\downarrow$	$\uparrow$	$\downarrow$	$\downarrow$
3.	$\uparrow$	$\rightarrow$	$\downarrow$	$\rightarrow$	$\rightarrow$	$\downarrow$
4.	$\uparrow$	$\uparrow$	?	$\uparrow$	$\uparrow$	$\downarrow$

conditional attributes (R) and decision attribute(D), incremental sampling gives four possible cases according to the increment of sets for R or D. Using this classification, the behavior of indices can be evaluated for four cases. In this paper, the updates of accuracy, coverage and lift are shown. Interestingly, the lift measure is monotonically decreasing for large sample. We also introduce a table for qualitative behavior of an index.

This scheme is extended by including the negations of R and D to evaluate the set-based indices where the negated terms are used. We applied this scheme to examining the behavior of sensitivity, specificity and two Bayesian confirmation measures. The results show that the evaluation framework gives a powerful tool for evaluation of set-based indices. Especially, it is found that the behavior of indices can be determined by a firstly given dataset..

This is a preliminary work on incremental learning based on rough set theory and it is our future work to conduct further empirical validatioons and to establish a theoretical basis of this method.

Acknowledgements. This research is supported by Grant-in-Aid for Scientific Research (B) 24300058 from Japan Society for the Promotion of Science(JSPS).

References

1. Breiman, L., Freidman, J., Olshen, R., Stone, C.: Classification And Regression Trees. Wadsworth International Group, Belmont (1984)
2. Cestnik, B., Kononenko, I., Bratko, I.: Assistant 86: A knowledge-elicitation tool for sophisticated users. In: EWSL, pp. 31–45 (1987)
3. Quinlan, J.: C4.5 - Programs for Machine Learning. Morgan Kaufmann, Palo Alto (1993)

4. Clark, P., Niblett, T.: The cn2 induction algorithm. Machine Learning 3 (1989)
5. Michalski, R., Mozetic, I., Hong, J., Lavrac, N.: The multi-purpose incremental learning system aq15 and its testing application to three medical domains. In: Proceedings of the Fifth National Conference on Artificial Intelligence, pp. 1041–1045. AAAI Press, Menlo Park (1986)
6. Shan, N., Ziarko, W.: Data-based acqusition and incremental modification of classification rules. Computational Intelligence 11, 357–370 (1995)
7. Utgoff, P.E.: Incremental induction of decision trees. Machine Learning 4, 161–186 (1989)
8. Tsumoto, S., Hirano, S.: Incremental rules induction based on rule layers. 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. 139–148. Springer, Heidelberg (2012)
9. Tsumoto, S.: Automated induction of medical expert system rules from clinical databases based on rough set theory. Information Sciences 112, 67–84 (1998)
10. Pawlak, Z.: Rough Sets. Kluwer Academic Publishers, Dordrecht (1991)
11. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46, 39–59 (1993)
12. Greco, S., Słowiński, R., Szczęch, I.: Analysis of symmetry properties for bayesian confirmation measures. 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. 207–214. Springer, Heidelberg (2012)
13. Greco, S., Pawlak, Z., Slowinski, R.: Can bayesian confirmation measures be useful for rough set decision rules? Engineering Applications of Artificial Intelligence 17, 345–361 (2004)
14. Li, T., Nguyen, H.S., Wang, G., Grzymala-Busse, J., Janicki, R., Hassanien, A.E., Yu, H. (eds.): RSKT 2012. LNCS, vol. 7414. Springer, Heidelberg (2012)

Recent Advances in Decision Bireducts: Complexity, Heuristics and Streams*

Sebastian Stawicki[1] and Dominik Ślęzak[1,2]

[1] Institute of Mathematics, University of Warsaw
ul. Banacha 2, 02-097 Warsaw, Poland
[2] Infobright Inc.
ul. Krzywickiego 34, lok. 219, 02-078 Warsaw, Poland

Abstract. We continue our research on decision bireducts. For a decision system $\mathbb{A} = (U, A \cup \{d\})$, a decision bireduct is a pair (B, X), where $B \subseteq A$ is a subset of attributes discerning all pairs of objects in $X \subseteq U$ with different values on the decision attribute d, and where B and X cannot be, respectively, reduced and extended. We report some new results related to NP-hardness of extraction of optimal decision bireducts, heuristics aimed at searching for sub-optimal decision bireducts, and applications of decision bireducts to stream data mining.

Keywords: Bireducts, NP-hardness, Heuristic Search, Data Streams.

1 Introduction

Decision reducts have been found a number of applications in feature selection and knowledge representation [1]. Notions analogous to decision reducts occur in many areas of science, such as Markov boundaries in probabilistic modeling [2] or signatures in bioinformatics [3]. As one of extensions, approximate decision reducts are studied in order to search for irreducible subsets of attributes that *almost* determine decisions in real-world, noisy data sets [4].

Bireducts were proposed as a new extension of decision reducts in [5] and further developed in [6]. Their interpretation seems to be simpler than in the case of most of types of approximate decision reducts known from the literature. The emphasis here is on both a subset of attributes, which describes decisions, and a subset of objects, for which such a description is valid.

This paper continues our research on bireducts, both with respect to their comparison to classical and approximate decision reducts, and their applications in new areas. In Section 2, we recall basics of decision bireducts. In Section 3, we prove NP-hardness of one of possible optimization problems related to extraction of decision reducts from data. In Section 3, we show some new interpretations of decision bireducts, which are useful for their heuristic search. In Section 4, we outline how to apply decision bireducts in data stream analysis. In Section 5, we discuss some of future perspectives and conclude the paper.

* This research was partly supported by the Polish National Science Centre (NCN) grants 2011/01/B/ST6/03867 and 2012/05/B/ST6/03215.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 200–212, 2013.

Table 1. System $\mathbb{A} = (U, A \cup \{d\})$ with 14 objects in U, four attributes in A, and d = Sport?

	Outlook	Temp.	Humid.	Wind	Sport?
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

Table 2. Several examples of bireducts (B, X) for $\mathbb{A}$ in Table 1

(B, X)
({O},{1..5,7..8,10,12..13})
({O},{1..3 6..8 12..14})
({O},{3 6..7 9 11..14})
({O T},{1..4 6..10 12..13})
({O H},{1..3 6..9 11..14})
({O T W},{1..14})
({O H W},{1..14})
({O W},{2..7 9..10 12..14})
({T},{3..4 6 10..13})
({T H},{1..2 6 8 10..11 13..14})
({T W},{1..2 4..5 7 9..10 14})
({T W},{2..6 9..13})
({H W},{1 5..6 8..10 12..13})
({W},{2..6 9..10 13..14})

2 Basics of Decision Bireducts

First formulation of decision bireducts occurred in [5], where their Boolean characteristics and simple permutation-based search algorithms were proposed in analogy to classical reducts [7]. It was also discussed in what sense ensembles of decision bireducts are better than ensembles of approximate reducts, which – although quite useful in practice [8] – do not allow for explicit analysis whether particular reducts repeat mistakes on the same cases.

We use a standard representation of tabular data in form of decision systems [9]. A decision system is a tuple $\mathbb{A} = (U, A \cup \{d\})$, where U is a set of objects, A is a set of attributes and $d \notin A$ is a decision attribute. For simplicity, we refer to the elements of U using their ordinal numbers $i = 1, ..., |U|$, where $|U|$ denotes the cardinality of U. We treat all attributes $a \in A \cup \{d\}$ as functions $a : U \to V_a$, V_a denoting a's domain. The values $v_d \in V_d$ correspond to decision classes that we want to describe using the values of attributes in A.

Definition 1. *[9] We say that $B \subseteq A$ is a decision reduct for decision system $\mathbb{A} = (U, A \cup \{d\})$, iff it is an irreducible subset of attributes such that each pair $i, j \in U$ satisfying inequality $d(i) \neq d(j)$ is discerned by B.*

As an example, for $\mathbb{A}$ in Table 1, there are two reducts: {Outlook, Temp., Wind} and {Outlook, Humid., Wind} (or {O, T, W} and {O, H, W} for short).

Definition 2. *[5] Let $\mathbb{A} = (U, A \cup \{d\})$ be a decision system. A pair (B, X), where $B \subseteq A$ and $X \subseteq U$, is called a decision bireduct, iff the following holds:*

- *B discerns all pairs $i, j \in X$ where $d(i) \neq d(j)$ (further denoted as $B \Rrightarrow_X d$);*
- *There is no $C \subsetneq B$ such that $C \Rrightarrow_X d$;*
- *There is no $Y \supsetneq X$ such that $B \Rrightarrow_Y d$.*

For the decision system $\mathbb{A}$ given in Table 1, some examples of decision bireducts are presented in Table 2.

A decision bireduct (B, X) can be regarded as the basis for an inexact functional dependency linking the subset of attributes B with the decision d in a degree X, denoted as $B \Rrightarrow_X d$ in Definition 2. Furthermore, the objects in $U \setminus X$ can be treated as outliers of $B \Rrightarrow_X d$.

Further in this paper, we focus on bireducts and their corresponding inexact dependencies formulated in terms of standard discernibility, where $B \subseteq A$ discerns objects $i, j \in U$ iff there is $a \in B$ such that $a(i) \neq a(j)$. However, as pointed out in Section 6, one can also consider some generalizations, such as e.g. bireducts based on fuzzy discernibility [10].

3 Decision Bireduct Optimization

There are a number of NP-hardness results related to extracting optimal decision reducts and approximate reducts from data [11]. In the case of decision bireducts, one may think about quite different optimization criteria with respect to a balance between the number of involved attributes and objects. The following form of a constraint for decision bireducts is somewhat analogous to those studied for frequent itemsets and patterns [12]. However, let us emphasize that this is just one of many ways of interpreting optimal decision bireducts.

Definition 3. *Let $\varepsilon \in [0, 1)$ be given. We say that a pair (B, X), $B \subseteq A$ and $X \subseteq U$, is a ε-bireduct, if it is a bireduct and the following holds: $|X| \geq (1-\varepsilon)|U|$.*

Definition 4. *Let $\varepsilon \in [0, 1)$ be given. By the Minimal ε-Decision Bireduct Problem ($M\varepsilon DBP$) we mean a task of finding for each given decision system $\mathbb{A} = (U, A \cup \{d\})$ a ε-bireduct (B, X) with the lowest cardinality of B.*

In order to prepare the ground for the major result in this section, let us recall the following correspondence between decision bireducts and one of specific types of approximate decision reducts.

Definition 5. *[13] Let $\varepsilon \in (0, 1]$ and a decision system $\mathbb{A} = (U, A \cup \{d\})$ be given. For each $B \subseteq A$, consider the quantity $M_{\mathbb{A}}(B) =$*

$$= \frac{1}{|U|} \left| \left\{ u \in U : d(u) = \underset{v_d \in V_d}{\operatorname{argmax}} \left| \{u' \in U : \forall_{a \in B}\, a(u') = a(u) \wedge d(u') = v_d\} \right| \right\} \right| \tag{1}$$

We say that $B \subseteq A$ is an (M, ε)-approximate reduct, iff

$$M_{\mathbb{A}}(B) \geq 1 - \varepsilon \tag{2}$$

and there is no proper subset of B, which would hold an analogous inequality.

Original formulation of the above definition in [13] was a bit different, with constraint $M_{\mathbb{A}}(B) \geq (1 - \varepsilon) M_{\mathbb{A}}(A)$ instead of $M_{\mathbb{A}}(B) \geq 1 - \varepsilon$. Thus, formally, we should refer to the above as to a *modified* (M, ε)-approximate reduct.

A way of defining $M_{\mathbb{A}}(B)$ is different as well, although mathematically equivalent to that in [13]. We rewrite it in the above form in order to emphasize that it is actually the ratio of objects in U that would be correctly classified by if-then decision rules learned from $\mathbb{A} = (U, A \cup \{d\})$ with the *attribute* = *value* conditions over B and *decision* = *value* consequences specified by identifying decision values assuring the highest confidence for each of rules.

For a consistent decision system, i.e. $\mathbb{A} = (U, A \cup \{d\})$, where A enables to fully discern all pairs of objects from different decision classes, there is $M_{\mathbb{A}}(A) = 1$. In such a case, original and modified conditions for an (M, ε)-approximate reduct are equivalent. Also, but only in consistent decision tables, (M, ε)-approximate reducts are equivalent to classical decision reducts for $\varepsilon = 0$.

In [6], a correspondence between decision bireducts and modified (M, ε)-approximate reducts was noticed. Consider a family of all subsets $X \subseteq U$ with which a given subset $B \subseteq A$ has a chance to form a bireduct:

$$X_B = \{X \subseteq U : \forall_{i,j \in X}\, d(i) \neq d(j) \Rightarrow \exists_{a \in B}\, a(i) \neq a(j)\} \tag{3}$$

Then the following equality holds:

$$M_{\mathbb{A}}(B) = \max_{X \in X_B} |X|/|U| \tag{4}$$

As a result, $B \subseteq A$ may be a modified (M, ε)-approximate reduct only if there is $X \subseteq U$ such that the pair (B, X) is an ε-bireduct. Given the computational complexity results reported in [13] for approximate decision reducts, we are now ready to formulate an analogous result for ε-bireducts:

Theorem 1. *Let $\varepsilon \in [0, 1)$ be given. $M\varepsilon DBP$ is NP-hard.*

Proof. In [13], it was shown that for each $\varepsilon \in [0, 1)$ treated as a constant, the problem of finding an (M, ε)-approximate reduct in an input decision system with minimum number of attributes is NP-hard. (Actually, in [13] it was presented for a far wider class of approximate decision reducts.)

The proof was based on polynomial reduction of the Minimal $\boldsymbol{\alpha}$-Dominating Set Problem ($M\alpha DSP$), aiming at finding minimal subsets of vertices that dominate at least $\alpha \times 100\%$ of all vertices in an input undirected graph. (NP-hardness of this problem was studied in [13] and later in [2].) For each $\varepsilon \in [0, 1)$, the formula for $\alpha(\varepsilon) \in (0, 1]$ can be constructed in such a way that for each graph $G = (V, E)$ being an input to $M\alpha(\varepsilon)DSP$ we can polynomially (with respect to the cardinality of V) construct a decision system with its minimal (M, ε)-approximate reducts equivalent to the $\alpha(\varepsilon)$-dominating sets in G.

Decision systems encoding graphs in the above reduction were consistent. Thus, following our earlier observation on equivalence of $M_{\mathbb{A}}(B) \geq (1 - \varepsilon)M_{\mathbb{A}}(A)$ and $M_{\mathbb{A}}(B) \geq 1 - \varepsilon$ in consistent decision systems, we can prove in the same way that finding of modified (M, ε)-approximate reducts is NP-hard too. As a result, by showing that the case of modified (M, ε)-approximate reducts can be polynomially reduced to $M\varepsilon DBP$ we will be able to finish the proof.

Such reduction is simple, as minimal modified (M, ε)-approximate reducts correspond to decision bireducts solving $M\varepsilon DBP$. Assume that a pair (B, X) is

an ε-bireduct with the lowest cardinality of B for a given $\mathbb{A} = (U, A \cup \{d\})$. Then B needs to be a minimal (M, ε)-approximate reduct for $\mathbb{A}$. This is because, first of all, thanks to (4) we have that $M_{\mathbb{A}}(B) \geq |X|/|U| \geq 1 - \varepsilon$. Secondly, assume that there is a subset $B' \subseteq A$ such that $M_{\mathbb{A}}(B') \geq 1 - \varepsilon$ and $|B'| < |B|$. Then, however, there would exist at least one ε-bireduct (B', X') for some $X' \subseteq U$, so (B, X) would not be a solution of $M\varepsilon DBP$. □

4 Heuristic Search for Bireducts

There are a number of possible algorithmic approaches to searching for decision bireducts. One can, e.g., extend techniques introduced earlier for decision reducts, like it was done for permutation-based algorithms in [5], where instead of orderings on attributes the orderings on mixed codes of attributes and objects were considered. One can also translate some algorithms aiming at finding approximate decision reducts onto the case of decision bireducts, basing on connections between both those notions outlined in [6]. Finally, specifically for the problem of searching for minimal ε-bireducts, one can adapt some mechanisms known from other areas, such as association rules with a constraint for minimum support [14], basing on representations developed for decision reducts [15].

Let us recall the above-mentioned algorithm proposed in [5], which is an extension of one of standard approaches to searching for decision reducts [7].

Proposition 1. *[5] Let $\mathbb{A} = (U, A \cup \{d\})$ be given. Enumerate attributes and objects as $A = \{a_1, ..., a_n\}$, $n = |A|$, and $U = \{1, ..., m\}$, $m = |U|$, respectively. Put $B = A$ and $X = \emptyset$. Let permutation $\sigma : \{1, ..., n + m\} \to \{1, ..., n + m\}$ be given. Consider the following procedure for each consecutive $i = 1, ..., n + m$:*

1. *If $\sigma(i) \leq n$, then attempt to remove attribute $a_{\sigma(i)}$ from B subject to the constraint $B \setminus \{a_{\sigma(i)}\} \Rrightarrow_X d$;*
2. *Else, attempt to add $\sigma(i) - n$ to X subject to the constraint $B \Rrightarrow_{X \cup \{\sigma(i)-n\}} d$.*

For each σ, the final outcome (B, X) is a decision bireduct. Moreover, for each bireduct (B, X) there exists input σ for which the above steps lead to (B, X).

The above method follows an idea of mixing the processes of reducing attributes and adding objects during the construction of bireducts. If we consider a special case of permutations $\sigma : \{1, ..., n + m\} \to \{1, ..., n + m\}$ where all objects are added to X prior to starting removing attributes from B, we will obtain the permutation-based characteristics of standard decision reducts. In a general case, the approximation threshold $\varepsilon \in [0, 1)$ introduced in Definition 3 is not defined explicitly but it is somehow expressed in a way permutations are generated. We can define a parameter that controls probability of selecting an attribute in first place rather than an object during the random generation of σ. When σ contains relatively more attributes at its beginning, a bireduct having smaller number of attributes but also higher number of outliers is likely to be obtained.

In the remainder of this section, we present two examples of algorithmic constructions enabling to harness various attribute reduction heuristics directly to

Table 3. $\mathbb{A}^* = (U, A \cup A^* \cup \{d\})$ corresponding to $\mathbb{A} = (U, A \cup \{d\})$ in Table 1

	Outlook	**Temp.**	**Humid.**	**Wind**	a_1^*	a_2^*	a_3^*	a_4^*	a_5^*	a_6^*	a_7^*	a_8^*	a_9^*	a_{10}^*	a_{11}^*	a_{12}^*	a_{13}^*	a_{14}^*	**Sport?**
1	sunny	hot	high	weak	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	no
2	sunny	hot	high	strong	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	no
3	overcast	hot	high	weak	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	yes
4	rain	mild	high	weak	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	yes
5	rain	cool	normal	weak	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	yes
6	rain	cool	normal	strong	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	no
7	overcast	cool	normal	strong	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	yes
8	sunny	mild	high	weak	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	no
9	sunny	cool	normal	weak	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	yes
10	rain	mild	normal	weak	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	yes
11	sunny	mild	normal	strong	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	yes
12	overcast	mild	high	strong	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	yes
13	overcast	hot	normal	weak	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	yes
14	rain	mild	high	strong	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	no

the task of searching for decision bireducts, after reformulation of the input data. The first of considered methods refers to the following representation:

Proposition 2. *[5] Let $\mathbb{A} = (U, A \cup \{d\})$ be a decision system. Consider the following Boolean formula with variables $\overline{i}$, $i = 1, ..., |U|$, and $\overline{a}$, $a \in A$:*

$$\tau_{\mathbb{A}}^{bi} = \bigwedge_{i,j:\, d(i) \neq d(j)} \left(\overline{i} \vee \overline{j} \vee \bigvee_{a:\, a(i) \neq a(j)} \overline{a} \right). \tag{5}$$

An arbitrary pair (B, X), $B \subseteq A$, $X \subseteq U$, is a decision bireduct, if and only if the Boolean formula $\bigwedge_{a \in B} \overline{a} \wedge \bigwedge_{i \notin X} \overline{i}$ is the prime implicant for $\tau_{\mathbb{A}}^{bi}$.

The above result shows a way to utilize techniques known from Boolean reasoning to search for decision bireducts as prime implicants [16]. It also illustrates that attributes and objects are to some extent equally important while constructing bireducts, analogously to some other approaches to deriving knowledge from data [17]. This intuition has led us to the following observation:

Proposition 3. *Let $\mathbb{A} = (U, A \cup \{d\})$ be a decision system. Consider a new system $\mathbb{A}^* = (U, A \cup A^* \cup \{d\})$, where the number of objects in U as well as their values for attributes from the original $\mathbb{A}$ remain unchanged, and where new attributes in $A^* = \{a_1^*, ..., a_m^*\}$, $m = |U|$, are defined as $a_j^*(i) = 1$ if $i = j$, and 0 otherwise. Then, the pair (B, X), $B \subseteq A$, $X \subseteq U$, is a decision bireduct in $\mathbb{A}$, iff $B \cup X^*$, for $X^* = \{a_i^* \in A^* : i \notin X\}$, is the decision reduct in $\mathbb{A}^*$.*

Proof. The proof is straightforward and we omit it because of space limitations.

An illustrative example of the considered transformation can be seen in Table 3. Certainly, it should be treated just as a starting point for developing more efficient algorithms, because decision systems of the form $\mathbb{A}^* = (U, A \cup A^* \cup \{d\})$ cannot be constructed explicitly for large data. An appropriate translation of methods aiming at searching for decision reducts in systems with large amount of attributes can be especially useful in this case [18].

Another way to employ standard reduct computations in order to search for decision bireducts can be generally referred to sampling methods [19].

Table 4. Indiscernibility classes induced by randomly selected attributes $\{T, H\}$ for decision system in Table 1

	Temp.	Humid.	Sport?
1	hot	high	no
2	hot	high	no
3	hot	high	yes
13	hot	normal	yes
4	mild	high	yes
8	mild	high	no
12	mild	high	yes
14	mild	high	no
10	mild	normal	yes
11	mild	normal	yes
5	cool	normal	yes
6	cool	normal	no
7	cool	normal	yes
9	cool	normal	yes

Table 5. $\mathbb{A}' = (U', A' \cup \{d\})$ for randomly selected representatives $U' = \{1, 6, 8, 10, 13\}$. Decision reduct $\{T, H\}$ in $\mathbb{A}'$ corresponds to bireduct $(\{T, H\}, \{1, 2, 6, 8, 10, 11, 13, 14\})$ in $\mathbb{A}$.

	Temp.	Humid.	Sport?
1	hot	high	no
6	cool	normal	no
8	mild	high	no
10	mild	normal	yes
13	hot	normal	yes

Table 6. The case of $U' = \{3, 6, 11, 12, 13\}$. Decision reduct $\{T\}$ in $\mathbb{A}'$ corresponds to bireduct $(\{T\}, \{3, 4, 6, 10, 11, 12, 13\})$ in $\mathbb{A}$.

	Temp.	Humid.	Sport?
3	hot	high	yes
6	cool	normal	no
11	mild	normal	yes
12	mild	high	yes
13	hot	normal	yes

Proposition 4. *For a given $\mathbb{A} = (U, A \cup \{d\})$, consider the three-step procedure:*

1. *Randomly select a subset of attributes $A' \subseteq A$;*
2. *Choose a single object from each of partition blocks induced by A' – all chosen objects form a subset denoted by $U' \subseteq U$;*
3. *Find a standard decision reduct $B \subseteq A'$ for the system $\mathbb{A}' = (U', A' \cup \{d\})$.*

Then the pair (B, X), where $X = \left\{u \in U : \exists_{x \in U'} \forall_{a \in B \cup \{d\}}\, a(x) = a(u)\right\}$, is a decision bireduct for $\mathbb{A}$. Moreover, each decision bireduct for $\mathbb{A}$ can be obtained as a result of the above steps, no matter what method is used in the third stage.

Proof. Again, we omit the proof because of space limitations.

We illustrate the above procedure by Tables 4, 5, 6. Let us note that the reduced decision systems obtained in the third of above steps are compact representations of if-then rules generated by attributes in B, with their supports summing up to the overall support $X \subseteq U$ of decision bireduct (B, X). However, consequences of those rules are not necessarily chosen in a way aiming at maximizing $|X|$. Quite oppositely, when combined with appropriate mechanisms of sampling, this process can lead to ensembles of decision bireducts based on possibly diversified subsets of attributes and objects, with the underlying if-then rules paying attention to the cases not covered by rules corresponding to other bireducts rather than the cases that are easiest to describe.

The algorithm outlined in Proposition 4 could be also modeled within the framework sketched in Proposition 1, by considering more specific permutations

$\sigma : \{1, ..., n+m\} \to \{1, ..., n+m\}$ with some amount of attributes at their beginning, an ordering of all objects in their middle, and the remainder of attributes at their very end. Indeed, in such a case, all attributes at the very beginning of σ will be removed; then, within each partition class induced by the remaining attributes, objects corresponding to only one of possible decision values will be added (precisely, it will be the decision value of the first element of a given partition class occurring in σ); and finally the algorithm will try to remove each of the remaining attributes according to their ordering in σ, subject the discernibility criteria with respect to the previously-added objects.

5 Bireducts in Data Streams

The main motivation for introducing decision bireducts in [5] was to establish a simple framework for constructing rough-set-based classifier ensembles, as well as to extend capabilities of decision reducts to model data dependencies. Going further, in [10] it was noticed that algorithms for extracting meaningful bireducts from data could be utilized to integrate the tasks of attribute and instance selection. Such a potential is also illustrated by Proposition 4, where the objects in U' actually define a classifier based on the resulting $B \subseteq A$.

Some areas of applications were also pointed out for other types of bireducts. In [20], so called information bireducts were employed to model context-based object similarities in multi-dimensional data sets. Information bireducts may be also able to approximate data complexity analogously to some well-known mathematical tools [21]. Indeed, by investigating cardinalities of minimal subsets of attributes discerning maximal subsets of objects we can attempt to express a potential of a data source to define different concepts of interest.

In this section, we study one more opportunity in front of bireducts. Let us consider a stream of objects that is too large to be stored or represents data collected on-line [22]. For our purposes, let us focus on a stream interpreted as a decision system $\mathbb{A} = (U, A \cup \{d\})$, where there is no possibility to look at the entire U at any moment of processing time. Instead, given a natural order over U, we can access some buffered data intervals, i.e., the subsets of objects that occur consecutively in a stream. The question is how to design and efficiently conduct a process of attribute reduction in such a dynamic situation.

One of possibilities would be to fix the amount of objects in each data interval and compare decision reducts obtained for such narrowed down decision systems, in a kind of sliding window fashion. However, an arbitrary choice of the interval length may significantly influence the results. Thus, it may be more reasonable to adaptively adjust data intervals with respect to the currently observed attribute dependencies. Moreover, if our goal is to search for stable subsets of attributes that remain decision reducts for possibly wide areas of data, then we should tend to maximizing data intervals in parallel to minimizing the amounts of attributes necessary to determine decision classes within them.

Definition 6. *Let $\mathbb{A} = (U, A \cup \{d\})$ be given. Let U be naturally ordered with its elements indexed by integers. Consider a pair (B, X), where $B \subseteq A$ and $X = \langle first, last \rangle$. We say that (B, X) is a temporal decision bireduct, iff:*

- *An inexact dependency $B \Rrightarrow_X d$ holds;*
- *There is no $C \subsetneq B$ such that $C \Rrightarrow_X d$;*
- *$B \Rrightarrow_Y d$ is not true for neither $Y = \langle first-1, last \rangle$ nor $Y = \langle first, last+1 \rangle$.*

The above modification of Definition 2 can serve as a background for producing bireducts (B, X) with no *holes* in X with respect to a given data flow. Below we sketch an example of heuristic extraction of such bireducts from data. From a technical point of view, it resembles Proposition 4 with respect to a random choice of a subset of attributes to be analyzed. From a more strategic perspective, let us note that our goal is now to save the identified temporal bireducts analogously to micro-clusters [23] or data blocks [24] constructed within other applications for the purposes of further steps of on-line or off-line analysis. This way of data stream processing may open new opportunities for the task of scalable attribute subset selection. For instance, basing on frequent occurrence of a given subset of attributes in the previously-found temporal bireducts, one can reason about its ability to induce a robust decision model.

Proposition 5. *Let $\mathbb{A} = (U, A \cup \{d\})$ be given. Let U be naturally ordered with its elements indexed by integers. Select an arbitrary $A' \subseteq A$ and put $B = X = \emptyset$. Consider the following steps for each consecutive i-th object in U:*

1. *If $B \Rrightarrow_{X \cup \{i\}} d$, then add i to X;*
2. *Else, save (B, X), add i to X, and do the following:*
   (a) *Put $B = A'$ and remove the oldest objects from X until there is $B \Rrightarrow_X d$;*
   (b) *Heuristically reduce redundant attributes under the constraint $B \Rrightarrow_X d$.*

Then, all pairs (B, X) saved during the above procedure are temporal bireducts for $\mathbb{A}$. Moreover, each temporal bireduct can be obtained as one of saved pairs (B, X) for some $A' \subseteq A$, no matter what method is used in the last step.

Proof. Consider a pair (B, X), where $X = \langle first, last \rangle$, which was saved in the step 2. For such a case, we know that $B \Rrightarrow_{\langle first, last \rangle} d$ and $B \not\Rrightarrow_{\langle first, last+1 \rangle} d$. Also, there is $B \not\Rrightarrow_{\langle first-1, last \rangle} d$ because the oldest object in X is removed only when the newly joined object cannot be handled together with some elements of X even when using the whole A'. Therefore, X cannot be extended backwards beyond object $first$. Also, because of reduction of redundant attributes, B is irreducible for X. Hence, all saved pairs (B, X) are temporal bireducts.

Now, consider a temporal bireduct $(B, \langle first, last \rangle)$ and put $A' = B$. Consider the first buffer including object $first$, i.e., $\langle older, first \rangle$, $older \leq first$. Each next entry until object $last$ will be added with no need of removing $first$ (otherwise there would be no $B \Rrightarrow_{\langle first, last \rangle} d$). Moreover, when adding $last$, all objects older than $first$ (if any of them are still present) will be erased from the buffer (otherwise there would be $B \Rrightarrow_{\langle first-1, last \rangle} d$). Finally, when adding object $last + 1$ to $\langle first, last \rangle$, we will need to remove $first$ (otherwise there would be $B \Rrightarrow_{\langle first, last+1 \rangle} d$), which results in saving $(B, \langle first, last \rangle)$. □

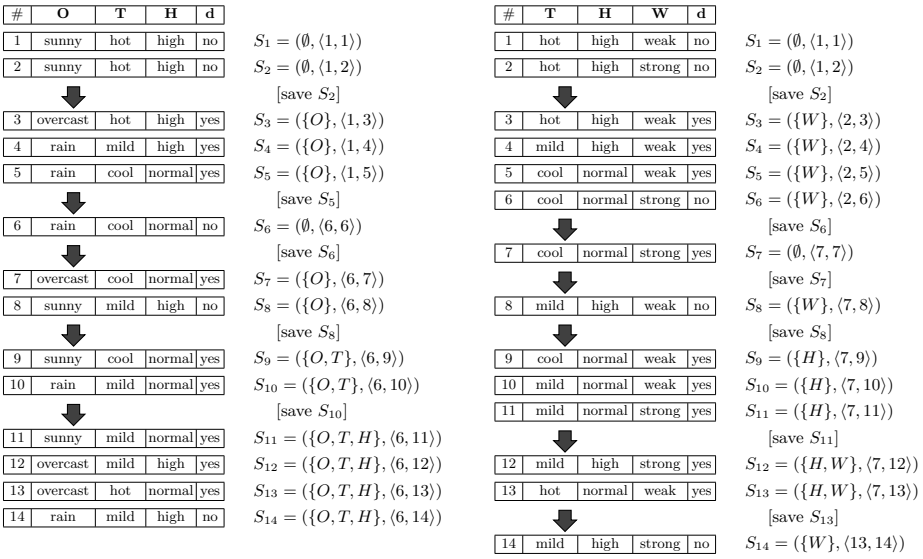


Fig. 1. Extraction of temporal bireducts from a data set in Table 1. The left- and right-side sequences correspond to subsets $A' = \{O, T, H\}$ and $A' = \{T, H, W\}$, respectively.

In Proposition 5, subsets $X \subseteq U$ are treated as the buffers of objects that appeared most recently in a data stream, within which a currently considered $B \subseteq A$ is sufficient to determine decision classes. As an illustration, consider the decision system in Table 1 and assume that we receive objects from $U = \{1, ..., 14\}$ one after the other. Let the i-th state of the process be denoted by $S_i = (B_i, X_i)$, where i is the number of objects already received from U and B_i is a decision reduct for the current buffer content X_i.

Figure 1 presents two examples of randomly chosen subsets of attributes. Let us concentrate on $A' = \{T, H, W\}$ and refer one more time to the permutation-based characteristics of decision reducts outlined e.g. in [7]. Namely, in the step 2(b) in Proposition 5, we are going to reduce attributes along $\sigma = \langle T, H, W\rangle$. In general, when following the same $\sigma : \{1, ..., n'\} \to \{1, ..., n'\}$, $n' = |A'|$, from the very beginning of a data stream, we can count on smoother evolution of subsets $B_i \subseteq A'$ for consecutive buffers. Furthermore, by working with a larger family of diversified subsets $A' \subseteq A$, we have a chance to witness the most representative changes of the observed temporal bireducts in time.

Let us now take a closer look at $A' = \{T, H, W\}$. The first two objects share the same decision. Thus, there is $S_2 = (\emptyset, \langle 1, 2\rangle)$. Further, $\emptyset \Rrightarrow_{\langle 1,3\rangle} d$ is not valid, so we save the temporal bireduct $(\emptyset, \langle 1, 2\rangle)$ and proceed with the step 2 in Proposition 5. As $\{T, H, W\}$ is insufficient to discern objects 1 and 3, we limit ourselves to $\langle 2, 3\rangle$. Starting from $B = A'$ and given $\sigma = \langle T, H, W\rangle$, we reduce T and H, which results in the pair $S_3 = (\{W\}, \langle 2, 3\rangle)$.

The next three objects do not break the dependency between $\{W\}$ and d. However, object 7 forces all earlier entries to be deleted. A different situation can be observed when adding the next two objects. In both cases, A' determines decision values, so we can keep buffers $\langle 7, 8\rangle$ and then $\langle 7, 9\rangle$. However, subsets of attributes generated by using the same σ will differ from each other. $\{W\}$ is not able to determine d within $\langle 7, 9\rangle$ although it was sufficient for $\langle 7, 8\rangle$. As a consequence, we need to restart from $B = A'$. We are allowed to remove T. Then, H turns out to be irreducible because of a need of keeping discernibility between objects 8 and 9. Finally, given the fact that H was not removed, W is not important any more, resulting in $S_9 = (\{H\}, \langle 7, 9\rangle)$.

6 Conclusions

In this paper, we attempted to establish better understanding of challenges and possibilities of searching for meaningful decision bireducts in data. We also outlined some examples of practical usage of decision bireducts in a new scenario of attribute subset selection in data streams. From this perspective, we need to remember that although decision bireducts were originally introduced in order to adopt some useful classifier ensemble principles, perhaps their major advantage lays in simple and flexible data-based knowledge representation.

In the nearest future, we intend to work on an enhanced interactive visualization of collections of decision bireducts, seeking for inspiration, e.g., in the areas of formal concept analysis [17] and visual bi-clustering [25]. We will also continue our studies on other types of bireducts, such as information bireducts which have been already successfully applied in [20]. Last but not least, following the research reported in [10], we are going to attempt to reconsider the discernibility-based bireduct construction criteria for the purposes of other rough set approaches, such as e.g. the dominance rough set model [26].

References

1. Świniarski, R.W., Skowron, A.: Rough Set Methods in Feature Selection and Recognition. Pattern Recognition Letters 24(6), 833–849 (2003)
2. Ślęzak, D.: Approximate Entropy Reducts. Fundamenta Informaticae 53(3-4), 365–390 (2002)
3. Abeel, T., Helleputte, T., de Peer, Y.V., Dupont, P., Saeys, Y.: Robust Biomarker Identification for Cancer Diagnosis with Ensemble Feature Selection Methods. Bioinformatics 26(3), 392–398 (2010)
4. Ślęzak, D.: Approximation Reducts in Decision Tables. In: Proc. of IPMU 1996, vol. 3, pp. 1159–1164 (1996)
5. Ślęzak, D., Janusz, A.: Ensembles of Bireducts: Towards Robust Classification and Simple Representation. In: Kim, T.-H., Adeli, H., Ślęzak, D., Sandnes, F.E., Song, X., Chung, K.-I., Arnett, K.P. (eds.) FGIT 2011. LNCS, vol. 7105, pp. 64–77. Springer, Heidelberg (2011)

6. Stawicki, S., Widz, S.: Decision Bireducts and Approximate Decision Reducts: Comparison of Two Approaches to Attribute Subset Ensemble Construction. In: Proc. of FedCSIS 2012, pp. 331–338. IEEE Computer Society (2012)
7. 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: New Developments in Knowledge Discovery in Information Systems. STUD FUZZ, vol. 56, pp. 49–88. Physica-Verlag, Heidelberg (2000)
8. Widz, S., Ślęzak, D.: Approximation Degrees in Decision Reduct-based MRI Segmentation. In: Proc. of FBIT 2007, pp. 431–436. IEEE Computer Society (2007)
9. Pawlak, Z., Skowron, A.: Rudiments of Rough Sets. Information Sciences 177(1), 3–27 (2007)
10. Mac Parthaláin, N., Jensen, R.: Simultaneous Feature And Instance Selection Using Fuzzy-Rough Bireducts. In: Proc. of FUZZ IEEE 2013 (2013)
11. Moshkov, M.J., Piliszczuk, M., Zielosko, B.: Partial Covers, Reducts and Decision Rules in Rough Sets – Theory and Applications. SCI, vol. 145. Springer, Heidelberg (2008)
12. Nguyen, S.H., Nguyen, H.S.: Pattern Extraction from Data. Fundamenta Informaticae 34(1-2), 129–144 (1998)
13. Ślęzak, D.: Normalized Decision Functions and Measures for Inconsistent Decision Tables Analysis. Fundamenta Informaticae 44(3), 291–319 (2000)
14. Sarawagi, S., Thomas, S., Agrawal, R.: Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications. Data Min. Knowl. Discov. 4(2/3), 89–125 (2000)
15. Kowalski, M., Stawicki, S.: SQL-based Heuristics for Selected KDD Tasks over Large Data Sets. In: Proc. of FedCSIS 2012, pp. 303–310. IEEE Computer Society (2012)
16. 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)
17. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer (1998)
18. Janusz, A., Ślęzak, D.: Utilization of Attribute Clustering Methods for Scalable Computation of Reducts from High-Dimensional Data. In: Proc. of FedCSIS 2012, pp. 295–302. IEEE Computer Society (2012)
19. Janusz, A., Stawicki, S.: Applications of Approximate Reducts to the Feature Selection Problem. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 45–50. Springer, Heidelberg (2011)
20. Janusz, A., Ślęzak, D., Nguyen, H.S.: Unsupervised Similarity Learning from Textual Data. Fundam. Inform. 119(3-4), 319–336 (2012)
21. Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K.: Learnability and the Vapnik-Chervonenkis Dimension. J. ACM 36(4), 929–965 (1989)
22. Aggarwal, C.C. (ed.): Data Streams – Models and Algorithms. Advances in Database Systems, vol. 31. Springer (2007)
23. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. In: Proc. of SIGMOD 1996, pp. 103–114. ACM Press (1996)
24. Ślęzak, D., Kowalski, M., Eastwood, V., Wróblewski, J.: Methods and Systems for Database Organization. US Patent 8,266,147 B2 (2012)

25. Havens, T.C., Bezdek, J.C.: A New Formulation of the coVAT Algorithm for Visual Assessment of Clustering Tendency in Rectangular Data. Int. J. Intell. Syst. 27(6), 590–612 (2012)
26. Słowiński, R., Greco, S., Matarazzo, B.: Dominance-Based Rough Set Approach to Reasoning About Ordinal Data. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 5–11. Springer, Heidelberg (2007)

Studies on the Necessary Data Size for Rule Induction by STRIM

Yuichi Kato[1], Tetsuro Saeki[2], and Shoutarou Mizuno[1]

[1] Shimane University,
1060 Nishikawatsu-cho, Matsue city, Shimane 690-8504, Japan
ykato@cis.shimane-u.ac.jp
[2] Yamaguchi University,
2-16-1 Tokiwadai, Ube city, Yamaguchi 755-8611, Japan
tsaeki@yamaguchi-u.ac.jp

Abstract. STRIM (Statistical Test Rule Induction Method) has been proposed as a method to effectively induct if-then rules from the decision table which is considered as a sample set obtained from the population of interest. Its usefulness has been confirmed by a simulation experiment specifying rules in advance, and by comparison with the conventional methods. However, there remains scope for future studies. One aspect which needs examination is determination of the size of the dataset needed for inducting true rules by simulation experiments, since finding statistically significant rules is the core of the method. This paper examines the theoretical necessary size of the dataset that STRIM needs to induct true rules with probability w [%] in connection with the rule length, and confirms the validity of this study by a simulation experiment at the rule length 2. The results provide useful guidelines for analyzing real-world datasets.

1 Introduction

Rough Sets theory as introduced by Pawlak [1] provides a database called the decision table, with various methods of inducting if-then rules and determining the structure of rating and/or knowledge in the database. Such rule induction methods are needed for disease diagnosis systems, discrimination problems, decision problems, and other aspects, and consequently many effective algorithms for rule induction by rough sets have been reported [2–7]. However, these methods and algorithms have paid little attention to mechanisms of generating the database, and have generally focused on logical analysis of the given database. This seems to narrow the scope of the analysis. In a previous study [8] we devised a model of data generation for the database, and proposed a statistical rule induction method and an algorithm named STRIM. In a simulation experiment based on the model of the data generation with if-then rules specified in advance, STRIM successfully inducted the specified true rules from different databases generated from the same specified rules [8]. In contrast, when conventional methods [4], [6], [7] were used, significant rules could barely be inducted.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 213–220, 2013.

Table 1. An example of a decision table

U	$C(1)$	$C(2)$	$C(3)$	$C(4)$	$C(5)$	$C(6)$	D
1	3	5	2	5	1	3	6
2	6	2	3	6	5	6	5
3	6	3	3	3	4	4	3
4	4	4	2	2	4	1	2
5	2	4	5	5	5	4	5
...	...	...	...	...	...	...	...
$N-1$	2	4	2	6	1	2	4
N	4	4	5	1	3	1	4

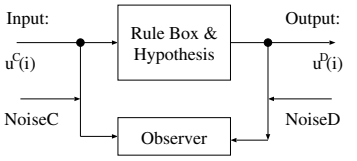


Fig. 1. Rough Sets system contaminated with noise: Input is a tuple of the condition attributes' value and its output is the decision attribute's value

Although the previous study [8] proposed a very effective and efficient method, several aspects required further studies. The first of these was to examine how many samples are needed for inducting true rules with high precision, depending on the rule induction problems, since the core point of STRIM was to find the rules of being statistically significant. This paper first summarizes the rule induction method by STRIM, and then focuses on the problem of the size of the dataset needed for STRIM to induct statistically significant rules. Specifically, this study derives an expression of evaluating the size of a dataset from STRIM. The expression can be used in two ways: the first is used for estimating how much probability w [%] STRIM can induct true rules for the size of a given dataset. The second is used for predicting the size of the dataset STRIM needs to induct true rules in the frame of a given decision table.

The validity of the expression is confirmed by a simulation experiment at the rule length 2. This study yields useful information for analyzing real-world datasets, since the conventional method can give no such guiding principle.

2 Data Generation Model and a Decision Table

Rough Sets theory is used for inducting if-then rules hidden in the decision table S. S is conventionally denoted $S = (U, A = C \cup \{D\}, V, \rho)$. Here, $U = \{u(i) | i = 1,$ ..., $|U| = N\}$ is a sample set, A is an attribute set, $C = \{C(j) | j = 1,$ ..., $|C|\}$ is a condition attribute set, $C(j)$ is a member of C and a condition attribute, and D is a decision attribute. V is a set of attribute values denoted by $V = \bigcup_{a \in A} V_a$ and is characterized by an information function ρ: $U \times A \rightarrow V$. Table 1 shows an example where $|C| = 6$, $|V_{a=C(j)}| = M_{C(j)} = 6$, $|V_{a=D}| = M_D = 6$, $\rho(x = u(1), a = C(1)) = 3$, $\rho(x = u(2), a = C(2)) = 2$ and so on.

Table 2. Hypothesis with regard to the decision attribute value

Hypothesis 1	$u^C(i)$ coincides with $R(d)$, and $u^D(i)$ is uniquely determined as $D = d$ (uniquely determined data).
Hypothesis 2	$u^C(i)$ does not coincide with any $R(d)$, and $u^D(i)$ can only be determined randomly (indifferent data).
Hypothesis 3	$u^C(i)$ coincides with several $R(d)$ ($d = d1, d2, ...$), and their outputs of $u^C(i)$ conflict with each other. Accordingly, the output of $u^C(i)$ must be randomly determined from the conflicted outputs (conflicted data).

STRIM considers the decision table to be a sample dataset obtained from an input-output system including a rule box, as shown in Fig. 1, and a hypothesis regarding the decision attribute values, as shown in Table 2. A sample $u(i)$ consists of its condition attributes values of $|C|$-tuple $u^C(i)$ and its decision attribute $u^D(i)$. $u^C(i)$ is the input into the rule box, and is transformed into the output $u^D(i)$ using the rules contained in the rule box and the hypothesis. For example, specify the following rules in the rule box:

$R(d)$: if R_d then $D = d$, ($d = 1, ..., M_D = 6$),

where R_d is a formula of the form $R_d = (C(1) = d) \wedge (C(2) = d) \vee (C(3) = d) \wedge (C(4) = d)$. Generate $u^C(i) = (v_{C(1)}(i), v_{C(2)}(i), ..., v_{C(|C|)}(i))$ of $u(i)$ by use of random numbers with a uniform distribution, and then $u^D(i)$ is determined using the rules specified in the rule box and the hypothesis.

In contrast, $u(i) = (u^C(i), u^D(i))$ is measured by an observer, as shown in Fig. 1. Existence of $NoiseC$ and $NoiseD$ makes missing values in $u^C(i)$, and changes $u^D(i)$ to create other values of $u^D(i)$, respectively. This model is closer to the real-world system. However, Table 1 shows an example generated by this specification without both noises for a plain explanation of the system. Inducting if-then rules from the decision table then identifies the rules in the rule box, by use of the observed inputs-output set $\{(u^C(i), u^D(i)) | i = 1, ..., |U| = N\}$.

3 Summaries of Rule Induction Procedures by STRIM

STRIM inducts if-then rules from the decision table through two processes, in separate stages. The first stage process is that of statistically discriminating and separating the set of indifferent data from the set of uniquely determined or conflicted data in the decision table (See Table 2). Specifically, assume $CP(k) = \bigwedge_j (C(j_k) = v_j)$ as the condition part of the if-then rule, and derive the set $U(CP(k)) = \{u(i) | u^C(i) \text{ satisfies } CP(k) \}$. Also derive $U(m) = \{u(i) | u^D(i) = m\}$ ($m = 1, ..., M_D$). For a set $U(CP(k))$, let us call distribution of decisions a tuple $f = (n_1, n_2, ..., n_{M_D})$, where $n_m = |U(CP(k)) \cap U(m)|$ for $m = 1, ..., M_D$. If the assumed $CP(k)$ does not satisfy the condition $U(R_d) \supseteq U(CP(k))$ (sufficient condition of specified rule R_d) or $U(CP(k)) \supseteq U(R_d)$ (necessary condition), $CP(k)$ only generates the indifferent dataset based on Hypothesis 2 in Table 2, and the distribution f does not have partiality of the distribution of decisions. Conversely, if $CP(k)$ satisfies either condition, f has partiality of the distribution, since $u^D(i)$ is determined by Hypothesis 1 or 3. Accordingly, whether f

Table 3. An example of a condition part and corresponding frequency of their decision attribute values

trying $CP(k)$	$C(1)$	$C(2)$	$C(3)$	$C(4)$	$C(5)$	$C(6)$	$(n_1, n_2, ..., n_6)$	z
1	1	0	0	0	0	0	(474, 246, 229, 246, 250, 238)	12.69
2	2	0	0	0	0	0	(247, 459, 213, 220, 223, 237)	12.95
3	0	0	0	0	0	5	(277, 268, 294, 258, 265, 261)	1.60
4	1	1	0	0	0	0	(240, 6, 4, 11, 6, 6)	31.67
5	1	2	0	0	0	0	(50, 48, 44, 51, 46, 49)	0.55
6	2	1	0	0	0	0	(45, 53, 51, 50, 34, 51)	0.98
7	2	2	0	0	0	0	(8, 260, 4, 7, 5, 6)	33.43
8	1	1	0	4	0	0	(52, 1, 1, 2, 1, 2)	14.90
9	1	1	0	5	0	0	(41, 1, 0, 0, 0, 1)	14.05
10	2	1	0	0	0	6	(8, 6, 5, 8, 8, 11)	1.52
11	2	2	1	0	0	0	(2, 51, 1, 1, 0, 0)	15.31
12	2	2	2	0	0	0	(0, 49, 0, 0, 0, 0)	15.84
13	3	3	0	6	0	0	(2, 0, 43, 2, 3, 2)	12.96
14	0	0	4	5	5	0	(12, 4, 11, 10, 5, 7)	1.66
15	0	0	4	5	0	1	(9, 6, 3, 9, 9, 11)	1.44
16	0	0	4	5	0	3	(7, 6, 8, 10, 9, 10)	0.82
17	0	0	4	5	0	6	(7, 11, 16, 10, 3, 10)	2.49
18	0	0	4	6	3	0	(9, 10, 8, 7, 6, 4)	1.28
19	0	0	4	6	4	0	(7, 5, 10, 6, 7, 12)	1.83
20	0	0	4	6	6	0	(7, 8, 5, 9, 8, 8)	0.80
21	0	0	0	6	6	3	(5, 9, 9, 7, 3, 12)	2.00

has the partiality or not determines whether the assumed $CP(k)$ is neither a necessary nor sufficient condition. Whether f has the partiality or not can be determined objectively by statistical test of the following null hypothesis $H0$ and its alternative hypothesis $H1$:

$H0$: f does not have partiality of the distribution of decisions. $H1$: f has partiality of the distribution of decisions.

Table 3 shows the number of examples of $CP(k)$, $(n_1, n_2, ..., n_{M_D})$ and an index of the partiality by z derived from Table 1 with $N = 10000$, in order to illustrate this concept. For example, the first row means: 100000 denotes $CP(k = 1) = (C(1) = 1)$ (the rule length is $RL = 1$) and its corresponding $f = (474, 246, 229, 246, 250, 238)$ and $z = 12.69$, where

$$z = \frac{n_d + 0.5 - np_d}{(np_d(1 - p_d))^{0.5}}, \tag{1}$$

$n_d = \max(n_1, n_2, ..., n_{M_D})$, $(d \in \{1, 2, ..., M_D = 6\})$, $p_d = P(D = d)$, $n = \sum_{m=1}^{M_D} n_m = |CP(k)|$. In principle, $(n_1, n_2, ..., n_{M_D})$ under $H0$ obeys a multinomial distribution which is sufficiently approximated by the standard normal distribution by use of n_d under the testing condition: $p_d n \geq 5$ and $n(1 - p_d) \geq 5$ [9]. In the same way, the fourth row 110000 denotes $CP(k = 4) = (C(1) = 1) \wedge (C(2) = 1)$ $(RL = 2)$, the eighth 110400 is $(C(1) = 1) \wedge (C(2) = 1) \wedge (C(4) = 4)$ $(RL = 3)$, and so on. Here, if we specify a standard of the significance level such as $z \geq z_\alpha = 3.0$ and reject $H0$, then the assumed $CP(k)$ becomes a candidate for the rules in the rule box.

```
int main(void) {
  int rule[|C|]={0,...,0}; //initialize trying rules
  int tail=-1; //initial value set
  input data; // set decision table
  rule_check(tail,rule); // Stage 1
  make Pyramid(l) (l=1,2,...) so that every r(k) belongs to one Pyramid at least;
  // Stage 2, r(k): rule candidate
  make rePyramid(l) (l=1,2,...); // Stage 2
  reduce rePyramid; // Stage 2
} // end of main

int rule_check(int tail,int rule[|C|]) { // Stage 1
  for (ci=tail+1; cj<|C|; ci++) {
    for (cj=1; cj<=|C[ci]|; cj++) {
      rule[ci]=cj; // a trying rule sets for test
      count frequency of the trying rule; // count n1, n2, ...
      if (frequency>=N0) { //sufficient frequency ?
        if (|z|>3.0) { //sufficient evidence ?
          store necessary data such as rule, frequency of n1 and n2, and z
        } // end of if |z|
        rule_check(ci,rule);
      } // end of if frequency
    } // end of for cj
    rule[ci]=0; // trying rules reset
  } // end of for ci
} // end of rule_check
```

Fig. 2. An algorithm for STRIM (Statistical Test Rule Induction Method)

The second stage process is that of arranging the set of rule candidates derived from the first process, and finally estimating the rules in the rule box, since some candidates may satisfy the relationship: $CP(ki) \subseteq CP(kj) \subseteq CP(kl) \cdots$. For example, in the case 100000 $\supset$ 110000 $\supset$ 110400 (see Table 3). The basic concept is to represent the $CP(k)$ of the maximum z, that is, the maximum partiality. In the above example, STRIM selects the $CP(k)$ of 110000, which by chance coincides with the rule specified in advance. Figure 2 shows the STRIM algorithm [8].

Table 4 shows the estimated results for Table 1 with $N = 10000$. STRIM inducts all of twelve rules specified in advance, and also twenty-two extra rules ($R(i)$ ($i =$ 13, ..., 34); $R(i)$ ($i = 16$, ..., 32) are omitted due to limited space). However, there are clear differences between them in the indexes of accuracy and coverage.

4 Remarks on Testing Conditions

As described in Section 3, the dataset applicable to STRIM must satisfy the testing condition: $p_d n \geq 5$ and $n(1 - p_d) \geq 5$. The least number satisfying the condition is denoted with N_0, and then consider the following event with a given probability w:

$$P(n \geq N_0) = P(z \geq z_0) = w \tag{2}$$

Here, $z = \dfrac{n + 0.5 - Np_c}{\sqrt{Np_c(1-p_c)}}$, $z_0 = \dfrac{N_0 + 0.5 - Np_c}{\sqrt{Np_c(1-p_c)}}$. $p_c = P(C = CP(k)) = \prod_j P(C(j_k) = v_k)$ is the outcome probability of $CP(k)$ in the decision table.

Table 4. Results of estimated rules for the decision table in Table 1 (without noise) by STRIM

estimated $R(i)$	$C(1)$	$C(2)$	$C(3)$	$C(4)$	$C(5)$	$C(6)$	D	$(n_1, ..., n_6)$	p-value (z)	accuracy	coverage
1	5	5	0	0	0	0	5	(7, 6, 2, 5, 258, 3)	0(33.88)	0.918	0.156
2	0	0	2	2	0	0	2	(6, 274, 7, 10, 8, 6)	0(38.88)	0.881	0.162
3	0	0	4	4	0	0	4	(7, 3, 4, 259, 6, 8)	0(33.53)	0.902	0.153
4	2	2	0	0	0	0	2	(8, 260, 4, 7, 5, 6)	0(33.43)	0.897	0.154
5	6	6	0	0	0	0	6	(4, 2, 3, 6, 8, 251)	0(33.37)	0.916	0.150
6	0	0	1	1	0	0	1	(240, 5, 3, 4, 6, 2)	0(32.81)	0.923	0.144
7	3	3	0	0	0	0	3	(5, 7, 250, 7, 7, 5)	0(32.60)	0.890	0.153
8	5	5	0	0	0	0	5	(11, 3, 3, 10, 246, 5)	0(32.21)	0.885	0.149
9	4	4	0	0	0	0	4	(10, 8, 5, 242, 5, 5)	0(31.82)	0.880	0.144
10	0	0	3	3	0	0	3	(5, 4, 238, 9, 6, 6)	0(31.77)	0.888	0.146
11	1	1	0	0	0	0	1	(240, 6, 4, 11, 6, 6)	0(31.67)	0.879	0.144
12	0	0	6	6	0	0	6	(12, 3, 5, 8, 8, 239)	0(31.34)	0.869	0.144
13	0	0	5	4	6	0	1	(15, 2, 8, 5, 4, 4)	3.3e-5(3.99)	0.395	0.009
14	3	5	0	0	1	0	4	(3, 5, 5, 15, 3, 10)	1.41e-4(3.63)	0.366	0.009
15	3	1	0	0	0	2	4	(9, 9, 5, 17, 5, 4)	1.72e-4(3.58)	0.347	0.010
...	...	...	...	...	...	...	...	...	...	...	...
33	0	0	1	4	0	5	3	(5, 6, 14, 7, 3, 8)	1.35e-3(3.00)	0.326	0.009
34	0	0	5	4	1	0	3	(8, 4, 15, 9, 8, 3)	1.35e-3(3.00)	0.319	0.009

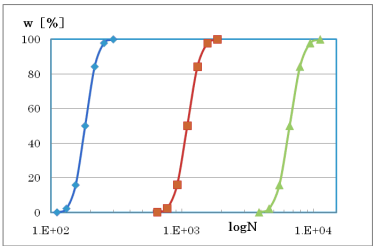


Fig. 3. Theoretical $N(w, RL)$ evaluated by (3) at $w = 0.1$ [%] ($z_0 = 3.0$), $= 2.3$ [%] ($z_0 = 2.0$), ..., $= 99.9$ [%] ($z_0 = -3.0$) (♦: $RL = 1$, ■: $RL = 2$, ▲: $RL = 3$)

For example, if $CP(K) = (C(1) = 1) \wedge (C(2) = 1)$ $(RL = 2)$ then $p_c = P(C(1) = 1) \cdot P(C(2) = 1)$. Assuming that z obeys the standard normal distribution, z_0 is explicitly determined, and the least N denoted with N_{lst} satisfying (2) is given by:

$$N_{lst} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{3}$$

where $+$: $z_0 \leq 0$, $-$: $z_0 > 0$, $a = p_c^2$, $b = -\{(2p_c(N_0 + 0.5) + z_0^2 p_c(1 - p_c)\}$ and $c = (N_0 + 0.5)^2$.

Accordingly, N_{lst} in (3) is mainly determined by parameters w and RL. So let us denote N_{lst} in (3) with $N_{lst}(w, RL)$.

Figure 3 shows $N(w, RL)$ evaluated by (3) at $w = 0.1$ [%] ($z_0 = 3.0$), $= 2.3$ [%] ($z_0 = 2.0$), $= 15.9$ [%] ($z_0 = 1.0$), $= 50.0$ [%] ($z_0 = 0.0$), $= 84.1$ [%] ($z_0 = -1.0$), $= 97.7$ [%] ($z_0 = -2.0$), $= 99.9$ [%] ($z_0 = -3.0$) every $RL = 1$, 2 and 3

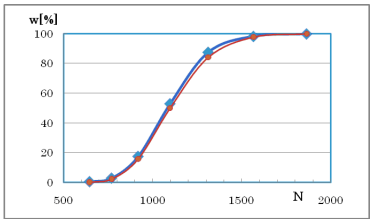


Fig. 4. Comparison of $N(w, RL = 2)$ between theoretical and experimental values at w = 0.1 [%] ($z_0 = 3.0$), = 2.3 [%] ($z_0 = 2.0$), ..., = 99.9 [%] ($z_0 = -3.0$)) (⧫: experimental value, •: theoretical value)

in the specification of Section 2. where $P(C(j) = v_k) = 1/6$ (for each $j = 1$, ..., $|C| = 6$). For example, Fig. 3 yields the following useful information:

1) Supposing $RL = 2$, $N = 1865$ at least is needed to induct true rules with the probability of almost $w = 100$ [%]. This meaning is denoted with 1865 $= N_{lst}$ ($w = 100$ [%], $RL = 2$).
2) If a dataset of $N = 1000$ is given, then the probability of inducting the true rules with $RL = 2$ is estimated to be about $w = 30$ [%]. This meaning is denoted with 30 [%] $= w = N_{lst}^{-1}$ ($N_{gvn} = 1000$, $RL = 2$).

To confirm the consideration outlined in this section, a simulation experiment was conducted using the decision table in Section 2, and the following procedures:

Step 1: Randomly select samples by $N(w, RL = 2)$ from the decision table (N = 10000) in Section 2, and make a new decision table.
Step 2: Apply STRIM to the new table, and count the number of inducted true rules specified in advance.
Step 3: Repeat Step 1 and Step 2 Nr times.
Step 4: Calculate the rate of true rules inducted out of Nr trials.

Figure 4 shows the comparison of $N(w, RL = 2)$ ($w = 0.1$ [%] ($z_0 = 3.0$), = 2.3 [%] ($z_0 = 2.0$), ..., = 99.9 [%] ($z_0 = -3.0$)) between theoretical values studied in this section, and the experimental values obtained from the above procedures by $Nr = 100$. The experimental value adequately represents the theoretical value, and confirms the validity of the theoretical considerations.

5 Conclusions

The basic concept of STRIM is that rules make partiality, and finding the partiality leads to finding the rule. After specifically summarizing the basic notion of STRIM, this paper focused on the problem of the size of the dataset needed for STRIM to statistically determine the partiality, i.e., to statistically induct true rules. This problem was previously identified as needing future work [8]. We then theoretically derived the dataset size as $N(w, RL)$, which directly depends

on the outcome probability of the condition part of the rule candidate; the validity was confirmed by a simulation experiment. The notion $N(w, RL)$ is highly though-provoking, since it can be used as $N_{lst}(w, RL)$ and/or $N_{lst}^{-1}(N_{gvn}, RL)$. Accordingly, $N(w, RL)$ seems to be useful when analyzing real-world datasets. This is one of the major advantages of STRIM. In future studies for analyzing real-world datasets based on $N(w, RL)$, STRIM should be applied to missing data sets such as the studies in reference [10], and/or to datasets including the type of noise shown in Fig. 1. The capacity of STRIM for rule induction must also be investigated in simulation experiments.

References

1. Pawlak, Z.: Rough sets. Internat. J. Inform. Comput. Sci. 11, 341–356 (1982)
2. Skowron, A., Rauszer, C.: The discernibility matrix and functions in information systems. In: Slowiński, R. (ed.) Intelligent Decision Support — Handbook of Application and Advances of Rough Set Theory, pp. 331–362. Kluwer Academic Publisher, Dordrecht (1992)
3. Bao, Y.G., Du, X.Y., Deng, M.G., Ishii, N.: An efficient method for computing all reducts. Transactions of the Japanese Society for Artificial Intelligence 19, 166–173 (2004)
4. Grzymała-Busse, J.W.: LERS — A system for learning from examples based on rough sets. In: Slowiński, R. (ed.) Intelligent Decision Support — Handbook of Applications and Advances of the Rough Sets Theory, pp. 3–18. Kluwer Academic Publisher, Dordrecht (1992)
5. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Science 46, 39–59 (1993)
6. Shan, N., Ziarko, W.: Data-based acquisition and incremental modification of classification rules. Computational Intelligence 11, 357–370 (1995)
7. Nishimura, T., Kato, Y., Saeki, T.: Studies on an effective algorithm to reduce the decision matrix. In: Kuznetsov, S.O., Ślęzak, D., Hepting, D.H., Mirkin, B.G. (eds.) RSFDGrC 2011. LNCS, vol. 6743, pp. 240–243. Springer, Heidelberg (2011)
8. Matsubayashi, T., Kato, Y., Saeki, T.: A new rule induction method from a decision table using a statistical test. 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. 81–90. Springer, Heidelberg (2012)
9. Walpole, R.E., Myers, R.H., Myers, S.L., Ye, K.: Probability and Statistics for Engineers and Scientists, 8th edn., pp. 187–191. Pearson Prentice Hall, New Jersey (2007)
10. Grzymała-Busse, J.W., Grzymała-Busse, W.J.: Handling missing attribute values. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, 2nd edn., pp. 33–49. Springer, Heidelberg (2010)

Applying Threshold SMOTE Algorithm with Attribute Bagging to Imbalanced Datasets

Jin Wang, Bo Yun, Pingli Huang, and Yu-Ao Liu

Chongqing Key Laboratory of Computational Intelligence
Chongqing University of Posts and Telecommunications
Chongqing, China
wangjin@cqupt.edu.cn, adainu@yeah.net, huang_ping_li@126.com, yuao_liu@yahoo.com.cn

Abstract. Synthetic minority over-sampling technique (SMOTE) is an effective over-sampling technique and specifically designed for learning from imbalanced data sets. However, in the process of synthetic sample generation, SMOTE is of some blindness. This paper proposes a novel approach for imbalanced problem, based on a combination of the Threshold SMOTE (TSMOTE) and the Attribute Bagging (AB) algorithms. TSMOTE takes full advantage of majority samples to adjust the neighbor selective strategy of SMOTE in order to control the quality of the new sample. Attribute Bagging, a famous ensemble learning algorithm, is also used to improve the predictive power of the classifier. A comprehensive suite of experiments tested on 7 imbalanced data sets collected from UCI machine learning repository is conducted. Experimental results show that TSMOTE-AB outperforms the SMOTE and other previously known algorithms.

Keywords: Imbalanced classification, SMOTE, Threshold SMOTE, Attribute Bagging, Ensemble learning, Over-sampling.

1 Introduction

There are many real-world applications where the data sets are highly imbalanced, such as credit card fraud detection [1], oil spill detection from satellite images [2], medical diagnosis [3], or face detection [4], et al. In these data sets, there are many examples of the negative (majority) class, and very few examples of the positive (minority) class. But often it is the rare occurrence, the positive (minority) class, that is our interest and of great importance. In data mining, most traditional learning systems are designed to work on balanced data sets and focusing on improving overall performance, but usually perform poorly on the minority class.

In recent years, a large amount of techniques have been developed trying to address the problem. These proposals can be categorized into two groups, the algorithm level approaches [5][6] and the data level techniques [7-9]. In general, data level learning approaches are more versatile than algorithm level approaches. Synthetic minority over-sampling technique (SMOTE) [8], which inserts synthetic data into the original data set to increase the number of minority examples, is one well-known technique in data level.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 221–228, 2013.

In addition to those two level approaches, another group of techniques emerges when the use of ensemble of classifier is considered. Ensemble methods [10][11] are well known in machine learning area. Boosting [12] and Bagging [13] are the most common ensemble learning algorithms among the ensemble methods. There are many variants and other different approaches [14].

In this paper, we propose a novel approach---Threshold SMOTE with Attribute Bagging (TSMOTE-AB), improving the selection of examples in SMOTE and combined with Attribute Bagging [15] ensemble algorithm to improve the results of classification performance.

The paper is organized as follows: Section 2 discusses related work of SMOTE and Bagging in imbalanced data sets problem. Section 3 describes our improved algorithm – TSMOTE-AB. Section 4 gives observations from experiments and analyzes experimental results. Finally, section 5 presents the conclusions.

2 Related Work

2.1 SMOTE

SMOTE algorithm is proposed by Chawla et al. [8]. The main idea in SMOTE is to generate new synthetic minority examples by interpolating between a minority sample and its near neighbors. Synthetic examples are introduced along the line segment between each minority class example and one of its k-minority nearest neighbors. Let the training set S contain examples from the minority class P and the majority class N. For each sample $x \in P$, search its k-nearest neighbors in P, and choose one of them randomly as x_1. Then, multiply the difference between the two vectors by a random number and add it to x, a liner interpolation is fulfilled randomly to produce a new sample called y. The formula is shown as follows:

$$y = x + rand(0,1) \times (x_1 - x)$$

Where the rand(0,1) means a random number between 0 and 1.

Although SMOTE has been proved to be successful it also has an obviously shortcoming. The over-generalization problem as it blindly generalizes the regions of the minority class without regard to the majority class. This strategy is particularly problematic in the case of skewed class distribution where the minority class is very sparse with respect to the majority class. In this condition, SMOTE generation of synthetic examples may increase the occurrence of overlapping between classes [16].

2.2 Attribute Bagging

Attribute Bagging (AB) algorithm is proposed by Robert Bryll et al [15] based on Bagging [13]. It establishes an appropriate size of the attribute subsets with a wrapper method and then randomly selects subsets of features, creating projections of the training set on which the ensemble classifiers are built. The induced classifiers are then used for voting. This method was used for handling configuration recognition, and it was found that bagging the attributes of strong generalization power improved the performance of the resulting ensemble.

Attribute Bagging is a wrapper method that can be used with any learning algorithm. It can improve the performance of prediction by using different attribute subsets to classify. The experiment in [15] shows that AB can gives consistently better results than Bagging, both in accuracy and stability.

3 Threshold SMOTE with Attribute Bagging

3.1 TSMOTE

In Fig. 1, we use SMOTE to generate the new sample for sample x_1. Firstly, we find K-nearest neighbors (K=5) of sample $x_1 \in P$ by comparing the distance between samples also in P, which is calculated with the Euclidean distance metric for numerical features. Unlike negative samples, the distribution of the positive samples is very sparse, so there is a far distance between x_1 and $\{x_4 \sim x_6\}$. If we use these three samples as the neighbor to generate the new samples as A, B and C, they will not only reduce the classification accuracy, but also confuse the classification of the positive samples because they are mixed with the negative samples.

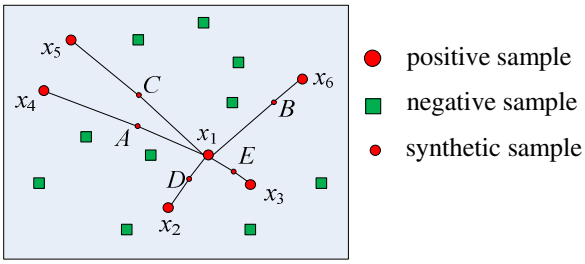


Fig. 1. Influence of incorrect neighbor on SMOTE algorithm

For the above reason, we use the Threshold SMOTE algorithm to generate the new samples. In training set $S = (x_i, y_i)$, i=1, 2, ···, M, where x_i is a vector of attributes and $y_i \in \{-1,1\}$ is the associated observed class label. For each sample $x_i \in P$, its K-positive nearest neighbors (K=5) set and K-negative nearest neighbors (K=5) set are:

$$NE\_P_i = \{ne\_p_{ik} \mid k = 1, \ldots, K, ne\_p_{ik} \in P\} \qquad NE\_N_i = \{ne\_n_{ik} \mid k = 1, \ldots, K, ne\_n_{ik} \in N\}$$

its candidate neighbor set is $CAND_i$, which set samples in will synthesize samples with x_i; $d(i, k)$ denotes the distance between x_i and ne_p_{ik} The detail steps as follow:

(1). Randomly choose a sample $x_i \in P$, find its NE_P_i and NE_N_i.

(2). Calculated $threshold = \dfrac{\sum_{i=1}^{N} \sum_{k=1}^{K} d(i,k)}{K \times N}$, where N is the number of positive samples, K is the number of neighbors.

(3). Sequentially select a ne_p_{ik} or ne_n_{ik} as $cand_{ik}$ according to followed formula.

$$cand_{ik} = \begin{cases} ne\_p_{ik}, if\ d(i,k) < threshold \\ ne\_n_{ik}, else \end{cases}$$

(4). Repeat the steps (2) and (3) until the $CAND_i$ is full (its capacity is K).

(5). Generate synthetic samples x_{ik} and add them into P until the number of P equals the number of N.

$$x_{ik} = \begin{cases} x_i + rand(0,1) \times (cand_{ik} - x_i), & cand_{ik} \in NE\_P_i \\ x_i + rand(0,0.5) \times (cand_{ik} - x_i), & cand_{ik} \in NE\_N_i \end{cases}$$

These synthetic samples would not be used for generate new samples.

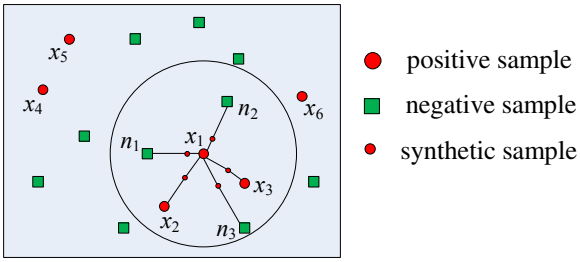


Fig. 2. Fundamental of ASMOTE algorithm

When these five steps process over, we gain a new training set and then use this new set to train the weak learner. Fig.2. shows the fundamental of TSMOTE algorithm. As we can see, when the K nearest neighbors $\{x_2 \sim x_6\}$ of x_1 are found, the nearer negative samples $\{n_1 \sim n_3\}$ were chosen to replace the far positive samples $\{x_4 \sim x_6\}$ to generate synthetic samples. In this way, TSOMTE avoid the confusion by mixing samples and take the advantage of the class information in data set which improves the quality of synthetic samples.

3.2 Attribute Bagging

Algorithm Attribute Bagging

1. **Input:** S' : Training data set; T: Number of iterations; I: Weak learner (KNN); M: Attribute set of data
2. **Output:** Attribute bagged classifier:
   $H(x) = sign\left(\sum_{t=1}^{T} h_t(x)\right)$, where $h_t \in [-1,1]$ are the induced classifiers.
3. **For** $i=0:T$ **repeat**
   a) Randomly select j features in M as a feature subset M_t. The features in subset can be repetitively selected.
   b) For each sample in S', only select the features in M_t to get the new training sets S_t'
   c) Use S_t' for KNN algorithm to gain a classifier h_t

   End For

After processing TSMOTE algorithm on original training set, a new training set S' is obtained. Then the weak classifiers are trained with S' by using attribute bagging. Breiman pointed out that the stability was the key factor to improve the accuracy of

prediction. If there is a little change on data set which can cause an obvious difference on the classification result, that is called instability. For instable learning algorithms, the accuracy of prediction will be improved by using bagging, while the effect will not be obvious for stable learning algorithms, sometimes even decrease it. Langley's research shows that the property of KNN classifier is sensitive to the number of features [17], so KNN is unstable for attribute sampling. Therefore, we can use attributes resampling to get different training samples and the KNN as the weak learner, so as to enhance the performance of KNN. The detail of attribute bagging is shown as above.

We set the number of iterations T=10. An important key to this algorithm is j, the number of the selected features. An appropriate attribute subset size is found by testing classification accuracy of variously sized random subsets of attributes.

4 Experiments and Results

4.1 Performance Evaluation

In this paper, we focus on two-class imbalanced data sets. A *confusion matrix*, as a key factor in the assessment, shown in Table 1, is used to formulate the representation of classification performance.

Table 1. Confusion Matrix

	Positive prediction	**Negative prediction**
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

From Table 1., a few kinds of new evaluations for imbalanced problem, recall [18], precision [19], *F-value* [6] and *G-mean* [20] defined as follow:

$$precision = TP/(TP + FP) \qquad recall = TP/(TP + FN)$$

$$F\text{ - }Value = \frac{2 \times precision \times recall}{precision + recall} \qquad G\text{ - }Mean = \sqrt{precision \times recall}$$

Besides, the AUC-ROC [21] can also be used to indicate the performance of the learning algorithm on the minority class. In this paper, we used *F-Value*, *G-Mean* and AUC-ROC to judge the performance.

4.2 Data Sets and Results

Seven data sets of UCI are used in our empirical studies. As shown in Table 2, all of them are highly imbalanced. For all the seven data sets, 10-fold cross validation is used. We compared SMOTE with our TSMOTE algorithm and the averaged results of 50 runs are shown in Table 3. The weak learner in all experiments is KNN (k=5) classifier.

After used the TSMOTE in all the datasets, we then use the AB algorithm for ensemble. The number of the selected features j must adjust by different datasets. We set j = 50%~100% of features number in each dataset and the result is shown in Fig.3.

Table 2. Characteristics of Data Sets

Dataset	Samples	Positive Samples	Features	Validation Method
Abalone(9 vs. 18)	731	42(5.75%)	8	10-fold CV
Ionosphere(bad vs. good)	351	126(35.9%)	33	10-fold CV
Vehicle(van vs. all)	846	212(25.1%)	18	10-fold CV
Satimage(4 vs. all)	6435	626(9.73%)	36	10-fold CV
Phoneme(1 vs. 0)	5404	1586(29.3)	5	10-fold CV
German(bad vs. good)	1000	300(30%)	24	10-fold CV
Yeast(CYT vs. POX)	483	20(4.14%)	8	10-fold CV

Table 3. The result of TSMOTE and SMOTE

Dataset	TSMOTE			SMOTE		
	F-Value	G-Mean	AUC-ROC	F-Value	G-Mean	AUC-ROC
Abalone	0.583	0.74	0.926	0.541	0.717	0.628
Ionosphere	0.813	0.866	0.897	0.711	0.764	0.821
Vehicle	0.928	0.953	0.994	0.913	0.951	0.919
Satimage	0.613	0.762	0.915	0.533	0.747	0.897
Phoneme	0.801	0.807	0.883	0.79	0.79	0.872
German	0.531	0.648	0.723	0.519	0.642	0.714
Yeast	0.591	0.762	0.825	0.576	0.703	0.834

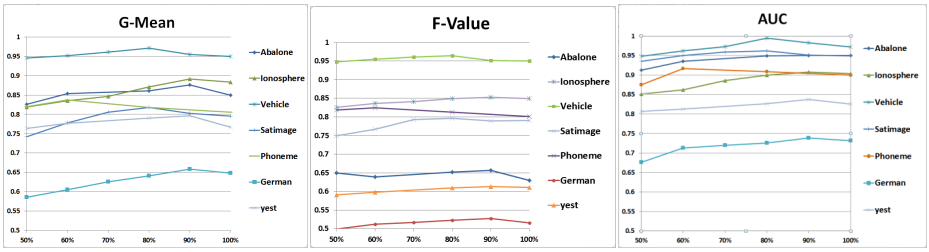


Fig. 3. Result of different number of the selected features *j*

And we also compared the AUC result of TSMOTE-AB with some other algorithms. The results are shown in Table 4., and the best result is marked with bold. From Table 3., we can find that TSMOTE achieved an obviously higher prediction then the original SMOTE in all the data sets. Table 4. shows that TSOMTE results are generally better than other algorithms.

Table 4. The AUC of TSMOTE and other algorithms

Dataset	TSMOTE-AB	Bagging	SMOTE-Bagging	AdaBagging	MSMOTE Bagging	AdaCost	SMOTE-Boost	RAMO-Boost
Abalone	**0.95**	0.605	0.753	0.634	0.716	0.924	0.766	0.976
Ionosphere	**0.904**	0.864	0.885	0.891	0.886	0.882	0.889	0.901
Vehicle	**0.995**	0.955	0.965	0.951	0.949	**0.995**	0.965	**0.995**
Satimage	**0.962**	0.946	0.948	0.939	0.936	0.933	0.947	0.949
Phoneme	**0.917**	0.881	0.889	0.903	0.897	0.894	0.894	0.906
German	0.739	0.716	0.724	0.731	0.737	0.713	0.734	**0.741**
Yeast	**0.837**	0.525	0.788	0.699	0.774	0.816	0.740	0.745

5 Conclusion

A TSMOTE-AB based scheme is proposed for learning from imbalanced data sets. Experimental result on seven UCI data sets shows that the proposed method can improve predictive performance of the learned classifier. The TSMOTE increases the number of the positive randomly by using both positive and negative samples to generate the synthetic samples through a threshold to overcome the shortcomings of original SMOTE which only uses positive samples and ignores the real distribution of data sets.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (61203308, 61075019).

References

1. Chan, P., Stolfo, S.J.: Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In: 4th International Conference on Knowledge Discovery and Data Mining, pp. 164–168. AAAI Press (1998)
2. Kubat, M., Holte, R.C., Matwin, S., Kohavi, R., Provost, F.: Machine learning for the detection of oil spills in satellite radar images. Machine Learning, 195–215 (1998)
3. Mazurowski, M.A., Habas, P.A., Zurada, J.M., Lo, J.Y., Baker, J.A., Tourassi, G.D.: Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural Netw. 21(2-3), 427–436 (2008)
4. Liu, Y.H., Chen, Y.T.: Total margin-based adaptive fuzzy support vector machines for multiview face recognition. In: Proc. IEEE Int. Conf. Syst., Man Cybern., vol. 2, pp. 1704–1711 (2005)
5. Zadrozny, B., Elkan, C.: Learning and making decisions when costs and probabilities are both unknown. In: 7th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, New York, pp. 204–213 (2001)
6. Wu, G., Chang, E.: KBA: kernel boundary alignment considering imbalanced data distribution. IEEE Trans. Knowl. Data Eng. 17(6), 786–795 (2005)

7. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Expl. Newslett. 6, 20–29 (2004)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357 (2002)
9. Chawla, N.V., Japkowicz, N., Kolcz, A.(eds.): Special Issue Learning Imbalanced Datasets. SIGKDD Explor. Newsl. 6(1) (2004)
10. Polikar, R.: Ensemble based systems in decision making. IEEE Circuits Syst. Mag. 6(3), 21–45 (2006)
11. Rokach, L.: Ensemble-based classifiers. Artif. Intell. Rev. 33, 1–39 (2010)
12. Freund, Y., Schapire, R.: Experiments with a New Boosting Algorithm. In: 13th International Conference on Machine Learning, pp. 325–332 (1996)
13. Breiman, L.: Bagging predictors. Mach. Learning 24, 123–140 (1996)
14. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, New York (2004)
15. Bryll, R., Gutierrez-Osuna, R., Quek, F.: Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. Pattern Recognition 36(6), 1291–1302 (2003)
16. Wang, B.X., Japkowicz, N.: Imbalanced Data Set Learning with Synthetic Samples. In: Proc. IRIS Machine Learning Workshop (2004)
17. Langley, P., Iba, W.: Average-case analysis of nearest neighbor algorithm. In: 13th International Joint Conference on Artificial Intelligence, pp. 889–894. Morgan Kaufmann Publishers, San Francisco (1993)
18. Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004)
19. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining Knowl. Discovery 2(2), 121–167 (1998)
20. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
21. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30(7), 1145–1159 (1997)

Parallel Reducts: A Hashing Approach

Minghua Pei[1], Dayong Deng[1], and Houkuan Huang[2]

[1] College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua, PR China, 321004
{pmh427,dayongd}@163.com
[2] School of Computer and Information Technology, Beijing Jiaotong University, Beijing, PR China, 100044
hkhuang@center.bjtu.edu.cn

Abstract. A hashing approach in parallel reducts is clearly presented in this paper. With the help of this new approach, time-consuming comparison operations reduce significantly, therefore, matrix of attribute significance can be calculated more efficiently. Experiments show that our method has advantage over PRMAS, our classical parallel reducts method.

Keywords: Rough sets, Hashing approach, Parallel reducts. Matrix of attribute significance.

1 Introduction

Rough set[1, 2] theory is an effective mathematical tool, dealing with imprecise, vague and incomplete information. It has been widely used in classification and feature selection(also called attribute reducts) in data mining. Concrete ways to obtain attribute reducts include discernibility matrix and function[3–5], information entropy [6, 7] and attribute significance etc. The method of attribute significance is an efficient approach to obtain condition attribute reducts.

To deal with incremental data, dynamic data and tremendously large data, various models of attribute reducts are constructed, such as dynamic reducts[8, 9] and parallel reducts[10–15].

The problem to obtain dynamic reducts is NP-hard, and the ways to obtain dynamic reducts are incomplete because the intersection of all the Pawlak reducts in a series of decision subsystems may be empty.

Parallel reducts[10–15] extend Pawlak reducts and dynamic reducts. They have all the advantages of dynamic reducts, and have excellent performance enough to match that of the best algorithm for Pawlak reducts. In [14] a matrix of attribute significance is proposed, and it could obtain both parallel reducts and dynamic reducts.

In parallel reducts, classification takes a great deal of time. Meanwhile, comparison operations dominate classification process. We here present a hashing approach to improve classification. D. E. Knuth[16] credited H. P. Luhn (1953) for inventing hash tables, along with the chaining method for resolving collisions. P. C. Wang[17] employed hash method to generate approximate decision rules.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 229–239, 2013.

Now, we continue to review several basic concepts in parallel reducts and F-attribute significance. Next we propose an algorithm to obtain parallel reducts in a hashing approach. Experimental results show that our new approach has a significant performance boost in attribute reducts.

2 Rough Sets

Readers are assumed to be familiar with rough set theory. So we only introduce some primary knowledge of rough sets briefly.

Let $DS = (U, A, d)$ be a decision system, where $\{d\} \cap A = \emptyset$, the decision attribute d divides the universe U into parts, denoted by $U/d = \{Y_1, Y_2, ...Y_p\}$, where Y_i is an equivalence class. The positive region is defined as

$$POS_A(d) = \bigcup_{Y_i \in U/d} POS_A(Y_i) \tag{1}$$

Sometimes the positive region $POS_A(d)$ is also denoted by $POS_A(DS, d)$. In rough set theory, the most popular definition of reduct is Pawlak reduct(reduct in short) in a decision system. It could be shown as below:

Definition 1. *Let $DS = (U, A, d)$ be a decision system, $B \subseteq A$ is called a reduct of the decision system DS iff B satisfies two conditions:*

1. $POS_B(d) = POS_A(d)$,
2. *For any* $S \subset B, POS_S(d) \neq POS_A(d)$

All reducts of a decision system DS is denoted by $RED(DS)$.

Definition 2. *In a decision system $DS = (U, A, d)$ we will say d depends on A to a degree $h(0 \leq h \leq 1)$, if*

$$h = \gamma(A, d) = \frac{|POS_A(d)|}{|U|} \tag{2}$$

Where $|.|$ denotes the cardinality of a set.

Definition 3. *The significance of an attribute a in a decision system $DS = (U, A, d)$ is defined by*

$$\sigma(a) = \frac{\gamma(A, d) - \gamma(A - \{a\}, d)}{\gamma(A, d)} = 1 - \frac{\gamma(A - \{a\}, d))}{\gamma(A, d)} \tag{3}$$

3 Parallel Reducts

$DS = (U, A, d)$ denotes a decision system. $P(DS)$ is the set of all subsystems of DS. The symbol F is a nonempty subset of $P(DS)$, which excludes the empty element ϕ, i.e. $\phi \notin F$.

Definition 4. *[14] Let DS = (U,A, d) be a decision system, and P(DS) be the set of all subsystems of DS, $F \subseteq P(DS)$. $B \subseteq A$ is called a parallel reduct of F iff B satisfies the following two conditions:*

1. *For any subsystem $DT \in F$ it satisfies $\gamma(B, d) = \gamma(A, d)$.*
2. *For any $S \subset B$, there exists at least a subsystem $DT \in F$ such that $\gamma(S, d) \neq \gamma(A, d)$.*

4 Matrix of Attribute Significance

In this section we review the matrix of attribute significance[14].

Definition 5. *Let $DS = (U, A, d)$ be a decision system, and $P(DS)$ be the set of all subsystems of DS, $F \subseteq P(DS)$, $B \subseteq A$, the matrix of attribute significance B relative to F is defined as:*

$$M(B,F) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nm} \end{bmatrix}. \tag{4}$$

Where $\sigma_{ij} = \gamma_i(B, d) - \gamma_i(B - \{a_j\}, d)$, $a_j \in B$, $(U_i, A, d) \in F$, $\gamma_i(B, d) = \frac{|POS_B(DT_i,d)|}{|U_i|}$, n denotes the number of decision tables in F, m denotes the number of conditional attributes in B.

Proposition 1. *The core of F-parallel reducts in the subsystems $F \subseteq P(DS)$ is the set of attributes whose attribute significance in corresponding column are all positive for every $DT \in F$.*

Definition 6. *Let $DS = (U, A, d)$ be a decision system, and $P(DS)$ be the set of all subsystems of DS, $F \subseteq P(DS)$, $B \subseteq A$, the modified matrix of attribute significance B relative to F is defined as:*

$$M'(B,F) = \begin{bmatrix} \sigma'_{11} & \sigma'_{12} & \cdots & \sigma'_{1m} \\ \sigma'_{21} & \sigma'_{22} & \cdots & \sigma'_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ \sigma'_{n1} & \sigma'_{n2} & \cdots & \sigma'_{nm} \end{bmatrix}. \tag{5}$$

Where $\sigma'_{ij} = \gamma_i(B \cup \{a_j\}, d) - \gamma_i(B, d)$, $a_j \in A$, $(U_i, A, d) \in F$, $\gamma_i(B, d) = \frac{|POS_B(DT_i,d)|}{|U_i|}$, n denotes the number of decision tables in F, m denotes the number of conditional attributes in DS.

It is easy to know that if $a_j \in B$, the element σ_{ij} in the matrix $M'(B, F)$ is 0.

5 Parallel Reducts Based on the Matrix of Attribute Significance

We illustrate parallel reducts with attribute significance algorithm[14] in this section.Core of parallel reducts can be got through the matrix of attribute significance at first. We obtain the rest of parallel reducts through the modified matrix of attribute significance. Elaborate algorithm is shown below.

Algorithm 1. Parallel reducts based on the matrix of attribute significance(PRMAS).

Input: A series of subsystems $F \subseteq P(DS)$.

Output: A parallel reduct.

Step 1. Establish matrix of attribute significance with subtract policy $M(A, F)$;

Step 2. $C = \bigcup_{j=1}^{m}\{a_j : \exists\sigma_{kj}(\sigma_{kj} \in M(A, F) \wedge \sigma_{kj} \neq 0)\}$;
B=A-C;
// Build core attributes which has at least one positive value in
// corresponding column in matrix $M(A, F)$.

Step 3. Do the following steps.
(1)rebuild matrix $M'(B, F)$ with addition policy;
if $M'(B, F) = 0$ break;
(2)For $j = 1$ to m do
For $k = 1$ to n do
If $\sigma_{kj} \neq 0$ then $t_j = t_j + 1$;
// count the number of $\sigma_{kj} \neq 0$ in a column.
(3)$C = C \bigcup\{a_j : \exists t_j(t_j \neq 0 \wedge \forall t_p(t_j \geq t_p))\}$;
$B = B - a_j$
// add the attribute, which is supported by maximum sub-tables,
// to the set of reduct C.
(4)goto step (1)

Step 4. Output the reduct C.

According to [14], the time complexity of this algorithm is $O(nm^3|U'|log|U'|)$ in the worst case, where $|U'|$ denotes the number of instances in one decision table which has the largest cardinality in F, n denotes the number of sub-tables, and m denotes the number of conditional attributes.

6 Hashing Classification

Classification takes a great deal of time in parallel reducts. Meanwhile, comparison operations dominate classification process. Early parallel reducts programs, adopting brute force strategy, need a lot of time to do comparison operations in classification. This can't be accepted in reducing large decision tables. Hashing approach can solve classification problem in an elegant and efficient manner especially in reducing comparison operations. Each instance can map into a unique value by means of hash function, which reduces comparison operations sharply. After hashing, any instances belonging to the same equivalence class must get the

same hash value, though, some instances not belonging to the same equivalence class can get the same hash value, either.

A good hash function must be found for classification purpose, and it should satisfy at least two conditions: easily calculating and enough discriminatory power. A trivial choice is to just add all the condition attribute value in one instance. However, this trivial hash function doesn't work well under many cases. An improved hash function is adopted in this paper: Every four conditional attribute values in one instance are sampled simply their least significant byte into a complete 32-bit integer, which is processed into the last hash value for the corresponding instance.

The complete classification procedure is as follows: A hash jump table and a conditional equivalence classes buffer are constructed firstly; secondly all the instances are mapped into corresponding hash slots in the jump table; thirdly each instance can find where they should store in the buffer exactly from corresponding jump table pointer; finally some inevitable comparison operations must be processed properly. Elaborate algorithm is shown below.

Algorithm 2. Quick partition equivalence class(QPEC).

Input: A decision table.

Output: Conditional equivalence classes of the decision table.

Step 1. Reserve memory for pointer table, initialized zero;

```
//pointer table has a fixed number of slots, whose index number
//is a hash value
Reserve memory for equivalence classes, initialized zero;
//a corresponding slot in pointer table points to an
//equivalence class cell
```

Step 2. Do the following steps for each instance of the decision table.

```
(1)r = hash(instance);
//bit operation is employed in obtaining a hash value
(2)if (r is a new hash value in pointer table){
      let corresponding slot in pointer table point to
      its equivalence class;
      store instance to corresponding equivalence class;
      //even the two instances have the same hash value,
      //one comparison operation is taken to insure
      //they belong to the same equivalence class
   }
   else{
      directly store instance to corresponding equivalence class;
      or
      some comparison steps to store instance to equivalence class;
      //actually speaking, equivalence class memory region is
      //a linked list, all the cells with the same hash value
      //are linked together
   }
```

Step 3. Output the conditional equivalence classes.

Figures 1,2 and 3 show that our hash function provides good discriminatory power for equivalence class partitioning. The left part of each figure shows how many instances in an equivalence class through hashing classification; the right part of each figure depicts how many instances in an equivalence class of its own. Abalone is an excellent example to demonstrate classification power of hashing approach, where the number of hashing slots is almost identical to the number of real equivalence classes. In Poker Hand, we can confirm that almost all the equivalence classes have no more than 100 instances, even below 10 for the majority of cases. In Forest CoverType, 581,012 instances are partitioned into 237,925 equivalences classes, where each equivalence class includes on average 2.44 instances. It is because of this improvement can we decrease comparison operations dramatically.

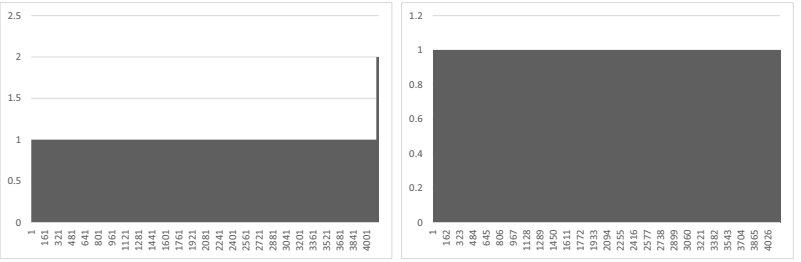


Fig. 1. Abalone

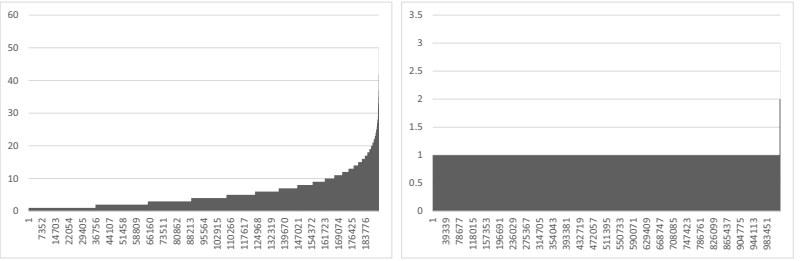


Fig. 2. Poker Hand

By the way, all the data in figures list above are sorted with ascending order. There is a simple cause here: One can draw a falsity conclusion with original Fig.4 just because there is too much data in the original excel chart! That is to say, one may have an illusion that any equivalence class in Poker Hand has at least 10 instances, but not below 8 in the majority of cases.

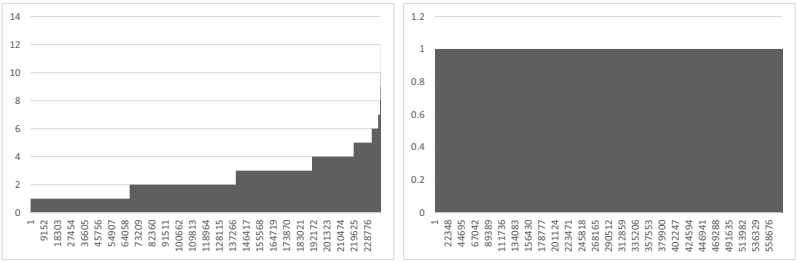


Fig. 3. Forest CoverType

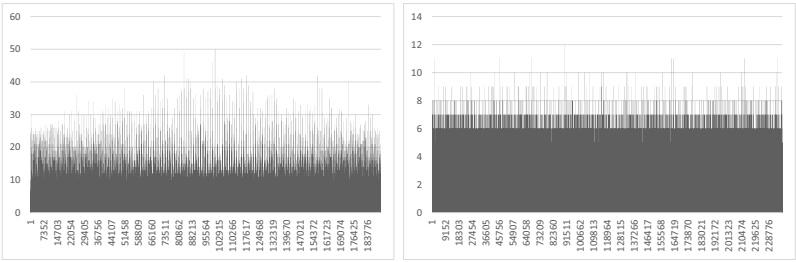


Fig. 4. Original hashing classification(upside Poker Hand, downside Forest CoverType)

7 Performance Analysis of Parallel Reducts with Hashing Approach

We come to analyze the algorithm performance of Parallel reducts with hashing approach(PRH). Its running time is mainly consumed in partitioning equivalence classes and comparison operations dominate the algorithm. So comparison operation is chosen as algorithm measuring benchmark in the following algorithm complexity analysis.

In our experiment, Poker Hand test needs 323 783 573 times comparison operations in all. This test needs 4 loops, in which there are 10 sub tables involved. Each sub table needs at least 11 loops of calculation. That is to say, each object needs about 1 comparison operation to complete equivalence class partition. Forest CoverType needs 2 330 252 207 times comparison operations in all. This test needs 7 loops, in which there are 10 sub tables involved. Each sub table needs at least 11 loops of calculation. That is to say, each object needs about 5 comparison operations to complete equivalence class partition.

Under general circumstances, hashing classification can partition a decision table into quite a lot of small equivalence classes, each having about the same size and containing elements less than 50 or even 20. In this case, the decision table is scanned once and partition is done. Assuming that each equivalence class contains 100 instances, we need about $|A|(|U|/100) \times 100^2$ times comparison operations. According to our experimental statistics, the number of instances in

a small equivalence class rarely reaches 100, and it is often less than 50. So the average complexity of our reducts algorithm is $O(|A||U|)$, where $|U|$ denotes the number of instances in decision table, $|A|$ is the number of conditional attributes. In the worst case, where every two instances need to compare with each other, performance of our algorithm degenerates to $O(|A||U|^2)$. Fortunately, the worst case seldom occurs in our experiment.

Finally we come to analyze the algorithm space efficiency. Our hashing algorithm needs to load the whole decision table into memory. At running time it needs to allocate more memory for jump table and conditional equivalence class buffer in which each record needs 140 bytes. Take Forest CoverType for example, the decision table needs to allocate about 125MB memory on the main thread heap, 38MB for the equivalence classes buffer, about 300KB for the jump table. So 168.3MB is required in all in Forest CoverType test process. They are all proportionate to the number of instances and the number of attribute. Now we get a conclusion that our algorithm space complexity is $O(|A||U|)$.

8 Experiments

UCI repository of machine learning databases is employed in our experiments. We use RIDAS system(developed by Chongqing University of Posts and Telecommunications) to normalize the data. 10 sub-tables are created from original data set. The first sub-table has 10% data of the complete data from the decision table, the second 20%, the third 30%, and so on.

We run the experiment on a Dell 14R Turbo laptop computer. The machine has an Intel(R) Core(TM) i7 3632QM 2.2GHz CPU, 8GB DDR3 memory and 1TB hard disk. Microsoft Visual Studio 2012 Express, running on Microsoft Windows 8 China edition, is employed as our development software.

In Table 1, the symbol '10(9)' denotes that there is 10 condition attributes in the original database, and there is a useless attribute for reducts.

In Table 2 Column 2 denotes the corresponding decision table attribute reducts result. Take "000 000 11" as example, the first six '0' symbols denote that attributes from column 1 to column 6 are exactly a parallel reduct for Abalone, the

Table 1. Data sets description

No.	Data	Features	Instances
1	Abalone	8	4177
2	Breast-cancer-Wisconsin	10(9)	699
3	Mushroom	22	8124
4	Letter Recognition	16	20000
5	Adult	14	48842
6	Chess (King-Rook vs. King)	6	28056
7	Shuttle	9	43500
8	Poker Hand	10	1025010
9	Forest CoverType	54	581012

Table 2. PRH reducts results

No.	Reduct Results
1	000 000 11
2	000 010 111
3	100 000 111 011 000 011 110 0
4	100 000 000 000 000 0
5	000 010 000 000 00
6	000 000
7	000 001 111
8	000 010 101 0
9	000 001 111 111 111 111 111 111 111 111 111 111 111 111 111 111 111 111

Table 3. Performance comparison between PRMAS and PRH

No.	Data set	PRMAS(s)	PRH(s)
1	Abalone	11.313000	0.078000
2	Breast-cancer-Wisconsin	0.219000	0.016000
3	Mushroom	62.507000	0.157000
4	Letter Recognition	100.167999	0.140000
5	Adult	469.014008	0.454000
6	Chess (King-Rook vs. King)	75.082001	0.078000
7	Shuttle	405.941986	0.516000
8	Poker Hand	time consuming	13.563000
9	Forest CoverType	time consuming	120.990997

last two '1' symbols denote that attributes from column 7 to column 8 are redundant attributes. Of course, each reduct result is identical with the corresponding result in PRMAS.

Table 3 demonstrates our parallel reducts results from the different size decision tables. The number of decision table instances less than 50,000, such as Adult, takes 0.454 second(s) to complete its attribute reduct. Even the large table like Poker Hand takes only 13.563 seconds to complete attribute reduct entirely, which is definitely an inspiring result comparing to the corresponding result in PRMAS.

Table 3 also presents the detailed comparison between PRH and PRMAS. Our hashing approach has significant advantage over the original approach. PRH has more than 100 times performance boost compared with PRMAS.

Our experiment also demonstrates that cache miss can influence program performance significantly, which can be confirmed from Table 4. Take Adult as example, the default configuration takes more than about 53% running time than the optimized performance. It is simply because each equivalence class needs to consume 140 bytes in default configuration, while a cache block size in 3632QM is 64 bytes, meaning that cache miss is likely to occur frequntly. On the other

Table 4. Cache miss rate impact on PRH reducts

No.	Data set	Default(s)	Optimized(s)
1	Abalone	0.078000	0.063000
2	Breast-cancer-Wisconsin	0.016000	0.015000
3	Mushroom	0.157000	0.109000
4	Letter Recognition	0.140000	0.110000
5	Adult	0.454000	0.296000
6	Chess (King-Rook vs. King)	0.078000	0.047000
7	Shuttle	0.516000	0.343000
8	Poker Hand	13.563000	12.813000
9	Forest CoverType	120.990997	115.693000

hand, we decrease equivalence class size to 16 bytes in some test data, which of course increases cache hit rate. Finally, large table like Forest CoverType, however, is almost no influenced by cache miss.

9 Conclusion

In this paper, we present a parallel reducts algorithm with hashing approach in matrix of attribute significance algorithm framework. For a family of decision subsystems, we define F-attribute significance and apply it to obtain parallel reducts. Experimental results show that our approach has evident advantages over the original reducts algorithms. The choice of heuristic information and design of hash function have an influence on the experimental results. Both of them can be further improved to enhance efficiency.

References

1. Pawlak, Z.: Rough Sets-Theoretical Aspect of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
2. Liu, Q.: Rough Sets and Rough Reasoning. Science Press (2001) (in Chinese)
3. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1991)
4. Hu, X., Cercone, N.: Learning in Relational Databases: A Rough Set Approach. Computational Intelligence 11(2), 323–337 (1995)
5. Deng, D., Huang, H.: A New Discernibility Matrix and Function. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 114–121. Springer, Heidelberg (2006)
6. Miao, D., Wang, J.: An Information Representation of the Concepts and Operations in Rough Set Theory. Chinese Journal of Software 10(2), 113–116 (1999)
7. Wang, G., Yu, H., Yang, D.: Decision Table Reduction based on Conditional Information Entropy. Chinese Journal of Computers 25(7), 759–766 (2002)

8. Bazan, G.J.: A Comparison of Dynamic Non-dynamic Rough Set Methods for Extracting Laws from Decision Tables. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 1: Methodology and Applications, pp. 321–365. Physica-Verlag, Heidelberg (1998)
9. Bazan, G.J., Nguyen, H.S., Nguyen, S.H., Synak, P., Wroblewski, J.: Rough Set Algorithms in Classification Problem. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications, pp. 49–88. Physica-Verlag (2000)
10. Deng, D., Wang, J., Li, X.: Parallel Reducts in a Series of Decision Subsystems. In: Proceedings of the Second International Joint Conference on Computational Sciences and Optimization (CSO 2009), Sanya, Hainan, China, pp. 377–380 (2009)
11. Deng, D.: Comparison of Parallel Reducts and Dynamic Reducts in Theory. Computer Science 36(8A), 176–178 (2009) (in Chinese)
12. Deng, D.: Parallel Reducts and Its Properties. In: Proceedings of 2009 IEEE International Conference on Granular Computing, pp. 121–125 (2009)
13. Deng, D.: (F, ϵ)-Parallel Reducts in a Series of Decision Subsystems. In: Proceedings of the Third International Joint Conference on Computational Sciences and Optimization(CSO 2010), pp. 372–376 (2010)
14. Deng, D., Yan, D., Wang, J.: Parallel Reducts Based on Attribute Significance. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS (LNAI), vol. 6401, pp. 336–343. Springer, Heidelberg (2010)
15. Deng, D., Yan, D., Chen, L.: Attribute Significance for F-Parallel Reducts. In: Proceedings of 2011 IEEE International Conference on Granular Computing, pp. 156–161 (2011)
16. Knuth, D.E.: Sorting and Searching, 2nd edn. The Art of Computer Programming, vol. 3. Addison-Wesley (1998)
17. Wang, P.C.: Efficient hash-based approximate reduct generation. In: Proceedings of 2011 IEEE International Conference on Granular Computing, pp. 703–707 (2011)

A Parallel Implementation of Computing Composite Rough Set Approximations on GPUs

Junbo Zhang[1,2], Yun Zhu[2], Yi Pan[2], and Tianrui Li[1,⋆]

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China
[2] Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA
JunboZhang86@163.com, yzhu7@student.gsu.edu, pan@cs.gsu.edu, trli@swjtu.edu.cn

Abstract. In information systems, there may exist multiple different types of attributes like categorical attributes, numerical attributes, set-valued attributes, interval-valued attributes, missing attributes, *etc.* Such information systems are called as composite information systems. To process such attributes with rough set theory, composite rough set model and corresponding matrix methods were introduced in our previous research. Rough set approximations of a concept are the basis for rule acquisition and attribute reduction in rough set based methods. To accelerate the computation process of rough set approximations, this paper first presents the boolean matrix representation of the lower and upper approximations in the composite information system, then designs a parallel method based on matrix, and implements it on GPUs. The experiments on data sets from UCI and user-defined data sets show that the proposed method can accelerate the computation process efficiently.

Keywords: Composite Rough Sets, Boolean Matrix, GPU, CUDA.

1 Introduction

The rough set (RS) theory is a powerful mathematical tool to describe the dependencies among attributes, evaluate the significance of attributes, and derive decision rules [15]. It plays an important role in the fields of data mining and machine learning [7,18,21,22]. Different attributes can be processed by different rough set models. For example, Hu et al. generalized classical rough set model with neighborhood relations to deal with numerical attributes [7]. Guan et al. defined the tolerance relation and used it to deal with set-valued attributes [5]. Grzymała-Busse integrated the tolerance relation [8] and the similarity relation [19], and proposed the characteristic relation [4] for missing attributes in incomplete information systems. In real-applications, there are multiple different types of attributes in information systems like categorical attributes, numerical

⋆ Corresponding author.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 240–250, 2013.

attributes, set-valued attributes, and missing attributes. Such information systems are called as composite information systems. Most of rough set models fail to deal with more than two types of attributes. To solve this problem, we gave the composite rough set model, defined a composite relation and used composite classes to drive approximations from composite information systems in our previous work [23]. Table 1 shows these rough set models for different types of attributes.

Table 1. Rough Set Models

Model	Relation	Data Types			
		C	N	S	M
Classical RS	Equivalence [15]	√	×	×	×
Neighborhood RS	Neighborhood [7]	√	√	×	×
Set-valued RS	Tolerance [5]	√	×	√	×
Characteristic RS	Characteristic [3]	√	×	×	√
Composite RS	Composite [23]	√	√	√	√

C: Categorical, N: Numerical, S: Set-valued, M: Missing

Rough set approximations of a concept are the basis for rule acquisition and attribute reduction in rough set based methods. The efficient calculation of rough set approximations can accelerate the process of knowledge discovery effectively. Parallelization of algorithms is a good way to speed up the computational process. In our previous work, we proposed a parallel algorithm for computing rough set approximation [21], however, it can only process categorical attributes. In this paper, to deal with composite attributes and compute rough set approximations from composite information systems, we give a boolean-based method for computing rough set approximations. It means that the calculation of rough set approximations can be processed as boolean matrix operations. We design the parallel matrix-based method and parallelize it on GPUs, which have recently been utilized in various domains, including high-performance computing [14]. NVIDIA GPUs [1] power millions of desktops, notebooks, workstations and supercomputers around the world, and accelerate computationally-intensive tasks for professionals, scientists, researchers, etc. NVIDIA CUDA [2] is a General Purpose Computation on GPUs (GPGPUs) framework, which uses a C-like programming language and does not require re-mapping algorithms to graphics concepts. These features help users develop correct and efficient GPU programs easily.

The remainder of the paper is organized as follows. Section 2 introduces some rough set models. Section 3 proposes the boolean matrix-based method. Section 4 designs the parallel method for computing composite rough set (CRS) approximations. Section 5 gives the experimental analysis. The paper ends with conclusions and future work in Section 6.

2 Rough Set Model

In this section, we first briefly review the concepts of rough set model as well as their extensions [5,7,10,15,16,23].

2.1 Classical Rough Set Model

Given a pair $K = (U, R)$, where U is a finite and non-empty set called the universe, and $R \subseteq U \times U$ is an indiscernibility relation on U. The pair $K = (U, R)$ is called an approximation space. $K = (U, R)$ is characterized by an information system $IS = (U, A, V, f)$, where U is a non-empty finite set of objects; A is a non-empty finite set of attributes; $V = \bigcup_{a \in A} V_a$ and V_a is a domain of attribute a; $f : U \times A \to V$ is an information function such that $f(x, a) \in V_a$ for every $x \in U$, $a \in A$. In the classical rough set model, R is the equivalence relation. Let $B \subseteq A$ and $[x]_{R_B}$ denote an equivalence class of an element $x \in U$ under the indiscernibility relation R_B, where $[x]_{R_B} = \{y \in U | x R_B y\}$.

Classical rough set model is based on the equivalence relation. The elements in an equivalence class satisfy reflexive, symmetric and transitive. It also cannot deal with the non-categorical attributes like numerical attributes, set-valued attributes, etc. However, non-categorical attributes appear frequently in real applications [3,6,8,17]. Therefore, it is necessary to investigate the situation of non-categorical attributes in information systems. In what follows, we just introduce two rough set models [5,7], which will be used in our examples. More rough set models for dealing with non-categorical attributes are available in the literatures [3,6,8,9,12,17,20].

2.2 Composite Rough Set (CRS) Model

In many practical issues, there are multiple different types of attributes in information systems, called composite information systems. A composite information system can be written as $CIS = (U, A, V, f)$, where
(i) U is a non-empty finite set of objects;
(ii) $A = \bigcup A_k$ is a union of attribute sets, and A_k is an attribute set with the same type of attributes;
(iii) $V = \bigcup_{A_k \subseteq A} V_{A_k}$, $V_{A_k} = \bigcup_{a \in A_k} V_a$, V_a is a domain of attribute a;
(iv) $f : U \times A \to V$, namely, $U \times \bigcup A_k \to \bigcup V_{A_k}$, and $U \times A_k \to V_{A_k}$ is an information function, $f(x, a)$ denotes the value of object x on attribute a.

Definition 1. [23] *Given $x, y \in U$ and $B = \bigcup B_k \subseteq A$, $B_k \subseteq A_k$, the composite relation CR_B is defined as*

$$CR_B = \{(x, y) | (x, y) \in \bigcap_{B_k \subseteq B} R_{B_k}\} \tag{1}$$

where $R_{B_k} \subseteq U \times U$ is an indiscernibility relation defined by an attribute subset B_k on U [16].

When $(x, y) \in CR_B$, we call x and y are indiscernible w.r.t. B. Let $CR_B(x) = \{y | y \in U, \forall B_k \in B, yR_{B_k}x\}$, we call $CR_B(x)$ the composite class for x w.r.t. CR_B.

Definition 2. [23] *Given a composite information system $CIS = (U, A, V, f)$, $\forall X \subseteq U$, $B \subseteq A$, the lower and upper approximations of X in terms of composite relation CR_B are defined as*

$$\underline{CR_B}(X) = \{x \in U | CR_B(x) \subseteq X\} \tag{2}$$

$$\overline{CR_B}(X) = \{x \in U | CR_B(x) \cap X \neq \emptyset\} \tag{3}$$

3 Boolean Matrix-Based Method

In this section, we present the boolean matrix representation of the lower and upper approximations in the composite information system. Before this, we review the matrix-based approaches in rough set model. A set of axioms were constructed to characterize classical rough set upper approximation from the matrix point of view by Liu [11]. Zhang et al. defined a basic vector $H(X)$, which was induced from the relation matrix. And four cut matrices of $H(X)$, denoted by $H^{[\mu,\nu]}(X)$, $H^{(\mu,\nu]}(X)$, $H^{[\mu,\nu)}(X)$ and $H^{(\mu,\nu)}(X)$, were derived for the computation of approximations, positive, boundary and negative regions intuitively in set-valued information systems [24]. Furthermore, Zhang et al. gave the matrix-based methods for computing rough set approximations in composite information systems [23]. Here, we follow their work and give a novel boolean matrix-based method to process composite data.

3.1 Boolean Matrix-Based Method in the Composite Information System

Definition 3. [11] *Let $U = \{x_1, x_2, \ldots, x_n\}$, and X be a subset of U. The characteristic function $G(X) = (g_1, g_2, \ldots, g_n)^T$ (T denotes the transpose operation) is defined as*

$$g_i = \begin{cases} 1, & x_i \in X \\ 0, & x_i \notin X \end{cases} \tag{4}$$

where $G(X)$ assigns 1 to an element that belongs to X and 0 to an element that does not belong to X.

Definition 4. *Given a composite information system $CIS = (U, A, V, f)$. Let $B \subseteq A$ and CR_B be a composite relation on U, $M_{n\times n}^{CR_B} = (m_{ij})_{n\times n}$ be an $n \times n$ matrix representing CR_B, called the relation matrix w.r.t. B. Then*

$$m_{ij} = \begin{cases} 1, & (x_i, x_j) \in CR_B \\ 0, & (x_i, x_j) \notin CR_B \end{cases} \tag{5}$$

Corollary 1. *Let* $M_{n\times n}^{CR_B} = (m_{ij})_{n\times n}$ *and* CR_B *be a composite relation on* U. *Then* $m_{ii} = 1$, $1 \le i \le n$.

Next, we discuss about boolean methods to derive lower and upper approximation in composite rough sets.

Lemma 1. *Given* $X \subseteq U$ *in a composite information system* $CIS = (U, A, V, f)$, *where* $U = \{x_1, x_2, \ldots, x_n\}$. $B \subseteq A$ *and* CR_B *is a composite relation on* U. *Then the lower and upper approximations of* X *in the composite information system can be computed as follows.*
(1) The n-column boolean vector $G(\overline{CR_B}(X))$ *of the upper approximation* $\overline{CR_B}(X)$*:*

$$G(\overline{CR_B}(X)) = M_{n\times n}^{CR_B} \otimes G(X) \tag{6}$$

where $\otimes$ *is the Boolean product of matrices.*
(2) The n-column boolean vector $G(\underline{CR_B}(X))$ *of the lower approximation* $\underline{CR_B}(X)$*:*

$$G(\underline{CR_B}(X)) = -(M_{n\times n}^{CR_B} \otimes G(-X)) \tag{7}$$

where $-X$ *denotes the complementary set to* X.

Proof. Suppose $M_{n\times n}^{CR_B} = (m_{ik})_{n\times n}$, $G(X) = (g_1, g_2, \cdots, g_n)^T$ and $G(\overline{CR_B}(X)) = (u_1, u_2, \cdots, u_n)^T$. $\wedge, \vee$ denote minimum and maximum operators, respectively.

(1) "$\Rightarrow$": $\forall i \in \{1, 2, \cdots, n\}$, if $u_i = 1$, then $x_i \in \overline{CR_B}(X)$, $CR_B(x_i) \cap X \neq \emptyset$, and $\exists x_j \in CR_B(x_i), x_j \in X$, that is to say $(x_i, x_j) \in CR_B$. Thus, $m_{ij} = 1$ and $g_j = 1$, and $\bigvee_{k=1}^{n} (m_{ik} \wedge g_k) = m_{ij} \wedge g_j = 1$. Hence, $\forall i \in \{1, 2, \cdots, n\}$, $u_i \le \bigvee_{k=1}^{n} (m_{ik} \wedge g_k)$.

"$\Leftarrow$": $\forall i \in \{1, 2, \cdots, n\}$, if $\bigvee_{k=1}^{n} (m_{ik} \wedge g_k) = 1$, then $\exists j \in \{1, 2, \cdots, n\}$, $m_{ij} = 1$ and $g_j = 1$. Thus, $x_j \in CR_B(x_i)$ and $x_j \in X$. Then, $CR_B(x_i) \cap X \neq \emptyset$, namely, $x_i \in \overline{CR_B}(X)$ and $u_i = 1$. Therefore, $\forall i \in \{1, 2, \cdots, n\}$, $u_i \ge \bigvee_{k=1}^{n} (m_{ik} \wedge g_k)$.

Thus, $\forall i \in \{1, 2, \cdots, n\}$, $u_i = \bigvee_{k=1}^{n} (m_{ik} \wedge g_k)$, namely, $G(\overline{CR_B}(X)) = M_{n\times n}^{CR_B} \otimes G(X)$.

(2) The proof is similar to that of (1).

Corollary 2. *Let* $M_{n\times n}^{CR_B} = (m_{ik})_{n\times n}$ *and* $G(X) = (g_1, g_2, \cdots, g_n)^T$. *Suppose* $G(\overline{CR_B}(X)) = (u_1, u_2, \cdots, u_n)^T$ *and* $G(\underline{CR_B}(X)) = (l_1, l_2, \cdots, l_n)^T$, $\forall i \in \{1, 2, \cdots, n\}$, *we have*

$$\begin{cases} u_i = \bigvee_{k=1}^{n} (m_{ik} \wedge g_k) \\ l_i = \bigwedge_{k=1}^{n} (m_{ik} \odot g_k) \end{cases} \tag{8}$$

where $\odot$ *is the logical operation NXOR (not exclusive or).*

3.2 Boolean Matrix-Based Method in the Composite Decision Table

Definition 5. *Given a composite decision table* $CDT = (U, A \cup D, V, f)$, $B \subseteq A$, *let* $U/D = \{D_1, D_2, \cdots, D_r\}$ *be a partition over the decision* D. $\forall D_j \in U/D$, $G(D_j) = (d_{j1}, d_{j2}, \cdots, d_{jn})^T$ *is an* n*-column boolean vector of* D_j. *Let* $G(D) = (G(D_1), G(D_2), \cdots, G(D_r)) = (d_{kj})_{n\times r}$ *and* $G(-D) = (G(-D_1), G(-D_2), \cdots, G(-D_r)) = (-d_{kj})_{n\times r}$ *be* $n \times r$ *boolean matrices, called decision matrix and decision complementary matrix.*

Lemma 2. *Given a composite decision table* $CDT = (U, A \cup D, V, f)$, $B \subseteq A$. *Let* $U/D = \{D_1, D_2, \cdots, D_r\}$ *be a partition over the decision* D. $\forall j = 1, 2, \cdots, r$, *the upper and lower approximations of the decision* D *in the composite information system can be computed as follows.*

(1) *The* $n \times r$ *boolean matrices* $G(\overline{CR_B}(D))$ *of the upper approximations of the decision* D*:*

$$G(\overline{CR_B}(D)) = M_{n\times n}^{CR_B} \otimes G(D) \tag{9}$$

(2) *The* $n \times r$ *boolean matrices* $G(\underline{CR_B}(D))$ *of the lower approximation of the decision* D*:*

$$G(\underline{CR_B}(D)) = -(M_{n\times n}^{CR_B} \otimes G(-D)) \tag{10}$$

Proof. The proof is similar to that of Lemma 1.

Corollary 3. *Let* $M_{n\times n}^{CR_B} = (m_{ik})_{n\times n}$ *and* $G(D) = (d_{kj})_{n\times r}$. *Suppose* $G(\overline{CR_B}(D)) = (u_{ij})_{n\times r}$ *and* $G(\underline{CR_B}(D)) = (l_{ij})_{n\times r}$, $\forall i \in \{1, 2, \cdots, n\}$, $j \in \{1, 2, \cdots, r\}$, *we have*

$$\begin{cases} u_{ij} = \bigvee\limits_{k=1}^{n} (m_{ik} \wedge d_{kj}) \\ l_{ij} = \bigwedge\limits_{k=1}^{n} (m_{ik} \odot d_{kj}) \end{cases} \tag{11}$$

4 Parallel Method for Computing CRS Approximations

4.1 Parallel Matrix-Based Method

According to the above matrix method, we first give the sequential algorithm for computing rough set approximations in the composite decision table, which is outlined in Algorithm 1. Step 2 is to construct the relation matrix and its time complexity is $O(n^2|B|)$; Step 3 is to construct the decision matrix and its time complexity is $O(n \log r + n)$; Step 4 is to compute the upper approximation matrix and its time complexity is $O(n^2 r)$; Step 5 is to compute the lower approximation matrix and its time complexity is $O(n^2 r)$; Hence, the total time complexity is $O(n^2|B| + n \log r + n + n^2 r + n^2 r) = O(n^2(|B| + r))$.

According to the above analysis, the most intensive computation to occur is the construction of the relation matrix and the computation of the upper and

Algorithm 1. The sequential algorithm for computing rough set approximations

Input: A composite decision table $CDT = (U, A \cup D, V, f)$, and attribute subset $B \subseteq A$.
Output: The rough set approximations of the decision D.

```
begin
    Construct the relation matrix: M^{CR_B}_{n×n} = (m_ij)_{n×n}. // According to Definition 4
    Construct the decision matrix: G(D) = (d_jk)_{n×r}. // According to Definition 5
    Compute the upper approximation matrix: G(overline{CR_B}(D)) = M^{CR_B}_{n×n} ⊗ G(D). //
    According to Corollary 2
    Compute the lower approximation matrix: G(underline{CR_B}(D)) = −M^{CR_B}_{n×n} ⊗ G(−D). //
    According to Corollary 2
    Output the rough set approximations.
end
```

Algorithm 2. The parallel algorithm for computing rough set approximations

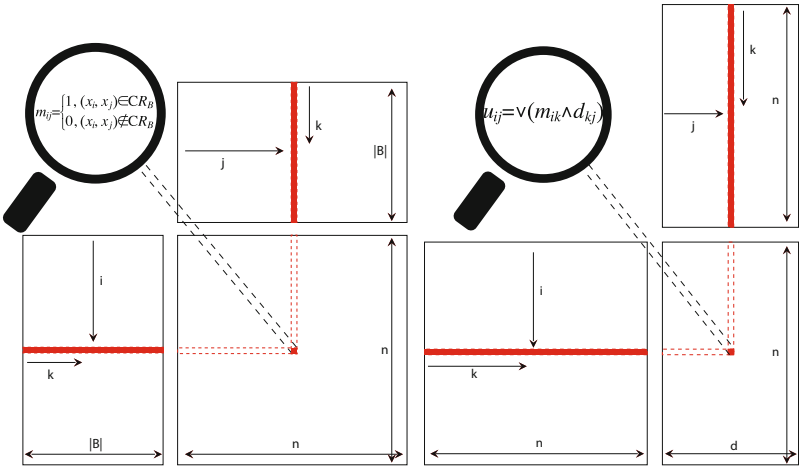
Input: A composite decision table $CDT = (U, A \cup D, V, f)$, and attribute subset $B \subseteq A$.
Output: The rough set approximations of the decision D.

```
begin
    [In Parallel    ] Construct the relation matrix: M^{CR_B}_{n×n} = (m_ij)_{n×n}.
    [In Sequential] Construct the decision matrix: G(D) = (d_jk)_{n×r}.
    [In Parallel    ] Compute the upper approximation matrix:
    G(overline{CR_B}(D)) = M^{CR_B}_{n×n} ⊗ G(D).
    [In Parallel    ] Compute the lower approximation matrix:
    G(underline{CR_B}(D)) = −M^{CR_B}_{n×n} ⊗ G(−D).
    [In Sequential] Output the rough set approximations.
end
```

lower approximation matrices. Obviously, we can accelerate the computational process through the parallelization of these steps, as shown in Algorithm 2.

There are n objects and $|B|$ attributes in raw data. It can be seen as the matrix data with n rows and $|B|$ columns denoted by $\mathcal{V} = (\nu)_{n\times|B|}$. In Figure 1(a), suppose $x_i = (c_{ik})_{1\times|B|}$ and $x_j^T = (c_{kj})_{|B|\times 1}$, then m_{ij} is the result of composite operation of two vectors x_i and x_j^T. If $(x_i, x_j) \in CR_B$, $m_{ij} = 1$; otherwise, $m_{ij} = 0$. Hence, each m_{ij} can be computed independently and in parallel.

Similarly, after constructing relation matrix $M_{n\times n}^{CR_B} = (m_{ik})_{n\times n}$ and decision matrix $G(D) = (d_{kj})_{n\times r}$, we suppose the upper approximation matrix $G(\overline{CR_B}(D)) = (u_{ij})_{n\times r}$. According to Corollary 2, we have $G(\overline{CR_B}(D)) = M_{n\times n}^{CR_B} \otimes G(D)$. Hence, suppose $m_i = (m_{ik})_{1\times n}$ and $d_j = (d_{kj})_{n\times 1}$, then $u_{ij} = \bigvee_{k=1}^{n} (m_{ik} \wedge d_{kj})$ according to the Boolean operation, as shown in Figure 1(b). Hence, each u_{ij} can be computed independently and in parallel. Similarly, we can also compute lower approximation.



(a) Relation matrix (b) Upper approximation matrix

Fig. 1. Computing relation matrix and upper approximation matrix in parallel

Algorithm 3. CUDA Computing CRS Approximations

Input: A composite decision table $CDT = (U, A \cup D, V, f)$, and attribute subset $B \subseteq A$.
Output: The rough set approximations of the decision D.

begin
  (a) **Construct** the relation matrix:
  for *each* $m_{ij} \in M_{n\times n}^{CR_B}$ **do**
    $m_{ij} = \begin{cases} 1, & (x_i, x_j) \in CR_B \\ 0, & (x_i, x_j) \notin CR_B \end{cases}$
  end
  (b) **Construct** the decision matrix: $G(D) = (d_{jk})_{n\times r}$.
  (c) **Compute** the upper approximation matrix:
  for *each* $u_{ij} \in G(\overline{CR_B}(D))$ **do**
    $u_{ij} = \bigvee_{k=1}^{n} (m_{ik} \wedge d_{kj})$
  end
  (d) **Compute** the lower approximation matrix:
  for *each* $l_{ij} \in G(\underline{CR_B}(D))$ **do**
    $l_{ij} = \bigwedge_{k=1}^{n} (m_{ik} \odot d_{kj})$
  end
  Output the rough set approximations.
end

4.2 CUDA Implementation

Algorithm 3 shows the CUDA algorithm of computing CRS approximations. The Stages (a), (c), (d) can be computed in parallel with CUDA. The time complexities of Stages (a), (c), (d) are $O(n^2|B|/p)$, $O(n^2r/p)$, $O(n^2r/p)$, respectively, where p is the total number of threads in kernel function with CUDA. The time complexity of Stage (b) is $O(n \log r + n)$. Hence, the total time complexity of the CUDA algorithm is $O(n^2|B|/p + n^2r/p + n^2r/p + n \log r + n) = O(n^2(|B| + r)/p)$.

5 Experimental Analysis

Our test system consists of a Inter(R) Core(TM)i7-2670QM @2.20GHz (4 cores, 8 threads in all) and an NVIDIA GeForce GT 555M. We have implemented a GPU version with CUDA C [2], and a CPU version with C/C++. Then, we give a performance comparison between these two versions.

To test the performance of the parallel algorithm, we download the data set *Connect* from the machine learning data repository, University of California at Irvine [13]. The data set *Connect* consists of tens of thousands of samples. Each sample consists of 42 condition attributes and 1 decision attribute. In our experiment, we extract the data from the data set *Connect* with different numbers of samples randomly, *i.e.*, 8000, 12000, 16000, 20000, 24000, 28000, and 32000. Besides, user-defined data sets are used in our experiments, which are generated randomly with different sizes of samples and features.

Table 2 shows the information of data sets with the computational time and speedup. From the result of the data set *Connect*, it is easy to know that the computational time of CPU and GPU implementations increases with the increase of the size of data, and the GPU implementation achieves 1.6-2.2x over the CPU implementation on this data set. From the result of used-defined data sets, we also find that the computational time of CPU and GPU implementations increases with the increase of the size of data. Moreover, the GPU implementation performs better, achieves 3.1-4.4x speedup over the CPU implementation on used-defined data sets.

6 Conclusions

In this paper, we presented the boolean matrix representation of the lower and upper approximations in the composite information system. According to characteristic of matrix operations, we proposed a parallel method based on matrix for computing approximations. By time complexity analysis, the key steps are to construct the relation matrix and to compute the boolean matrices of the lower and upper approximations. Therefore, we used GPUs to parallelize and accelerate these steps. The performance comparison between these GPU implementation and CPU implementation was given, which showed our implementation could accelerate the process of computing rough set approximations. We will optimize the GPU implementation and design the composite rough set based feature selection on GPUs in future.

Table 2. Comparison of GPU and CPU

Data Set	Samples×Features(Classes)	Computational time (s)		Speedup
		CPU	GPU	
Connect	8000 × 42(3)	0.641315	0.377427	1.699176
	12000 × 42(3)	1.424260	0.860306	1.655527
	16000 × 42(3)	2.624810	1.462150	1.795171
	20000 × 42(3)	4.029880	2.253230	1.788490
	24000 × 42(3)	6.022810	3.301330	1.824358
	28000 × 42(3)	8.470160	4.541230	1.865168
	32000 × 42(3)	13.08390	5.910770	2.213569
User-defined	640 × 64(8)	0.006069	0.001801	3.369794
	1280 × 128(8)	0.026063	0.007054	3.694783
	2560 × 256(16)	0.159052	0.038413	4.140577
	3840 × 384(16)	0.366586	0.085946	4.265306
	5120 × 512(32)	1.217970	0.277423	4.390299
	10240 × 1024(32)	4.972580	1.114920	4.460033
	15360 × 1536(64)	18.809700	5.913800	3.180645
	20480 × 2048(64)	33.803200	10.527700	3.210881

Acknowledgements. This work is supported by the National Science Foundation of China (Nos. 60873108, 61175047, 61100117) and NSAF (No. U1230117), and the Science and Technology Planning Project of Sichuan Province (No. 2012RZ0009), China, and the Fostering Foundation for the Excellent Ph.D. Dissertation of Southwest Jiaotong University 2012.

References

1. NVIDIA GPUs (2013), `https://developer.nvidia.com/cuda-gpus`
2. NVIDIA CUDA (2013), `http://www.nvidia.com/object/cuda_home_new.html`
3. Grzymała-Busse, J.W.: Characteristic relations for incomplete data: A generalization of the indiscernibility relation. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 244–253. Springer, Heidelberg (2004)
4. Grzymała-Busse, J.W.: Characteristic relations for incomplete data: A generalization of the indiscernibility relation. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets IV. LNCS, vol. 3700, pp. 58–68. Springer, Heidelberg (2005)
5. Guan, Y., Wang, H.: Set-valued information systems. Information Sciences 176(17), 2507–2525 (2006)
6. Hu, Q., Xie, Z., Yu, D.: Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation. Pattern Recognition 40, 3509–3521 (2007)
7. Hu, Q., Yu, D., Liu, J., Wu, C.: Neighborhood rough set based heterogeneous feature subset selection. Information Sciences 178(18), 3577–3594 (2008)
8. Kryszkiewicz, M.: Rough set approach to incomplete information systems. Information Sciences 112(1-4), 39–49 (1998)
9. Leung, Y., Fischer, M.M., Wu, W.Z., Mi, J.S.: A rough set approach for the discovery of classification rules in interval-valued information systems. International Journal of Approximate Reasoning 47(2), 233–246 (2008)

10. Li, T., Ruan, D., Geert, W., 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)
11. Liu, G.: The axiomatization of the rough set upper approximation operations. Fundamenta Informaticae 69(3), 331–342 (2006)
12. Mi, J.S., Zhang, W.X.: An axiomatic characterization of a fuzzy generalization of rough sets. Information Sciences 160(1–4), 235–249 (2004)
13. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI Repository of Machine Learning Databases. University of California, Department of Information and Computer Science, Irvine, CA (1998), http://archive.ics.uci.edu/ml/
14. Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E., Purcell, T.: A survey of general-purpose computation on graphics hardware (2007)
15. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory, Knowledge Engineering and Problem Solving, vol. 9. Kluwer Academic Publishers, Dordrecht (1991)
16. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177(1), 28–40 (2007)
17. Qian, Y., Dang, C., Liang, J., Tang, D.: Set-valued ordered information systems. Information Sciences 179(16), 2809–2832 (2009)
18. Qian, Y., Liang, J., Pedrycz, W., Dang, C.: Positive approximation: An accelerator for attribute reduction in rough set theory. Artificial Intelligence 174(9-10), 597–618 (2010)
19. Stefanowski, J., Tsoukiàs, A.: On the extension of rough sets under incomplete information. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 73–82. Springer, Heidelberg (1999)
20. Yao, Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111(1-4), 239–259 (1998)
21. Zhang, J., Li, T., Ruan, D., Gao, Z., Zhao, C.: A parallel method for computing rough set approximations. Information Sciences (2012)
22. Zhang, J., Li, T., Ruan, D., Liu, D.: Neighborhood rough sets for dynamic data mining. International Journal of Intelligent Systems 27(4), 317–342 (2012)
23. Zhang, J., Li, T., Chen, H.: Composite rough sets. In: Lei, J., Wang, F.L., Deng, H., Miao, D. (eds.) AICI 2012. LNCS, vol. 7530, pp. 150–159. Springer, Heidelberg (2012)
24. 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)

GPU Implementation of MCE Approach to Finding Near Neighbourhoods*

Tariq Alusaifeer[1], Sheela Ramanna[1], Christopher J. Henry[1], and James Peters[2]

[1] Department of Applied Computer Science, University of Winnipeg, Winnipeg, Manitoba R3B 2E9 Canada
alusaifeer-t@webmail.uwinnipeg.ca,s.ramanna@uwinnipeg.ca, ch.henry@uwinnipeg.ca

[2] Computational Intelligence Laboratory, ECE Department, University of Manitoba Winnipeg, MB R3T 5V6
james.peters3@ad.umanitoba.ca

Abstract. This paper presents a parallel version of the Maximal Clique Enumeration (MCE) approach for discovering tolerance classes. Finding such classes is a computationally complex problem, especially in the case of large data sets or in content-based retrieval applications(CBIR). The GPU implementation is an extension of earlier work by the authors on finding efficient methods for computing tolerance classes in images. The experimental results demonstrate that the GPU-based MCE algorithm is faster than the serial MCE implementation and can perform computations with higher values of tolerance ε.

Keywords: CBIR, GPU, maximal clique enumeration, near sets, nearness measure, pre-class, tolerance near sets, tolerance space, tolerance relation.

1 Introduction

The focus of this article is on an efficient method for finding all tolerance classes on a set of objects using a parallel version of Maximal Clique Enumeration [5,2]. Tolerance classes are sets where all the pairs of objects within a set must satisfy the tolerance relation and the set is maximal with respect to inclusion [6,22,19]. Finding such classes is a computationally complex problem, especially in content-based image retrieval (CBIR) [25] involving sets of objects with similar features. In the proposed application to content-based image retrieval (CBIR), classes in image covers determined by a tolerance relation provide the content used in CBIR.

Tolerance near sets are near sets defined by a description-based tolerance relation [17,19]. In [6], a serial version for finding most tolerance classes using the Fast Library for Approximate Nearest Neighbours (FLANN) was used

* This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grants 194376, 418413 and 185986. Tariq Alusaifeer has been supported by Government of Saudi Arabia.The authors wish to thank Prof. J. Wang for sharing the SIMPlicity image database.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 251–262, 2013.

in the tolerance nearness measure (tNM) based on near set theory. In [11], a signature-based tNM was compared with the Earth Movers Distance (EMD) [23] and the Integrated Region Matching (IRM) [28]. Serial and parallel computing approaches for finding all tolerance classes using NVIDIA's Compute Unified Device Architecture (CUDA) Graphics Processing Unit (GPU) were reported in [9]. In [10], a new solution to the problem of finding tolerance classes was proposed. The solution was based on the observation that the problem of discovery of all tolerance classes can be mapped to the graph theory-based Maximal Clique Enumeration (MCE) problem. The experimental results demonstrated that the MCE algorithm has reduced complexity and is 10 times faster than the serial FLANN algorithm. In [8], the serial MCE approach was compared with the EMD and IRM approaches. The contributions of this article are i) a more effective GPU-based MCE algorithm for computing tolerance classes ii) ability to conduct experiments with higher values for ε.

The article is organized as follows: Section 2 introduces the foundations for this work namely: tolerance classes and nearness measure are used in this article. Section 5 provides a brief review MCE approach. Finally, Section 6 presents the results and discussion.

2 Foundation: Tolerance Classes and Nearness Measure

2.1 Tolerance Classes

Tolerance relations provide a view of the world without transitivity [26]. Consequently, tolerance near sets provide a formal foundation for *almost solutions*, solutions that are valid within some approximation, which is required for real world problems and applications [26]. The basic structure which underlies near set theory is a perceptual system [21]. A perceptual system is a specialised form of information system consisting of a set of objects equipped with a family of probe functions. The probe functions give rise to a number of perceptual relations between objects of a perceptual system [20,29].

A perceptual system $\langle O, \mathbb{F} \rangle$ consists of a non-empty set O of sample perceptual objects and a non-empty set $\mathbb{F}$ of real-valued functions $\phi \in \mathbb{F}$ such that $\phi : O \to \mathbb{R}$ [21]. Let $\mathcal{B} \subseteq \mathbb{F}$ be a set of probe functions. Then, the description of a perceptual object $x \in O$ is a feature vector given by

$$\boldsymbol{\phi}_{\mathcal{B}}(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_i(x), \ldots, \phi_l(x)),$$

where l is the length of the vector $\boldsymbol{\phi}_{\mathcal{B}}$, and each $\phi_i(x)$ in $\boldsymbol{\phi}_{\mathcal{B}}(x)$ is a probe function value that is part of the description of the object $x \in O$. Formally, a tolerance space can be defined as follows [30,26,22]. Let O be a set of sample perceptual objects, and let ξ be a binary relation (called a tolerance relation) on X ($\xi \subset X \times X$) that is reflexive (for all $x \in X$, $x \xi x$) and symmetric (for all $x, y \in X$, if $x \xi y$, then $y \xi x$) but transitivity of ξ is not required. Then a tolerance space is defined as $\langle X, \xi \rangle$. Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $\varepsilon \in \mathbb{R}_0^+$. For every $\mathcal{B} \subseteq \mathbb{F}$, the perceptual tolerance relation $\cong_{\mathcal{B},\varepsilon}$ is defined by:

$$\cong_{\mathcal{B},\varepsilon} = \{(x, y) \in O \times O : \| \boldsymbol{\phi}(x) - \boldsymbol{\phi}(y) \|_2 \leq \varepsilon\},$$

where $\|\cdot\|_2$ is the L^2 norm. Finally, the algorithms presented in this paper are based on the concepts of neighbourhoods and tolerance classes. Formally, these concepts are defined as follows. Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $x \in O$. For a set $\mathcal{B} \subseteq \mathbb{F}$ and $\varepsilon \in \mathbb{R}_0^+$, a neighbourhood is defined as

$$N(x) = \{y \in O : x \cong_{\mathcal{B},\varepsilon} y\}.$$

Note, all objects satisfy the tolerance relation with a single object in a neighbourhood. In contrast, all the pairs of objects within a pre-class must satisfy the tolerance relation. Thus, let $\langle O, \mathbb{F}\rangle$ be a perceptual system. For $\mathcal{B} \subseteq \mathbb{F}$ and $\varepsilon \in \mathbb{R}_0^+$, a set $X \subseteq O$ is a pre-class iff $x \cong_{\mathcal{B},\varepsilon} y$ for any pair $x, y \in X$. Similarly, a maximal pre-class with respect to inclusion is called a tolerance class.

2.2 Nearness Measure

The following two definitions enunciate the fundamental notion of nearness between two sets and provide the foundation for applying near set theory to the problem of CBIR.

Definition 1. Tolerance Nearness Relation [18,19]. *Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $X, Y \subseteq O, \varepsilon \in \mathbb{R}_0^+$. A set X is near to a set Y within the perceptual system $\langle O, \mathbb{F}\rangle$ ($X \underline{\underline{\bowtie}}_{\mathbb{F}} Y$) iff there exists $x \in X$ and $y \in Y$ and there is $\mathcal{B} \subseteq \mathbb{F}$ such that $x \cong_{\mathcal{B},\varepsilon} y$.*

Definition 2. Tolerance Near Sets [18,19]. *Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $\varepsilon \in \mathbb{R}_0^+, \mathcal{B} \subseteq \mathbb{F}$. Further, let $X, Y \subseteq O$, denote disjoint sets with coverings determined by the tolerance relation $\cong_{\mathcal{B},\varepsilon}$, and let $H_{\cong_{\mathcal{B},\varepsilon}}(X), H_{\cong_{\mathcal{B},\varepsilon}}(Y)$ denote the set of tolerance classes for X, Y, respectively. Sets X, Y are* tolerance near sets *iff there are tolerance classes $A \in H_{\cong_{\mathcal{B},\varepsilon}}(X), B \in H_{\cong_{\mathcal{B},\varepsilon}}(Y)$ such that $A \underline{\underline{\bowtie}}_{\mathbb{F}} B$.*

Observe that two sets $X, Y \subseteq O$ are tolerance near sets, if they satisfy the tolerance nearness relation. The tolerance nearness measure between two sets X, Y is based on the idea that tolerance classes formed from objects in the union $Z = X \cup Y$ should be evenly divided among X and Y if these sets are similar, where similarity is always determined with respect to the selected probe functions. The tolerance nearness measure is defined as follows. Let $\langle O, \mathbb{F}\rangle$ be a perceptual system, with $\varepsilon \in \mathbb{R}_0^+$, and $\mathcal{B} \subseteq \mathbb{F}$. Furthermore, let X and Y be two disjoint sets and let $Z = X \cup Y$. Then a tolerance nearness measure between two sets is given by

$$tNM_{\cong_{\mathcal{B},\varepsilon}}(X, Y) = 1 - \left(\sum_{C \in H_{\cong_{\mathcal{B},\varepsilon}}(Z)} |C|\right)^{-1} \cdot \sum_{C \in H_{\cong_{\mathcal{B},\varepsilon}}(Z)} |C| \frac{\min(|C \cap X|, |[C \cap Y|)}{\max(|C \cap X|, |C \cap Y|)}. \quad (1)$$

3 Maximal Clique Enumeration

Maximal Clique Enumeration (MCE) consists of finding all maximal cliques among an undirected graph, and is a well studied problem [2,4]. Briefly, let $G = (V, E)$ denote an undirected graph, where V is a set of vertices and E is set of edges that connect pairs of distinct vertices from V. A clique is a set of vertices where each pair of vertices in the clique is connected by an edge in E. A maximal clique in G is a clique whose vertices are not all contained in some larger clique, *i.e.* there is no other vertex that is connected to all the vertices in the clique by edges in E.

The first serial algorithm for MCE was developed by Harary and Ross [5,2]. Since then, two main approaches have been established to solve the MCE problem [4], namely the greedy approach reported by Bron-Kerbosh [3] (and concurrent discovery by E. Akkoyunlu [1]), and output-sensitive approaches such as those in [27,16]. Both implementations of the MCE algorithm in this paper are a modification of the Bron-Kerbosh approach. The CPU-based approach is a single system implementation of the algorithm reported in [24] (see [10] for further details), while the GPU version is a port of the approach reported by Bron-Kerbosh. Note, Jenkins *et al.* [12] explore the backtracking paradigm (*i.e.* depth-first search methods) for GPU architectures, and use the MCE problem as a case study. They report backtracking GPU algorithms are limited to 1.4-2.25 times a single CPU core. However, their results are for the general case, and the solution here is tailored to a specific CBIR problem. A discussion on the comparison of the GPU runtime versus a single CPU core is outside the scope of this paper.

The Bron-Kerbosh approach is given in Algorithm 1, where the general idea is to find maximal cliques through a depth-first search. Branches are formed based on candidate cliques, and backtracking occurs once a maximal clique has been discovered. Both algorithms mark new nodes and processes them, either sequentially or in parallel, where processing nodes consists of either identifying child nodes, or recording a maximal clique if a terminal node is discovered. Child nodes are identified through the use of three disjoint sets and a pivot vertex, v_p. In particular, pivot vertices are used to prune equivalent sub-trees appearing in different branches [4,14]; R is a set of vertices consisting of the (non-maximal) clique formed up to the currently selected pivot; P is a set of potential vertices that are connected to every vertex in R; and X is a set of vertices that are connected to every vertex in R, but, if selected as a pivot, would constitute a repeated maximal clique. Since our problem consists of a maximum of 456 vertices, these sets are represented as a series of 16 (4-byte) integers, where bit i indicates where $v_i \in V$ belongs to the set.

4 CPU-Based Approach

The CPU-based approach used for comparison is a single system implementation of the MCE algorithm reported in [24]. To simplify their implementation, our

Algorithm 1. The BK algorithm

Input : A graph G with vertex V and edge set E
Output: MCE for graph G

```
1  R ← {};
2  P ← V;
3  X ← {};
4  CliqueEnumerate(R, P, X);
```

Procedure CliqueEnumerate(R, P, X)

```
1  if P = {} then
2      if X = {} then
3          Output R
4  else
5      v_p ← The vertex in P that is connected to the greatest number of other
       vertices in P;
6      cur_v ← v_p;
7      while cur_v ≠ NULL do
8          X' ← All vertices in X that are connected to cur_v;
9          P' ← All vertices in P that are connected to cur_v;
10         R' ← R ∪ cur_v;
11         CliqueEnumerate(R', P', X');
12         X ← X ∪ cur_v;
13         P ← P \ cur_v;
14         if there is a vertex v in P that is not connected to v_p then
15             cur_v ← v;
16         else
17             cur_v ← NULL;
```

results were generated using a single process with multiple threads. The MCE algorithm uses a stack of structure, which contains the nodes in the tree and each thread process a single node at a time. The modified version of the algorithm in [24] is given in Algorithm 2.

Algorithm 2. The Multi-threaded BK algorithm

Input : A graph G with vertex V and edge set E
Output: MCE for graph G

```
1  for i = 0; i < num_threads; i++ do
2      Spawn thread T_i;
3      Have T_i run MCliqueEnumerate();
4  Wait for threads to finish processing;
```

```
Procedure MCliqueEnumerate
1  foreach vertex v_i assigned to the thread do
2      cp ← New candidate path node structure for v_i;
3      for v_j ∈ V do
4          if connected(v_i, v_j) then
5              if i < j then
6                  Vertex v_j is in cp's P set;
7              else
8                  Vertex v_j is in cp's X set;
9      Push cp onto shared stack;
10 while shared stack is not empty do
11     cur ← Pop a candidate path node structure from stack;
12     if cur's P and X lists are empty then
13         Output cur's compsub
14     else
15         Generate all cur's children (create child nodes and push onto stack);
```

5 Parallel MCE Implementation

This section presents the parallel GPU MCE Implementation. A GPU consists of many of cores, ranging from several hundred to several thousand. For instance, the GeForce GTX 460 used to generate the results in this paper consists of 336 CUDA cores, whereas a Tesla K20 contains 2496 cores. Under the Compute Unified Device Architecture (CUDA), these cores are organized into groups called Streaming Multiprocessors (SM). The code that is executed on these cores is called a kernel, and the abstraction that executes this code is called a thread. CPUs are designed to minimize thread latency through the use of a memory hierarchy based on caching data sets, while GPUs seek to maximize throughput of parallel applications by hiding latency using many more threads. In fact, for a GPU to be efficient, one must generate 1000s of threads for execution [13]. Threads are arranged into groups called blocks, and blocks are allocated to SM for execution. Threads and blocks[1] can be organized into 1, 2, or 3 dimensions.

Our approach consists of the six stages (Steps 1-7, excluding 3) given in Algorithm 3, where each stage consists of one or more kernel calls. Notice, the kernel calls within the loop at Step 3 are executed iteratively to avoid the irregular data access and load balancing problems reported in [12]. Specifically, for a given iteration, nodes on the stack[2] are processed by a series of GPU kernel calls located in the body of a loop. The result is that the number of thread blocks

[1] Note, only devices with computer capability greater than 2.0 can support 3 dimensional block organization.

[2] The data structure containing the nodes is called a stack, but nodes are processed in parallel.

is dynamic. Also, in Step 4, stack nodes are grouped into contiguous memory locations to ensure memory accesses are coalesced. Lastly, for most of the kernel calls, each block of threads processes one node; each row of threads in a block processes one vertex in a set (for example, one vertex in P for the case of finding the pivot); and each thread in a row processes a byte from a set.

The specifics of each step in Algorithm 3 are as follows. Step 1, is a parallel kernel for finding neighbourhoods, and is based on the matrix multiplication example from the CUDA SDK, where the main calculation is replaced by Euclidean distance (without the square root operator). A more detailed explanation of this procedure can be found in [9]. The output is a 456×456 boolean adjacency matrix where a 1 in row r_i of column c_j means object j belongs to the neighbourhood of object i. Next, as mentioned above, each set will have a maximum of 456 objects, thus the adjacency matrix can be represented as an array of length 4×512[3], where each series of four integers represents the neighbourhood of one object. This array is created in Step 2, and will be used to select pivots in each iteration in Step 4. Note, this choice of representation also enables coalesced access to the GPU global memory and allows for a higher compute to global memory access ratio due to the reduction in storage (over representing sets as, for example, a series of integers). Also, the stack is initialized in Step 2.

Step 4 consists of three kernel calls. First, for each node on the stack, v_p is identified as the node in P that is connected to the greatest number other vertices in P. These results are stored temporarily in global memory. Second, all nodes that generated a v_p (*i.e.* $P \neq \emptyset$) are identified. Third, based on this information, the nodes in the stack are reordered to ensure nodes to be processed in Step 6 are contiguous in global memory. Next, Step 5 is used to allocate memory for the stack size in the next iteration of the loop, with a maximum stack size of 512 (since the nature of MCE does not allow prediction of the search tree size). Then, Step 6 visits each node and generates new nodes (with new R, P, and X sets) using the v_p identified in Step 4. Finally, Step 7 outputs the set R as a tolerance class for any node where $P = \emptyset$ and $X = \emptyset$, and discards any nodes where $P = \emptyset$ and $|X| > 0$.

Algorithm 3. Main Loop Iteration for Parallel Algorithm

Input : O and $\phi_{\mathcal{B}}(x) \forall x \in O$
Output: Set of tolerance classes $H_{\cong_{\mathcal{B},\varepsilon}}(O)$
1 Generate neighbourhood matrix;
2 Reduce neighbourhood matrix to integers;
3 **while** *nodes in stack* **do**
4 Find v_p for each node on stack;
5 Calculate the number of output nodes for each node on stack;
6 Generate the new nodes resulting from choice of v_p;
7 Detect terminal nodes to extract maximal cliques;

[3] 512 is the next power of 2 larger than 456.

6 Results and Discussion

The algorithm presented here is compared using CBIR, where the goal is to retrieve images from databases based on the content of an image rather than on some semantic string or keywords associated with the image. The content of the image is determined by functions that characterize features such as colour, texture, shape of objects, and edges. In our approach to CBIR, a search entails analysis of content, based on the tNM nearness measure (see, *e.g.* [6]) between a query image and test image. Moreover, the nearness measure on tolerance classes of objects derived from two perspective images provides a quantitative approach for accessing the similarity of images. To generate our results, the SIMPLIcity image database [15], a database of images containing 10 categories with 100 images in each category, was used, where the dimensions of each image is either 384×256, or *vice versa*. To perform the experiment, each image in the database is compared to the 1000 images in the database, and ranked using the tNM nearness measure. Then, precision and recall values can be calculated based on this ranking and the category of each image in the ranked list.

Notice, results obtained using the tNM are dependent on the selection of ε used in the perceptual tolerance relation, which determines the covering for a set of objects (obtained from the images being compared). Recall, in any given application (regardless of the distance metric), there is always an optimal ε when performing experiments using the perceptual tolerance relation [6]. For instance, a value of $\varepsilon = 0$ produces little or no pairs of objects that satisfy the perceptual tolerance relation, and a value of $\varepsilon = \sqrt{l}$, means that all pairs of objects satisfy the tolerance relation[4]. Consequently, ε should be selected such that the objects that are relatively[5] close in feature space satisfy the tolerance relation, and the rest of the pairs of objects do not. The selection of ε is straightforward when a metric is available for measuring the success of the experiment. Thus, if runtime were not an issue, the value of ε should be selected based on the best result of the evaluation metric, which, in the context of CBIR, is the best results in terms of precision vs. recall.

The results were generated by partitioning the images into 228 subimages (using a size of 20×20 pixels), where each subimage was considered as an object in the near set sense, *i.e.* each subimage is a perceptual object, and each object description consists of the values obtained from image processing techniques on the subimage. This technique of partitioning an image, and assigning feature vectors to each subimage is an approach that has also been traditionally used in CBIR. Formally, an RGB image is defined as $f = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_T\}$, where $\mathbf{p}_i = (c, r, R, G, B)^{\mathrm{T}}$, $c \in [1, M]$, $r \in [1, N]$, $R, G, B \in [0, 255]$, and M, N respectively denote the width and height of the image and $M \times N = T$. Further, define a square subimage as $f_i \subset f$ such that $f_i \cap f_j = \{\}$ for $i \neq j$ and $f_1 \cup f_2 \ldots \cup f_s = f$,

[4] For normalized feature values, the largest distance between two objects occurs in the interval $[0, \sqrt{l}]$, where l is the length of the feature vectors.

[5] Here, distance of objects *that are relatively close* will be determined by the application.

where s is the number of subimages in f. Next, O can be defined as the set of all subimages, *i.e.*, $O = \{f_1, \ldots, f_s\}$, and $\mathbb{F}$ is a set of image processing descriptors or functions that operate on images. Then, the nearness of two images can be discovered by partitioning each of the images into subimages and letting these represent objects in a perceptual system, *i.e*, let the sets X and Y represent the two images to be compared where each set consists of the subimages obtained by partitioning the images. Then, the set of all objects in this perceptual system is given by $Z = X \cup Y$.

In the ideal case, all images from the same category would be retrieved before any images from other categories. In this case, precision would be 100% until recall reached 100%, at which point precision would drop to # of images in query category / # of images in the database. As a result, our final value of precision will be ∼11% since we used 9 categories each containing 100 images. Note, only 9 categories were used since category 4 is easy to classify since it consists of drawn dinasours, while the rest are natural images.

The results are presented in Table 1 and Fig. 1, where the average runtime to compare two images is given in Table 1, and the average precision vs. recall plots are given in Fig. 1. The results were generated using a system containing an Intel CORE i7-930 CPU, 6 GB of RAM, and a GeForce GTX 460 GPU containing 768 MB of RAM. First, notice the GPU MCE algorithm outperforms the CPU MCE implementation for all values of ε. Moreover, the CPU runtime is already prohibitively long using $\varepsilon = 0.3$. Recall, each CBIR test consists of $900 \times 901/2 = 405450$ comparisons. Thus, the total runtime for the CPU algorithm at $\varepsilon = 0.3$ is approximately 11.5 days (which can be reduced using multithreading and multiple CPU cores). Also, precision vs. recall results are not reported for $\varepsilon = 0.3$ for images from category 7 (see, *e.g.* Fig. 1(h)), since the runtime was too large for some of the images in this category. For instance, some image pairs produced in excess of 700,000 tolerance classes (on only 456 objects) and had run times of over 2 hours. Lastly, the run times for the CPU approach are almost the same for $\varepsilon = 0.1$ and $\varepsilon = 0.2$, which can be attributed to the overhead limiting the minimum runtime. This can be verified by considering the number of non-zero tNM values for a particular image query. For instance, image 704 produces 37 non-zero values for $\varepsilon = 0.1$, and 285 for $\varepsilon = 0.2$.

Next, Fig. 1 presents a comparison of the precision vs. recall for both approaches. Notice, the plots are the same, indicating that both approaches produce the same results for a given value of ε. Consequently, for equivalent run times, the GPU algorithm can produce results using a larger value of epsilon, which, as reported in [7] leads to improvement in CBIR results. In the case of the CPU implementation, results for $\varepsilon = 0.4$ are not given due to prohibitive run times.

Next, the following presents some observations of the reported results. First, notice that some of the curves have a sharp point of inflection (see, *e.g.*, $\varepsilon = 0.1$ close to 20% recall in Fig. 1(b)). These points represent the location at which the remaining tNM values for all query images in the category produce a tNM value of zero. In order to provide this clear demarcation, any images from the

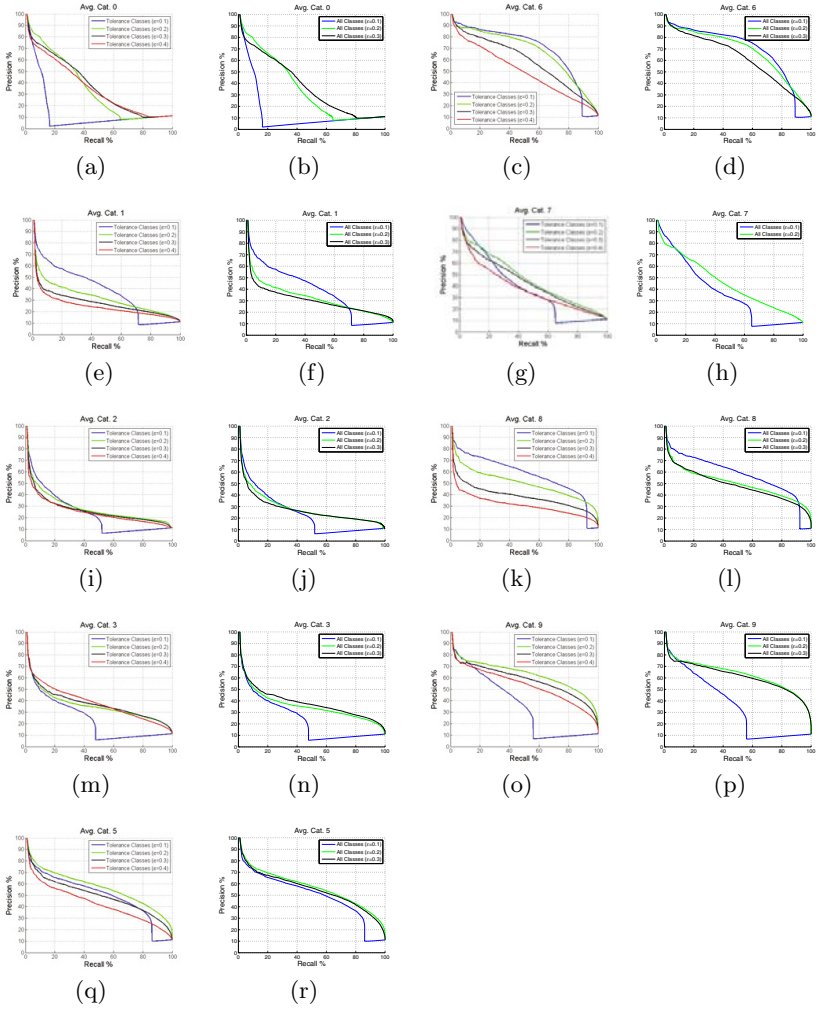


Fig. 1. Average precision versus recall plots grouped into four columns. From left to right: Col. 1 GPU results (Cat. 0-5), Col. 2 CPU results (Cat. 0-5), Col. 3 GPU results (Cat. 6-9), and Col. 4 CPU results (Cat. 6-9).

Table 1. Algorithm Runtimes

ε	CPU MCE (sec.)	GPU MCE (sec.)
0.1	0.85	0.10
0.2	0.84	0.06
0.3	2.45	1.42
0.4	28.71	3.39

same category as the query image that produced a tNM value of zero were ranked last in the search. Finally, the precision vs. recall results for $\varepsilon = 0.3$ are

better than $\varepsilon = 0.4$. These results suggest that the optimal value of ε is in the interval $[0.3, 0.4)$ for this application.

7 Conclusion

This article presents results in the context of CBIR, where perceptual information within the framework of near set theory is used to discern affinities between pairs of images. Specifically, perceptually relevant information was extracted from a set objects formed from pairs of images, where each object has an associated object description. It is the information contained in these feature vectors that is used to extract perceptual information represented by the discovered tolerance classes. The experimental results demonstrate that the GPU-based MCE algorithm is faster than the serial MCE implementation and is also able to perform computations with higher values of tolerance ε which in turn leads to better retrieval results.

References

1. Akkoyunlu, E.A.: The enumeration of maximal cliques of large graphs. SIAM Journal on Computing 2(1), 1–6 (1973)
2. Bomze, I., Budinich, M., Pardalos, P., Pelillo, M.: The maximum clique problem. In: Du, D.Z., Pardalos, P.M. (eds.) Handbook of Combinatorial Optimization, vol. 4. Kluwer (1999)
3. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. Communications of the ACM 16(9), 575–577 (1973)
4. Cazals, F., Karande, C.: A note on the problem of reporting maximal cliques. Theoretical Computer Science 407(1), 564–568 (2008)
5. Harary, F., Ross, I.C.: A procedure for clique detection using the group matrix. Sociometry 20(3), 205–215 (1957)
6. Henry, C.J.: Near Sets: Theory and Applications. Ph.D. thesis (2010)
7. Henry, C.J.: Perceptual indiscernibility, rough sets, descriptively near sets, and image analysis. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets XV. LNCS, vol. 7255, pp. 41–121. Springer, Heidelberg (2012)
8. Henry, C.J., Ramanna, S.: Signature-based perceptual nearness. Application of near sets to image retrieval. Mathematics in Computer Science, 71–85
9. Henry, C.J., Ramanna, S.: Parallel computation in finding near neighbourhoods. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 523–532. Springer, Heidelberg (2011)
10. Henry, C.J., Ramanna, S.: Maximal clique enumeration in finding near neighbourhoods. In: Peters, J.F., Skowron, A., Ramanna, S., Suraj, Z., Wang, X. (eds.) Transactions on Rough Sets XVI. LNCS, vol. 7736, pp. 103–124. Springer, Heidelberg (2013)
11. Henry, C.J., Ramanna, S., Levi, D.: Qantifying nearness in visual spaces. Cybernetics and Systems 44(1), 38–56 (2013)
12. Jenkins, J., Arkatkar, I., Owens, J.D., Choudhary, A., Samatova, N.F.: Lessons learned from exploring the backtracking paradigm on the GPU. In: Proceedings of the 17th International Conference on Parallel Processing, vol. II, pp. 425–437 (2011)

13. Kirk, D.B., Hwu, W.W.: Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann, Waltham (2013)
14. Koch, I.: Fundamental study: Enumerating all connected maximal common subgraphs in two graphs. Theoretical Computer Science 250(1-2), 1–30 (2001)
15. Li, J., Wang, J.Z.: Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(9), 1075–1088 (2003)
16. Makino, K., Uno, T.: New algorithms for enumerating all maximal cliques. In: Hagerup, T., Katajainen, J. (eds.) SWAT 2004. LNCS, vol. 3111, pp. 260–272. Springer, Heidelberg (2004)
17. Peters, J.F.: Near sets. General theory about nearness of objects. Applied Mathematical Sciences 1(53), 2609–2629 (2007)
18. Peters, J.F.: Tolerance near sets and image correspondence. International Journal of Bio-Inspired Computation 1(4), 239–245 (2009)
19. Peters, J.F.: Corrigenda and addenda: Tolerance near sets and image correspondence. International Journal of Bio-Inspired Computation 2(5), 310–318 (2010)
20. Peters, J.F., Naimpally, S.: Applications of near sets. Amer. Math. Soc. Notices 59(4), 536–542 (2012)
21. Peters, J.F., Wasilewski, P.: Foundations of near sets. Information Sciences 179(18), 3091–3109 (2009)
22. Peters, J.F., Wasilewski, P.: Tolerance spaces: Origins, theoretical aspects and applications. Information Sciences 195, 211–225 (2012)
23. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Proceedings of the 1998 IEEE International Conference on Computer Vision, pp. 59–66 (1998)
24. Schmidt, M.C., Samatova, N.F., Thomas, K., Byung-Hoon, P.: A scalable, parallel algorithm for maximal clique enumeration. Journal of Parallel and Distributed Computing 69, 417–428 (2009)
25. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), 1349–1380 (2000)
26. Sossinsky, A.B.: Tolerance space theory and some applications. Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications 5(2), 137–167 (1986)
27. Tsukiyama, S., Ide, M., Ariyoshi, H., Shirakawa, I.: A new algorithm for generating all the maximal independent sets. SIAM Journal on Computing 6, 505–517 (1977)
28. Wang, J.Z., Li, J., Wiederhold, G.: SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(9), 947–963 (2001)
29. Wolski, M.: Granular computing: Topological and categorical aspects of near and rough set approaches to granulation of knowledge. In: Peters, J.F., Skowron, A., Ramanna, S., Suraj, Z., Wang, X. (eds.) Transactions on Rough Sets XVI. LNCS, vol. 7736, pp. 34–52. Springer, Heidelberg (2013)
30. Zeeman, E.C.: The topology of the brain and the visual perception, pp. 240–256. Prentice Hall, New Jersey (1965)

FPGA in Rough Set Based Core and Reduct Computation

Tomasz Grześ, Maciej Kopczyński, and Jarosław Stepaniuk

Faculty of Computer Science
Bialystok University of Technology
Wiejska 45A, 15-351 Bialystok, Poland
{t.grzes,m.kopczynski,j.stepaniuk}@pb.edu.pl
http://www.wi.pb.edu.pl

Abstract. In this paper we propose a combination of capabilities of the FPGA based device and PC computer for data processing using rough set methods. Presented architecture has been tested on a random data. Obtained results confirm the significant acceleration of the computation time using hardware supporting rough sets operations in comparison to software implementation.

Keywords: Rough sets, FPGA, hardware, reduct, core.

1 Introduction

The theory of rough sets has been developed in the eighties of the twentieth century by Prof. Z. Pawlak. Rough sets are used as a tool for data analysis and classification as well as for the extraction of important characteristics that describe the objects. Rough sets allow dealing with uncertain and incomplete data, and due to its versatility, are widely used in various areas of life, including medicine, pharmacology, banking, market and stock research, process control, image and audio processing and exploration of the web.

There exist many software implementations of rough set methods and algorithms. However, they require significant amount of resources of a computer system and a lot of time for algorithms to complete, especially during processing large amount of data. This is an important obstacle to use rough set methods in the data analysis in computer systems because of time needed for operation to complete. This type of problem also exists in embedded systems with limited resources in terms of processor core's and memory clock frequency.

Field Programmable Gate Arrays (FPGAs) are a group of integrated circuits, whose functionality is not defined by the manufacturer, but by the user. The user can implement his own project of the specialized digital system in the FPGA structure. Significant facilitation in the project creation is the possibility of using a hardware description language, such as VHDL (Very High Speed Integrated Circuits Hardware Description Language), which allows and speeds up describing architecture and the functional properties of the digital system. All these features makes a hardware implementation of rough set method in FPGAs possible, and

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 263–270, 2013.

thanks to that they can be easily used in embedded systems, as well as in desktop systems to process huge amounts of data.

At the moment there is no comprehensive hardware implementation of rough set methods. In the literature one can find descriptions of concepts or partial rough set methods hardware implementations. The idea of sample processor generating decision rules from decision tables was described by Pawlak in [7]. Lewis, Perkowski and Jozwiak in [5] presented architecture of rough sets processor based on cellur networks described in [6]. Kanasugi and Yokoyama [2] developed a concept of hardware device capable of minimizing the large logic functions created on the basis of discernibility matrix. Tiwari, Kothari and Keskar [10] has presented the design for generating reduct from binary discernibility matrix. More detailed summary of the existing ideas and hardware implementations of rough set methods can be found in [3,4].

Solution proposed in this article is a combination of capabilities of the FPGA based device and PC computer for data processing using rough sets. Presented architecture has been tested on a random data. Obtained results confirm the significant acceleration of the computation time using hardware supporting rough set operations in comparison to software implementation.

The paper is organized as follows. In Section 2 we present some selected information about core and reduct calculation using discernibility matrix. The Section 3 is devoted to our system architecture. The Section 4 contains results of experiments.

2 Discernibility Matrix in Core and Reduct Computation

The notion of a discernibility matrix was introduced by Prof. A. Skowron. For a formal definition of the discernibility matrix see e.g. [8,9]. Both the rows and columns of the discernibility matrix are labeled by the objects. An entry of the discernibility matrix is the set that consists of all condition attributes on which the corresponding two objects have distinct values. If an entry consists of only one attribute, the unique attribute must be a member of core (for a formal definition of the core see e.g. [8,9]). It is possible for the core to be empty. This means that there is no indispensable attribute. Therefore, any single condition attribute in such a decision table (for a formal definition see e.g. [9]) can be deleted without altering the quality of approximation of classification. Core can be used as the starting point of reduct computation. Some attributes can be removed from the set of condition attributes but the information which we need from the decision table is not lost. We can compute some reduct based on discernibility matrix using the following observation: If a condition attribute appears more times in the discernibility matrix, then the more important the attribute might be.

Sketch of the algorithm
Input: a discernibility matrix
Output: Core and short reduct
Method:

1. Set: $Core = \emptyset$ and $Reduct = \emptyset$.
2. Step Core: For each attribute $a \in A$ if a is a singleton in some entry of discernibility matrix, then $Core = Core \cup \{a\}$. In a hardware every entry of discernibility matrix is tested if it is a singleton and for such entries is calculated bitwise OR to obtain a core.
3. Step Superreduct: Calculate the number of occurrences of each attribute in the discernibility matrix and choose the most frequent attribute e.g. b, $Reduct = \{b\}$. Now we are taking under consideration only those cells of discernibility matrix that are not containing b and find the most frequent attribute. We repeat this step until in every cell of discernibility matrix appears some attribute from *Reduct*. In a hardware a temporary reduct register bit is set at the position representing the most frequent attribute. For further calculations the entries from discernibility matrix where the bits are set at the same positions as in temporary reduct register are not taken.
4. Step Elimination: For each attribute $a \in Reduct$ if discernibility with respect to $Reduct \setminus \{a\}$ is the same as for *Reduct*, then $Reduct = Reduct \setminus \{a\}$.

Example 1. Let the set of objects $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ and the set of condition attributes $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$, see an illustrative decision table in Table 1.

Table 1. Illustrative decision table

$U/A \cup \{d\}$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	d
x_1	0	0	0	0	0	0	0	0
x_2	1	1	0	0	0	0	0	1
x_3	1	0	1	0	0	0	0	1
x_4	1	0	0	1	0	0	0	1
x_5	0	1	0	0	1	0	0	1
x_6	0	0	1	0	0	1	0	1
x_7	0	0	0	1	0	0	1	1

In Table 2 we present a discernibility matrix for the illustrative decision table from Table 1.

Table 2. Discernibility matrix for illustrative decision table

U/U	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	$\emptyset$	$\{a_1, a_2\}$	$\{a_1, a_3\}$	$\{a_1, a_4\}$	$\{a_2, a_5\}$	$\{a_3, a_6\}$	$\{a_4, a_7\}$
x_2	$\{a_1, a_2\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x_3	$\{a_1, a_3\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x_4	$\{a_1, a_4\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x_5	$\{a_2, a_5\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x_6	$\{a_3, a_6\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x_7	$\{a_4, a_7\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Using "Step Core" we obtain $Core = \emptyset$ and using "Step Superreduct" we obtain one superreduct $Reduct = \{a_1, a_2, a_3, a_4\}$. But, we can obtain shorter reducts, e.g. $Reduct = \{a_2, a_3, a_4\}$. This example shows, that "Step Elimination" is necessary for reduct calculations.

3 Hardware Solution Architecture

The hardware system was written in VHDL (*Very high speed integrated circuits Hardware Description Language*). The solution can be used for binary decision tables (tables with binary attributes only). The method of calculation of the discernibility matrix is implemented in *Discernibility Matrix Comparator Block*. This block also contains implementation of the core calculation method. Calculation of single reduct is implemented in *Reduct Evaluation Block*.

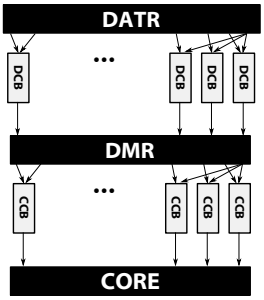


Fig. 1. Discernibility Matrix Comparator Block

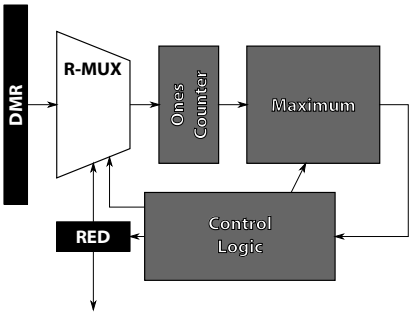


Fig. 2. Reduct Evaluation Block

3.1 Discernibility Matrix Comparator Block

The system design of the Discernibility Matrix Comparator Block (**DMCB**) is shown on Fig. 1. DMCB consists of Discernibility Comparator Block (**DCB**) and Core Comparator Block (**CCB**). DCB is designed to compare values of two objects from a decision table. CCB is designed to find the singletons from the discernibility matrix.

Decision table is passed to the DMCB through the DATa Register **DATR** and the results are stored in two registers:

- Discernibility Matrix Register (**DMR**) - a register for storing the discernibility matrix; size of this register is equal to $\frac{n(n-1)}{2}$ elements, where n is the number of objects in the decision table (in Example 1 size is equal to $\frac{7\cdot(7-1)}{2} = 21$ elements) and the size of each element (in bits) is equal to the number of the attributes,
- COre REgister (**CORE**) - a register that stores result of core calculation.

DMCB is designed as a combinational circuit and thus do not need a clock signal for proper work. Its functionality is basing on principles described in Section 2 (for Example 1 CORE register contains value 0). Amount of time needed to obtain correct results depends only on propagation time of logic blocks inside the FPGA. This property allows to significantly increase the speed of calculations because the time of propagation in contemporary FPGAs usually do not exceed 10 ns.

3.2 Reduct Evaluation Block

The system design of the Reduct Evaluation Block (**REB**) is shown on Fig. 2. REB consists of multiplexer R-MUX (for selecting appropriate attribute), Ones Counter (for counting the number of occurrences of an attribute), Maximum (for choosing the most common attribute, in Example 1 in first step the attribute a_1 is choosen) and Control Logic.

REB is connected to **DMR** register and has one output register - the REDuct register (**RED**), that stores calculated value of reduct.

REB is designed as sequential circuit (finite state machine) and its functionality is basing on principles described in Section 2.

4 Experimental Results

For the research purpose some of the rough set methods were implemented in C language. The main reason for choosing such language was deterministic program execution time, huge flexibility in the software creation, easiness of low-level communication implementation and the future plans of moving control program to the microprocessor independent from PC. The role of the microprocessor would be controlling operation of rough set hardware implementation modules. Microcontroller, due to the limited memory and computational resources in comparision to the PC, should not use additional runtime environments required by e.g. Java.

The results of the software implementation were obtained using a PC equipped with an Intel Core 2 Duo T9400 with 2.53 GHz clock speed running Windows XP Professional SP3. The source code of application was compiled using the GNU GCC 4.6.2 compiler. Given times are averaged for 10 000 of the algorithm runs with the same set of data.

The hardware implementation used single execution unit module covering the entire decision table. VHDL simulator and the development board equipped with an Altera FPGA were used during the research.

Table 3 presents the results of the time elapsed for software and hardware solutions for the calculating core and single reduct using exemplary randomly generated binary data sets. The core and reduct calculations were performed on discernibility matrix basis. k in the table denotes number of conditional attributes and n is the number of objects in decision table (in Example 1 $k = 7$ and $n = 7$).

Table 3. Comparision of execution time for calculating core and single reduct

Data size	**Software - t_S**		**Hardware - t_H**		$\frac{t_S}{t_H}$	
$(k+1) \times n$	Core $[\mu s]$	Reduct $[\mu s]$	Core $[\mu s]$	Reduct $[\mu s]$	Core —	Reduct —
8×8	53	75	3.05	4.75	17	16
9×9	77	105	3.43	5.54	22	19
10×10	103	144	3.81	6.38	27	23
11×11	148	202	4.19	4.19	35	28
12×12	197	263	4.57	4.57	43	32
13×13	245	325	4.95	9.15	50	36
14×14	341	452	5.33	10.16	64	44
15×15	423	563	5.72	11.22	74	50
16×16	525	698	6.10	12.32	86	57
32×32	5 234	7 484	12.19	64.58	429	116
40×48	16 875	24 531	18.29	100.06	923	245
48×64	38 750	56 719	24.38	141.95	1 589	400
64×64	56 250	87 500	31.25	198.54	1 800	441
128×128	467 187	996 875	48.77	872.70	9 580	1 142

Fig. 3 and Fig. 4 contains a graphs showing the relationship between the size of data (number of objects times the number of attributes) and execution time of calculating the core and single reduct in given data set in software and hardware implementation respectively. Data size axis has the logarithmic scale on both graphs.

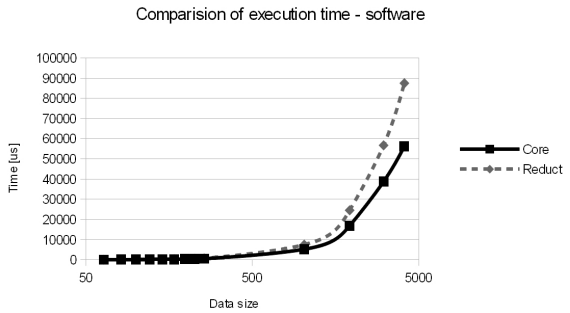


Fig. 3. Comparison of execution time for calculating core and single reduct in software

Results show a significant increase in the speed of data processing. Hardware module execution time compared to the software implementation is at least 1 order of magnitude shorter what is shown in Table 3 in columns $\frac{t_S}{t_H}$ and is increasing with larger data sets. Let comparison of objects' attribute value in the decision table or getting an attribute from the data container (e.g. linked list)

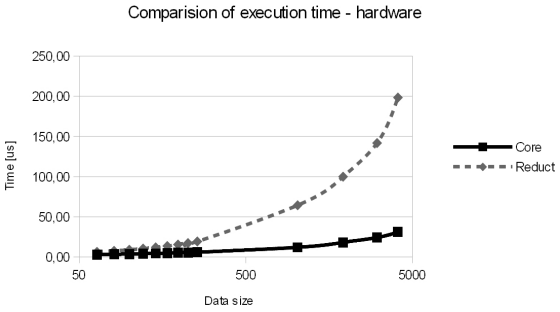


Fig. 4. Comparison of execution time for calculating core and single reduct in hardware

be an elementary operation. Let assume that hardware module is big enough to store the entire decision table.

First step of implemented algorithms is creating discernibility matrix. Using this matrix, the core and single reduct are calculated. Computational complexity of software implementation for the creating discernibility matrix is $\Theta(n^2k)$ and using hardware implementation, complexity is $\Theta(n)$, where k is the number of conditional attributes and n is the number of objects in decision table. Computational complexity of software implementation for the core calculation is $\Theta(n^2)$, while determining the single reduct, the complexity is $\Theta(n^2k^2)$. Using hardware implementation, complexity of core calculation is $\Theta(1)$, while complexity of single reduct calculation is $\Theta(k^2)$.

Of course, for most real data sets it will be impossible to create a single hardware structure capacious enough to store the entire data set. In this case, the input data set must be divided into a number of subsets, where each of them will be separately processed by a single hardware unit. The decomposition must be done in terms of objects and attributes. In such case, the computational complexity of software and hardware implementation will be similar, but in terms of time needed for data processing, hardware implementation will be still much faster than software implementation and increasing with larger data sets.

5 Conclusions and Future Research

The hardware implementation is the main direction of using scalable rough set methods in real time solutions. Software implementations are universal, but rather slow. Hardware realizations are deprived of this universality, however, allow us performing specific calculations in substantially shorter time.

As it was presented, performing calculations using hardware implementations of elementary rough sets methods - calculating discernibility matrix and determining cores and reducts, gives us a huge acceleration in comparison to software solution.

The system with hardware implementation of rough sets methods can be used in embedded systems such as industrial controllers or as an alternative and very

fast method of process control and data classification. The field of potential usage of the system can be very wide due to its versatility.

Further research will focus on developing methods for efficient storing discernibility matrices for larger data sets. The effort will be put also towards the hardware implementation of other elementary rough set methods. It is also required to improve the software control part of the entire system.

Acknowledgements. The research by T. Grzes is supported by the scientific grant S/WI/1/2013. The research by J. Stepaniuk is supported by the Polish National Science Centre under the grant 2012/07/B/ST6/01504. The research of Maciej Kopczynski is supported by the grant for young scientists W/WI/2/2012.

References

1. Athanas, P., Pnevmatikatos, D., Sklavos, N. (eds.): Embedded Systems Design with FPGAs. Springer (2013)
2. Kanasugi, A., Yokoyama, A.: A basic design for rough set processor. In: The 15th Annual Conference of Japanese Society for Artificial Intelligence (2001)
3. Kopczyński, M., Stepaniuk, J.: Rough set methods and hardware implementations, Zeszyty Naukowe Politechniki Białostockiej. Informatyka Zeszyt 8, 5–18 (2011)
4. Kopczyński, M., Stepaniuk, J.: Hardware Implementations of Rough Set Methods in Programmable Logic Devices. In: Skowron, A., Suraj, Z. (eds.) Rough Sets and Intelligent Systems. ISRL, vol. 43, pp. 309–321. Springer, Heidelberg (2013)
5. Lewis, T., Perkowski, M., Jozwiak, L.: Learning in Hardware: Architecture and Implementation of an FPGA-Based Rough Set Machine. In: 25th Euromicro Conference (EUROMICRO 1999), vol. 1, p. 1326 (1999)
6. Muraszkiewicz, M., Rybiński, H.: Towards a Parallel Rough Sets Computer. In: Rough Sets, Fuzzy Sets and Knowledge Discovery, pp. 434–443. Springer (1994)
7. Pawlak, Z.: Elementary rough set granules: Toward a rough set processor. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neurocomputing: Techniques for Computing with Words, Cognitive Technologies, pp. 5–14. Springer, Berlin (2004)
8. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
9. Stepaniuk, J.: Rough–Granular Computing in Knowledge Discovery and Data Mining. Springer (2008)
10. Tiwari, K.S., Kothari, A.G., Keskar, A.G.: Reduct Generation from Binary Discernibility Matrix: An Hardware Approach. International Journal of Future Computer and Communication 1(3), 270–272 (2012)

Fast Approximate Attribute Reduction with MapReduce

Ping Li, Jianyang Wu, and Lin Shang

Department of Computer Science and Technology,
Nanjing University, Nanjing 210023, China

Abstract. Massive data processing is a challenging problem in the age of big data. Traditional attribute reduction algorithms are generally time-consuming when facing massive data. For fast processing, we introduce a parallel fast approximate attribute reduction algorithm with *MapReduce*. We divide the original data into many small blocks, and use reduction algorithm for each block. The reduction algorithm is based on attribute significance. We compute the dependency of each reduction on testing data in order to select the best reduction. Data with different sizes are experimented. The experimental results show that our proposed algorithm can efficiently process large-scale data on *Hadoop* platform. In particular, on high dimensional data, the algorithm runs significantly faster than other latest parallel reduction methods.

Keywords: attribute reduction, MapReduce, rough set.

1 Introduction

Pawlak proposed rough set theory in 1982 [1]. It is a useful mathematical tool in uncertainty study. Attribute reduction is one of the most important issues in rough set theory, which removes redundant condition attributes and ensures the same classification ability. Varieties of attribute reduction algorithms based on rough sets have been proposed [2-7]. They are based on attribute significance [2-3], discernibility matrix [4-5], entropy [6-7], and etc.., among which the attribute significance-based method is efficient and easy to understand.

Internet companies analyze massive data sets coming from a variety of web applications every day. Analysis of massive data is becoming increasingly valuable for businesses. By analyzing data, companies can improve their service quality and detect changes in patterns over time according to the worked-out results. Due to the size of massive data, the complexity of data processing has also been increased. Traditional centralized data mining algorithms [8] were not able to process massive data efficiently. Parallel computing is one way to deal with large-scale data. There are many parallel computing technologies, such as *OpenMP* [9], *MPI*[10], and etc.. However low-level details must be considered when using these technologies. In this background, *Google* has distributed file system *GFS* [11] and parallel programming mode *MapReduce* [12]. *MapReduce* is a software programming framework and high-performance parallel computing platform for

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 271–278, 2013.

the large-scale data processing. The main idea is to divide and conquer. The task is divided into many sub-tasks for parallel computing. Then the system aggregates results which come from each sub-task.

Classic attribute reduction algorithm loads all the data into main memory to obtain reduction, which is not suitable for massive data processing. Many researchers have done a lot of work on large-scale data reduction. Most of them use parallel technology. Xiao et al.[13] have taken advantage of parallel computing. They divided the big task into many small tasks. Those small tasks were assigned to multiple processors at the same time, which significantly improved the efficiency. Liang et al.[14] have proposed a rough feature selection algorithm with a multi-granulation view. The algorithm first divided the large-scale data into different small granularities and then computed the reduction of each small granularity. Investigating all of the estimates on small granularities together, the algorithm could obtain an approximate reduction. However the algorithm must work with additional code to divide data. Qian et al.[15] have proposed a parallel algorithm, which used $MapReduce$ to divide data automatically. In their algorithm, the calculating of equivalence classes was parallelized with $MapReduce$. Four reduction algorithms were tested and compared in the aspect of running time. One of four algorithms is based on positive region, which had the disadvantage of time-consuming for high dimensional data.

Aiming to solve the time-consuming problem, we will put forward a parallel fast approximate attribute reduction algorithm with $MapReduce$ framework. We divide the original data into many small blocks, then use reduction algorithm for each block. At last, we calculate the dependency of each reduction on testing data in order to select the best reduction. Dependency of attribute sets is defined in Definition 7. Experiments show that our algorithm runs significantly faster than the newly proposed algorithm by Qian[15] on high dimensional data.

The paper is organized as follows: In section 2, we introduce the basic theory of rough sets. In section 3, we introduce the parallel fast approximate reduction algorithm in detail. In section 4, we use twelve data sets to illustrate the feasibility and efficiency of our proposed algorithm. In the last, we draw a conclusion.

2 Preliminaries

In this Section, we will present some basic knowledge about rough sets [16].
Definition 1. A decision table is defined as $S = \langle U, C, D, f\rangle$, where U is the domain. $A = C \cup D$ is the attribute set, among which C is the condition attribute set and D is the decision attribute set, at the same time, $C \cap D = \phi$. $V = \bigcup_{\alpha \in U} V_\alpha$ is the set of attribute values. $f : U \times (C \cup D) \to V$ is a function, which gives attribute a its value.

Definition 2. For decision table $S = \langle U, C, D, f\rangle$, attribute subset $P \subseteq (C \cup D)$ determines an indiscernibility relation in the following way:

$$IND(P) = \{(x, y) \in U \times U | \forall a \in P, f(x, a) = f(y, a)\}. \quad (1)$$

Definition 3. Suppose $S = \langle U, C, D, f\rangle$ is a decision table, $P \subseteq (C \cup D)$, $Q \subseteq (C \cup D)$, we define

$$POS_P(Q) = \bigcup_{X \in U/Q} P\_(X). \tag{2}$$

$POS_P(Q)$, called a positive region of the partition U/Q with respect to P, is the set of all elements of U that can be uniquely classified into partitions of U/Q, by means of P. $P\_(X) = \{x \in U : [x]_P \subseteq X\}$ called *P-lower* approximation. $[x]_p$ is an equivalence class of x concerning P.

Definition 4. According to the understanding of the definition of the positive region, Liu gives an equivalent definition of the positive region[17]:

$$POS_P(Q) = \bigcup_{Y \in U/P, |Y/Q|=1} Y. \tag{3}$$

$U/P = \{Y_1, Y_2, ...Y_n\}$. For Y_i, $(i = 1, 2, ...n)$, calculate $|Y_i/Q|$, if $|Y_i/Q| = 1$, add Y_i into $POS_P(Q)$. In our reduction algorithm, we use this definition and approach to calculate the positive region.

We can say that attribute $a \in C$ is *D-dispensable* in C, if $POS_C(D) = POS_{(C-\{a\})}(D)$, otherwise the attribute a is *D-indispensable* in C. If all attributes $a \in C$ are *C-indispensable* in C, then C will be called *D-independent*.

Definition 5. Subset $C' \subseteq C$ is a *D-reduct* of C, iff C' is *D-independent* and

$$POS_C(D) = POS_{C'}(D). \tag{4}$$

Definition 6. The significance of attribute a, $a \in C$, is defined by

$$Sig_a = \frac{|POS_{\{a\}}(Q)|}{|U|}, \tag{5}$$

where $Q \subseteq D$, U is the domain.

Definition 7. The dependency of attributes P, $P \subseteq C$, is defined by

$$\gamma(P, Q) = \frac{|POS_{\{P\}}(Q)|}{|U|}, \tag{6}$$

where $Q \subseteq D$, U is the domain.

3 A Fast Approximate Attribute Reduction Algorithm with *MapReduce*

With *MapReduce*, we can divide the huge amounts of data into small blocks, and assign each small block to the Mapper node. In each Mapper node, it reduces the attributes using the reduction algorithm based on attribute significance. In each Reducer node which accepts the reduction results from Mapper nodes, we calculate the dependency of reductions on testing data in order to estimate which is better. Reducer nodes output the final results in the form like $\langle reduction, dependecy \rangle$, where *reduction* is the reduction result from mapper node and *dependency* is the dependency of each reduction on testing data.
In Algorithm 1, we describe attribute reduction algorithm based on the attribute significance.

Algorithm 1. Attribute reduction algorithm based on the attribute significance.

Input:

Decision table $S = (U, C, D, V, f)$.

Output:

Reduction.

1: Compute significance of all the condition attributes according to Definition 6 and store them in an array.
2: Sort the attributes according to the significance.
3: Given $red = \phi$. $Left = sortedArray$.
Calculate the dependency of condition attributes C on decision attributes D, denoted by k.
4: Calculate the dependency of decision attributes D on red, denoted by k_{temp}.
if $k_{temp} == k$ goto 5,
otherwise Select the attribute a which has the greatest significance from $Left$ sets, $red = red \cup \{a\}$. $Left = Left - \{a\}$.
Repeat 4.
5: Output reduction results.

Based on the formula of attribute significance defined in Definition 6, we select one attribute which has the greatest significance to put into reduction set each time until the dependency of all the condition attributes on decision attribute equals the dependency of reduction on decision attribute.
We describe the parallel reduction algorithm in detail. Algorithm 2 illustrates function Map and Algorithm 3 illustrates function *Reduce*.

Algorithm 2. map()

Input:

$\langle Null, S_i \rangle$ where the key is $Null$, the value is $S_i = (U_i, C, D, V, f)$.

Output:

$\langle red_i, 1 \rangle$ where the key is red_i which is the reduction of S_i, the value is 1.

1: calculate reduction red_i for S_i according to algorithm 1.
2: output $\langle red_i, 1 \rangle$.

Algorithm 3. reduce()

Input:

$\langle red, < 1, 1, 1, ..., 1 > \rangle$ where the key is red which is the reduction result from Mapper nodes, the value is the list of values that have same reduction.

Output:

$\langle red, dependency \rangle$ where the key is red, the value is $dependency$.

1: according to Definition 7, calculate the dependency of red on testing data.
2: output $\langle red, dependency \rangle$

Fig 1 shows how our parallel algorithm works with $MapReduce$ framework.

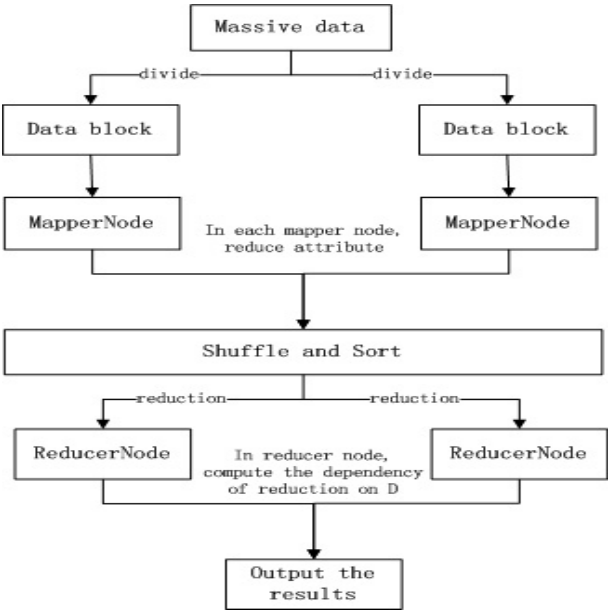


Fig. 1. The flow of parallel algorithm with *MapReduce*

4 Experiments

4.1 Experiments Setup

Our experimental environment is as follows: Hardware environment: Inter(R) Core(TM) 2 Duo, 2.4GHz, 4GB memory; Software environment: the operation system is *Ubuntu* 11.04; *Hadoop* cluster configuration: the version of *Hadoop* is 1.0.4 with 1 master node and 8 slave nodes.

The experiments have 3 steps. Experiment 1 aims to investigate the relationship between number of samples and running time. Experiment 2 aims to investigate the relationship between number of condition attributes and running time. Experiment 3 aims to investigate the relationship between number of slave nodes and running time.

4.2 Data Sets

We carried out three groups of experiments. In experiment 1, node sizes and condition attribute sizes are unchanged while sample sizes are increased; In experiment 2, node sizes and sample sizes are unchanged while condition attribute sizes are increased; In experiment 3, sample sizes and condition attribute sizes are unchanged while node sizes are increased. We use artificial data sets to test the performance of our parallel algorithm, and use twelve data sets to test each experiment. Data sets are as follows:

Table 1. Data Sets and Nodes Configuration Table For Experiments

Data Set	Sample Size	Attribute size	Node Size	Data Size	Experiments
DS1	200	2000	8	2.45MB	1
DS2	2000	2000	8	24.5MB	1
DS3	20000	2000	8	245MB	1
DS4	200000	2000	8	2.45GB	1
DS5	2000	200	8	2.08MB	2
DS6	2000	2000	8	24.5MB	2
DS7	2000	20000	8	403MB	2
DS8	2000	200000	8	3.14GB	2
DS9	20000	20000	1	2.77GB	3
DS10	20000	20000	2	2.77GB	3
DS11	20000	20000	4	2.77GB	3
DS12	20000	20000	8	2.77GB	3

4.3 Experiment Results

Our experimental results are as follows. From the results of experiments, we can see that the numbers of attributes after reduction are about 5 to 9 and the classification accuracy of Algorithm 1 is almost close to 1. That is partly for the reason of the generated random data.

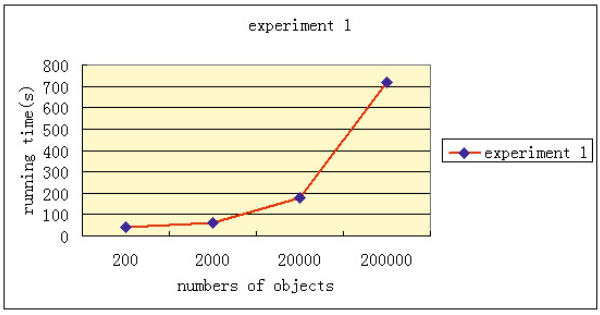


Fig. 2. Result with Increasing Sample Size

In Figure 2, when the number of samples is less than 20000, the running time is less than 200s. When the number increases to 200000, with the total data size about 2.45GB, running time is about 700s, which is still acceptable.

From Figure 3, we can see that when number of condition attributes is 20000, the running time is about 100 seconds. However in [15], when the number of condition attributes is 5000, the running time of algorithm based on positive region is over 5 hours. The speed-up effect of our parallel fast approximate reduction algorithm can be seen significantly. When number of condition attributes increases to 200000, the running time is about 700 seconds, which is very encouraging.

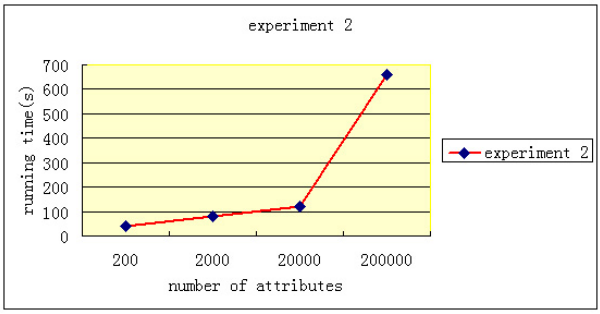


Fig. 3. Result with Increasing Attribute Size

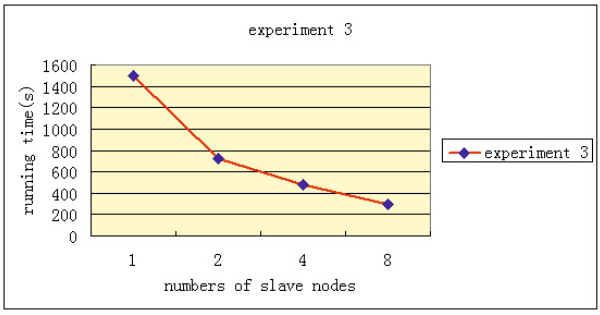


Fig. 4. Result with Increasing Nodes Size

From Figure 4, we know that with the the number of nodes increasing twice, running time reduces almost close to twice. This is because of the network delay and data loading time. We can not improve speed-up times to catch up with increasing times of nodes. However we have already reduced the running time to about 4 minutes for DS9 which contains about 2.77GB Data on 8 nodes. From our experiments results, it is seen that our proposed method can work on large-scale data efficiently. Especially on high-dimensional data, our method has outstanding performance.

5 Conclusion

This paper has presented a parallel fast approximate attribute reduction algorithm using *MapReduce*. Traditional stand-alone algorithm has not been suitable for processing massive data. We take advantage of *MapReduce* to obtain reduct. The algorithm is based on attribute significance. We propose a parallel algorithm based on *MapReduce*. Experimental results show that the parallel algorithm is effective and more efficient on large-scale data.

Acknowledgements. This work is supported by the National Science Foundation of China (NSFC No. 61170180, NSFC No. 61035003).

References

1. Pawlak, Z.: Rough sets. International Journal of Computer and Information Science 11(5), 341–356 (1982)
2. Wu, J., Zou, H.: Attribute reduction algorithm based on importance of attribute value. Computer Applications and Software 27(2), 255–257 (2010)
3. Kong, L.S., Mai, J.Y., Mei, S.K., Fan, Y.J.: An Improved Attribute Importance Degree Algorithm Based on Rough Set. In: Proceedings of IEEE International Conference on Progress in Informatics and Computing (PIC), pp. 122–126 (2010)
4. Skonwron, A., Rauszer, C.: The Discernibility Matrices and Functions in Information Systems. In: Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, pp. 331–362. Springer Netherlands Publisher (1992)
5. Qian, J., Miao, D.Q., Zhang, Z.H., Li, W.: Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation. International Journal of Approximate Reasoning 52(2), 212–230 (2011)
6. Miao, D.Q., Hu, G.R.: A heuristic algorithm for reduction of knowledge. Journal of Computer Research and Development 36(6), 681–684 (1999)
7. Wang, G.Y., Yu, H., Yang, D.C.: Decision table reduction based on conditional information entropy. Chinese Journal of Computers 25(7), 759–766 (2002)
8. Han, J., Kamber, M.: Data mining-concepts and techniques. Morgan Kaufmann Press (2006)
9. Dagum, L., Menon, R.: OpenMP: an industry standard API for shared-memory programming. In: Proceedings of IEEE International Conference on Computational Science and Engineering, vol. 5(1), pp. 46–55 (1998)
10. Gropp, W., Lusk, E., Skjellum, A.: Using MPI: Portable Parallel Programming with the Message Passing Interface. The MIT Press (1999)
11. Ghemawat, S., Gobioff, H., Leung, S.T.: The Google file system. Proceedings of ACM SIGOPS on Operating Systems Review 37(5), 29–43 (2003)
12. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Communications of the ACM 51(1), 107–113 (2008)
13. Xiao, D.W., Wang, G.Y., Hu, F.: Fast Parallel Attribute Reduction Algorithm Based Rough Set Theory. Journal of Computer Science 36(3) (2009) (in Chinese)
14. Liang, J.Y., Wang, F., Dang, C.Y., Qian, Y.H.: An efficient rough feature selection algorithm with a multi-granulation view. International Journal of Approximate Reasoning 53(6), 912–926 (2012)
15. Qian, J., Miao, D.Q., Zhang, Z.H., Zhang, Z.F.: Parallel Algorithm Model for Knowledge Reduction Using MapReduce. Journal of Frontiers of Computer Science and Technology 7(1), 35–44 (2013)
16. Zeng, H.L.: Rough set theory and its applications - a new method of data reasoning(revised edition). Chongqing University Press (1996)
17. Liu, S.H., Sheng, Q.J., Wu, B., Shi, Z.Z., Hu, F.: Research on Efficient Algorithm of Rough Sets. Chinese Journal of Computers 26(5) (2013)

Three-Way Decision Based Overlapping Community Detection

Youli Liu[1,2], Lei Pan[1,2], Xiuyi Jia[3], Chongjun Wang[1,2], and Junyuan Xie[1,2]

[1] Department of Computer Science and Technology, Nanjing University, Nanjing, China
[2] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
{liuyoulilyl,panleipanlei}@gmail.com,{chjwang,jyxie}@nju.edu.cn
[3] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
jiaxy@njust.edu.cn

Abstract. The three-way decision based overlapping community detection algorithm (OCDBTWD) divides the vesting relationship between communities into three types: completely belong relation, completely not belong relation and incompletely belong relation, and it uses the positive domain, negative domain and boundary domain to describe those vesting relationships respectively. OCDBTWD defines the similarity between communities to quantify the conditional probability when two communities have the vesting relationship, and uses the increment values of extended modularity to reflect the inclusion ratio thresholds. OCDBTWD uses the three-way decision to decide the vesting relationship between communities to guide the merger of them. When the vesting relationship between communities is incompletely belong relation, then the overlapping vertex detection algorithm (OVDA) is proposed to detect overlapping vertices. OCDBTWD has been tested on both synthetic and real world networks and also compared with other algorithms. The experiments demonstrate its feasibility and efficiency.

Keywords: overlapping community detection, three-way decision, social network.

1 Introduction

In the real world, a lot of things are presented in the form of social networks. Such as the World Wide Web network, metabolic network, genetic network, criminal network and proteins interaction network. A large number of studies shown that social networks have the properties of small world [1], scale-free [2] and community structure [3]. The community is a "cluster" which formed by a group of nodes. The internal connections of "cluster" are intensive while the external connections are extensive. Community detection has great theoretical significance and practical application value, such as proteins which have the same function are easy to form protein groups in proteins interaction networks [4-6], the financial crimes [7] and outliers [8] can be identified based on community structure.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 279–290, 2013.

How to use computer to effectively imitate human intelligence is an important problem and many researches have done a lot of works on that. Yao extended the algebraic inclusion relation in Pawlak algebraic rough sets model [15] into probability inclusion relation and proposed the decision-theoretic rough set (DTRS) model [16]. The three-way decision is the core of DTRS, and it divides the whole domain of discourse into three parts: positive domain, negative domain and boundary domain. It effectively resolved the completeness of the general decision class.

In the real social networks, some vertices belong to more than one communities. Such as a person not only belongs to the community formed by his family members, but also belongs to the community formed by his colleagues. Because of the existence of the overlapping vertices in social networks, the vesting relationship between communities can be divided into three types: completely belong relation, completely not belong relation and incompletely belong relation. The positive domain, negative domain and boundary domain of three-way decision can clearly describe those vesting relationship. So we use the positive domain, negative domain and boundary domain to reflect the completely belong relation, completely not belong relation and incompletely belong relation respectively. When the vesting relationship between two communities is completely belong relation, then merge them; when the vesting relationship is completely not belong relation, then do nothing; when the vesting relationship is incompletely belong relation, that means there are overlapping vertices. This paper proposes the three-way decision based overlapping community algorithm (OCDBTWD) to detect overlapping communities in social networks. OCDBTWD initializes every vertex to a community firstly, then uses the three-way decision to decide the vesting relationship between two communities to guide communities merged. When overlapping vertices existed, we propose the overlapping vertex detection algorithm (OVDA) to detect overlapping vertices, and thus detects overlapping communities in social networks finally.

This paper is organized as follows. Section 2 presents the related work. Section 3 introduces the three-way decision. Section 4 describes some definitions, the OCDBTWD and OVDA. Section 2 conducts experiments on synthetic and real world networks, and analyst the experiments' results. Finally, Section 6 concludes the paper.

2 Related Work

Many researchers have proposed algorithms to detect community structure in social networks. Newman and Girvan proposed the modularity function (Q function) to evaluate the community structure, and the GN algorithm [9] and FN algorithm [10] are all based on it. In order to detect overlapping communities, Palla et al. proposed the clique percolation theory and CPM algorithm [11]. A series of overlapping community detection algorithms have been proposed based on Palla clique percolation theory, such as GEC [12], EAGLE [13], LFM [14] and so on.

Yao proposed the Decision-Theoretic Rough Sets (DTRS) model [16]. After this, many researchers have done further research on it. Yao and Zhao pointed out some characteristic which needs to stay the same in attribute reductions of DTRS and

proposed the attribute reductions theory [17]; Yao, Liu and Li et al. researched on the three-way decision semantic in DTRS and proposed the three-way decision rough set model [18-20, 23]. The three-way decision excellently simulates the thinking mode of human intelligence to resolve the practical problems and was widely used in the real life. Such as the risk preferences of decision-making [21, 22], oil exploration decision [23], text classification [24], automatic clustering [25-27].

We uses the positive domain, negative domain and boundary domain in the three-way decision to quantify the vesting relationship between two communities when they are in the certain state. We define the similarity between communities to reflect the conditional probability when they are in the state and use the increment values of extended modularity to reflect inclusion ratio thresholds. And then the three-way decision is used to decide the vesting relationship between communities to guide the merger of them, and detect the overlapping communities in social networks.

3 Three-Way Decision

In the three-way decision [18-20], the state set is $\Omega = \{S, \vec{S}\}$, where S and $\vec{S}$ are complementary. The actions set is $A = \{a_P, a_N, a_B\}$, where a_P, a_N, a_B represent the actions which decide the object to $POS(X)$, $NEG(X)$ and $BND(X)$ respectively, where $POS(X), NEG(X)$ and $BND(X)$ represent positive domain, negative domain and boundary domain respectively. $\lambda_{PP}, \lambda_{NP}, \lambda_{BP}, \lambda_{PN}, \lambda_{NN}, \lambda_{BN}$ represent the loss function values when the decision actions are a_P, a_N, a_B and the object is in state of S and $\vec{S}$ respectively. $P(S|X)$ and $P(\vec{S}|X)$ represent the conditional probability when the object X is in state of S and $\vec{S}$ respectively. The expectation risk loss values of the actions a_P, a_N, a_B are shown respectively as formulas (1) to (3) below:

$$R(a_P|X) = \lambda_{PP}P(S|X) + \lambda_{PN}P(\vec{S}|X) \quad (1)$$

$$R(a_N|X) = \lambda_{NP}P(S|X) + \lambda_{NN}P(\vec{S}|X) \quad (2)$$

$$R(a_B|X) = \lambda_{BP}P(S|X) + \lambda_{BN}P(\vec{S}|X) \quad (3)$$

Where $R(a_P|X), R(a_N|X), R(a_B|X)$ denote the expectation risk loss values of the actions a_P, a_N, a_B respectively. According to the Bayesian decision procedure, the minimum-risk decision rules are shown as formulas (4) to (6) below:

If $(R(a_P|X) \le R(a_N|X)$ and $R(a_P|X) \le R(a_B|X))$ then decide $POS(X)$; (4)

If $(R(a_N|X) \le R(a_P|X)$ and $R(a_N|X) \le R(a_B|X))$ then decide $NEG(X)$; (5)

If $(R(a_B|X) \le R(a_P|X)$ and $R(a_B|X) \le R(a_N|X))$ then decide $BNG(X)$. (6)

According to the loss function means in real life, we can obtain those tions: $\lambda_{PP} \le \lambda_{BP} \le \lambda_{NP}, \lambda_{NN} \le \lambda_{BN} \le \lambda_{PN}$. And $P(S|X) + P(\vec{S}|X) = 1$ can be ob-

tained because of the complementation of S and $\vec{S}$. According to those formulas, the decision rules are shown respectively as formulas (7) to (9) below:

$$\text{If } (P(S|X) \geq \gamma \text{ and } P(S|X) \geq \alpha) \text{ then decide } POS(X); \tag{7}$$

$$\text{If } (P(S|X) \leq \beta \text{ and } P(S|X) \leq \gamma) \text{ then decide } NEG(X); \tag{8}$$

$$\text{If } (P(S|X) \geq \beta \text{ and } P(S|X) \leq \alpha) \text{ then decide } BNG(X). \tag{9}$$

Where α, β, γ are the inclusion ratio thresholds, the method to calculate them as follows:

$$\alpha = \frac{\lambda_{PN}-\lambda_{BN}}{(\lambda_{PN}-\lambda_{BN})+(\lambda_{BP}-\lambda_{PP})}, \beta = \frac{\lambda_{BN}-\lambda_{NN}}{(\lambda_{BN}-\lambda_{NN})+(\lambda_{NP}-\lambda_{BP})}, \gamma = \frac{\lambda_{PN}-\lambda_{NN}}{(\lambda_{PN}-\lambda_{NN})+(\lambda_{NP}-\lambda_{PP})}.$$

Assume that $(\lambda_{PN} - \lambda_{BN})(\lambda_{NP} - \lambda_{BP}) > (\lambda_{BP} - \lambda_{PP})(\lambda_{BN} - \lambda_{NN})$, thus $\alpha > \beta$. Then the $\alpha > \gamma > \beta$ can be derived from it. Thus the variations of decision rules are shown as formulas (10) to (12) below:

$$\text{If } (P(S|X) \geq \alpha) \text{ then decide } POS(X); \tag{10}$$

$$\text{If } (P(S|X) \leq \beta) \text{ then decide } NEG(X); \tag{11}$$

$$\text{If } (\beta < P(S|X) < \alpha) \text{ then decide } BNG(X). \tag{12}$$

4 Algorithm

4.1 Related Definitions

Given a network $G = (V, E)$, where V denotes the vertex set, E denotes the edge set. Assume that the community set is $C = \{c_1, c_2, \ldots, c_k\}$, where $c_1 \cup c_2 \cup \ldots \cup c_k = V$, and $\exists i, j$ such that $c_i \cap c_j \neq \emptyset$. The purpose of overlapping community detection is to detect the community set C. In order to describe this problem better, we give the following definitions.

Definition 1 (The vesting relationship between communities). Assume that c_i, c_j are the communities in social network, then there are three types of vesting relations between c_i and c_j and they are defined as follows:

1. When $c_i \subseteq c_j$, then the vesting relationship between c_i and c_j is called completely belong relation, and we use the positive domain to depicts it;
2. When $c_i \cap c_j = \emptyset$, then the vesting relationship between c_i and c_j is called completely not belong relation, and we use the negative domain to depicts it;
3. When $c_i \cap c_j \neq \emptyset, c_i \cap c_j \neq c_i$ and $c_i \cap c_j \neq c_j$, then the vesting relationship between c_i and c_j is called incompletely belong relation, and we use the boundary domain to depicts it.

Definition 2 (The similarity between vertices). $\Gamma(v_i), \Gamma(v_j)$ denote the neighborhoods of the vertiecs v_i and v_j, $SVV(v_i, v_j)$ denotes the similarity between v_i and v_j, then $SVV(v_i, v_j)$ is defined as:

$$SVV(v_i, v_j) = \frac{|\Gamma(v_i) \cap \Gamma(v_j)|}{|\Gamma(v_i) \cup \Gamma(v_j)|} . \tag{13}$$

Definition 3 (The similarity between communities). Assume that c_i and c_j denote two communities, $SCC(c_i, c_j)$ denotes the similarity between c_i and c_j, then $SCC(c_i, c_j)$ is defined as:

$$SCC(c_i, c_j) = \frac{\sum_{v_n \in c_i, v_m \in c_j} SVV(v_n, v_m)}{\sqrt{|c_i||c_j|}}. \tag{14}$$

Where $|c_i|$ and $|c_j|$ are the total number of vertices in c_i and c_j respectively. The similarity between communities reflects the compactness of two communities. The bigger the value of $SCC(c_i, c_j)$ is, the more compact c_i and c_j are.

Definition 4 (The membership ratio between vertex and community). Assume that $c_j$denotes a community in social network, v_i denotes a vertex, $MVC(v_i, c_j)$ denotes the membership ratio between v_i and c_j, then $MVC(v_i, c_j)$is defined as:

$$MVC(v_i, c_j) = \frac{\sum_{v \in c_j} SVV(v, v_i)}{|c_j|}. \tag{15}$$

The membership ratio between vertex and community depicts the ratio of one vertex belonging to the community from the numerical point. And its value is the average of the similarity between the vertex and the all vertices in the community.

4.2 Objective Function

The modularity function (Q function) proposed by Newman and Girvan [9] is an effective method to quantify the strength of community structure in social networks. Though Q function is very popular in community detection, it is unsuited to overlapping community detection. Shen et al. proposed the extended modularity (EQ function) which based on Q function to quantify the strength of overlapping community [13]. EQ function is defined as:

$$EQ = \frac{1}{2m} \sum_C \sum_{i,j \in C} \frac{1}{O_i O_j} \left(A_{ij} - \frac{k_i k_j}{2m}\right). \tag{16}$$

Where i and j are two arbitrary vertices, O_i and O_j are the total numbers of communities which i and j belong to respectively, A_{ij} is the adjacency matrix, m is the total number of edges. k_i is the degree of vertex i and $k_i = \sum_j A_{ij}$. EQ function quantifies the strength of overlapping community structure in social networks. A higher value of EQ indicates a significant overlapping community structure. We use the EQ function as the objective function when using the three-way decision to decide the vesting relationship between communities in this paper.

4.3 The Method to Calculate the Conditional Probability and Inclusion Ratio Thresholds

In Section 3, we have provided the decision rules of three-way decision (seen as formula (10) to (12)). The conditional probability $P(S|X)$ and the inclusion ratio thresholds α and β are the keys to decide the vesting relationship between communities. The similarity between communities corresponds to the closeness of two communities. The bigger the value is, the closer two communities are. The vesting relationship between communities also reflects the closeness of two communities. So we use the similarity between communities to quantify the $P(S|X)$, that is, when communities c_i and c_j in the state S, $P(S|X)$ is expressed by the similarity between c_i and c_j. There the object X represents the community and S represents the vesting relationship between communities.

The inclusion ratio thresholds α and β are the boundary of decision. Usually the loss function values are obtained from prior knowledge or the experience of experts and the values of α and β can be calculated by those loss function values. In this paper, we use the EQ function reflecting the strength of community structure as the objective function. If the value of EQ function increases after the merge of two communities, it indicates that the merger enhances the strength of community in social network. When two communities are very similar, then the connection between two communities is very close. If they are merged, their connection will become the internal connections, and the value of EQ function also will increase with it. We use the increments of EQ function to reflect the inclusion ratio thresholds α and β. That is, α is the maximum value of the increments of EQ function, and β is the minimum value of the increments of EQ function. Thus we can get values of α and β automatically.

4.4 Algorithm Description

The three-way decision based overlapping community detection algorithm (OCDBTWD) uses the positive domain, negative domain and boundary domain to depict the vesting relationship between communities and uses increments of extended modularity to reflect the inclusion ratio thresholds. OCDBTWD initializes every vertex to a community firstly, and then uses the three-way decision to decide the vesting relationship between communities to guide their merger until the value of extended modularity doesn't increase. The detail of OCDBTWD is described as follows:

Algorithm 1. OCDBTWD.
Input: $G = (V, E)$
Output: $C = \{c_1, c_2, \ldots, c_k\}$
$C \leftarrow \{c_1, c_2, \ldots, c_{|V|}\}$ //initialization, $|v|$ is number of vertices
while(true) do
$M \leftarrow \emptyset$ // M is set of increments of EQ function
$EQ \leftarrow$calculate the exended modularity
$C' \leftarrow C$ // to find increments of EQ function

```
for all c_i, c_j in C' do
      if c_i, c_j are connected
           c_i ← c_i ∪ c_j, delete c_j from C'
           EQ' ←calculate the exended modularity
           if (EQ' > EQ)
                M ← M ∪ EQ'
           end if
      end if
end for
if (M == ∅) break
α ←maximum value in M − EQ
β ←minimum value in M − EQ
for all c_i, c_j in C do
      if c_i, c_j are connected
            if (SCC(c_i, c_j) ≥ α)          //completely belong
                c_i ← c_i ∪ c_j, delete c_j from C// merge the c_i, c_j
            else if(β < SCC(c_i, c_j) < α)//incompletely belong
                 V' ←OVDA(c_i, c_j)          // Algorithm 2
                 c_i ← c_i ∪ V', c_j ← c_j ∪ V'
            end if                  //completely not belong
      end if
end for
end while
output C
```

When the similarity between communities c_i and c_j satisfies that $\beta < SCC(c_i, c_j) < \alpha$, the vesting relationship between c_i and $c_j$is incompletely belong relation, c_i and c_j are overlapping communities. So we propose the overlapping vertex detection algorithm (OVDA) to detect overlapping vertices. The idea of OVDA is that: assume that communities c_i and c_j are overlapping, and vertex v_n is belong to c_i. If the $MVC(v_n, c_j) > MVC(v_n, c_i)$, then v_n is an overlapping vertex. The details of OCDBTWD are described as follows:

```
Algorithm 2. OVDA.
Input: communities c_i, c_j
Output: overlapping vertices set V'
V' ← ∅          // initialization
for all v_n in c_i do
      if (MVC(v_n, c_j) > MVC(v_n, c_i))
           V' ← V' ∪ v_n
      end if
end for
for all v_m in c_j do
      if (MVC(v_m, c_i) > MVC(v_m, c_j))
```

```
            V' ← V' ∪ v_m
        end if
    end for
    output V'
```

5 Experiments

We test OCDBTWD on computer-generated networks and real world networks respectively. And we compare it with CPM. Our experiment environment is that: AMD Athlon X2 QL-64 CPU with 2.1GHz, 2G bytes memory, 250G bytes hard disk, Windows7 OS, programming language is Java6.0.

5.1 Computer-Generated Networks

We use the LFR-benchmark model [28] to generate networks and use the NMI (Normalized Mutual Information) [14] to verify accuracy of the algorithm. The definition of NMI is:

$$NMI = \frac{I(C_a, C_b)}{\sqrt{H(C_a)H(C_b)}}$$

Where $I(C_a, C_b) = \sum_{i=1}^{|C_a|}\sum_{j=1}^{|C_b|}\frac{n_{i,j}}{n} log\ (\frac{n_{a,b}}{n}/(\frac{n_i^a}{n}\frac{n_j^b}{n}))$, $H(C_a) = \sum_{i=1}^{|C_a|}\frac{n_i^a}{n} log\ (\frac{n_i^a}{n})$, $H(C_b) = \sum_{j=1}^{|C_b|}\frac{n_j^b}{n} log\ (\frac{n_j^b}{n})$.$C_a$ and C_b are the community sets of networks a, b respectively. n_i^a, n_j^b are the number of vertices in the i^{th}, j^{th} community of a, b respectively. n is the number of vertices in network, $n_{i,j}$ is the number of vertices both in the i^{th} community of a and the j^{th} community of b. And the value of NMI is close to 1, which indicates that the two communities match well.

This paper uses the LFR-benchmark model to generate 4 networks (G1 to G4), the parameters setting are shown in Table 1. The results are shown on Fig 1.

Table 1. LFR-benchmark model parameters setting

Description	G1	G2	G3	G4
number of vertices(N)	1000	1000	1000	1000
degree exponent(τ_1)	2	2	2	2
community exponent(τ_2)	1	1	1	1
max degree(k_{max})	40	40	40	40
average degress(k)	20	20	20	20
mixing parameter(μ)	0.1	0.3	0.1-0.3	0.1
num of overlap vertices(O_n)	50-300	50-300	100	50-300
vertex per community(O_m)	2	2	2	2
max comm size(C_{max})	100	100	100	50
min comm size(C_{min})	30	30	30	30

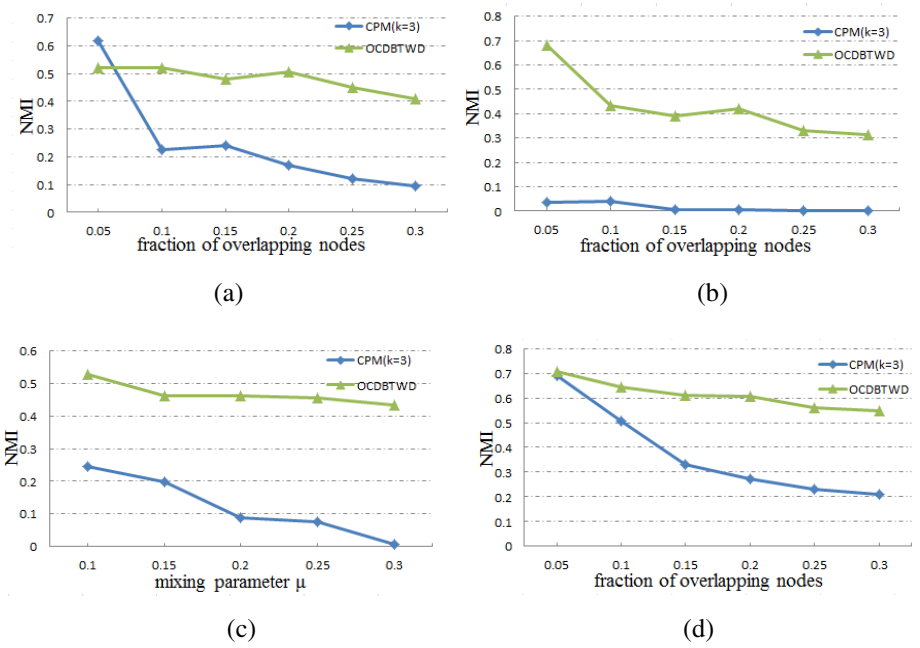


Fig. 1. Comparison OCDBTWD with CPM: (a) On G1 with small mixing parameter. (b) On G2 with big mixing parameter. (c) On G3. (d) On G4 with small maximum community size.

In Fig.1(a), the mixing parameter μ=0.1 (The bigger the value of μ is, the less obvious the community structure is). When the overlapping nodes are little (the fraction of overlapping nodes is 0.05), CPM is better than OCDBTWD, but when the overlapping nodes are more (the fraction of overlapping nodes is from 0.1 to 0.3), OCDBTWD is better than CPM. In Fig.1(b) the mixing parameter μ=0.3, the OCDBTWD is much better than CPM. We can find that the OCDBTWD and CPM will become poor with the increment of mixing parameter μ from Fig.1(a) and Fig.1(b), but the OCDBTWD descends slowly, that means OCDBTWD is more stable than CPM. This can also be found out from Fig.1(c). In Fig.1(d) the maximum community size is 50, when the fraction of overlapping nodes is 0.05, CPM is close to OCDBTWD, but when the fraction of overlapping nodes is from 0.1 to 0.3, OCDBTWD is much better than CPM. Combined with Fig.1(a) and Fig.1(d), we can find that OCDBTWD is better than CPM, no matter the community size is big or small (In Fig.1(a) the maximum community size is 100, while in Fig.1(b) the maximum community size is 50). Therefore, the results of computer-generated networks shows that OCDBTWD is feasible and stable.

5.2 Real-World Networks

In real-world networks, the communities are formed by some certain relationship. Their topology structures are different from those generated by computers. Here, we use Zachary's karate club network [29], American college football league network

[3], Dolphins' network [30] and Renren friends' relationship network to test OCDBTWD.

Zachary's karate club network has 34 vertices denoted the members in club and 78 edges denoted relationship between members. There are two communities centered on the administrator and the teacher. American college football league network has 115 vertices denoted the football teams and 616 edges denoted games between two teams. Usually, 8 to 12 teams to form a federation in network and each federation is a community. Dolphins' network has 62 vertices denoted the dolphins and 160 edges denoted the frequent association between dolphins. The network divides into to two subgroups because of the leaving of an important dolphin. Renren friends' relationship network is form by the friends of the author of this paper in Renren website. There are 109 vertices denoted the friends and 868 edges denoted the relationship between friends. The friends are the classmates in the stages of middle school, high school, undergraduate college and postgraduate. The classmates in each stage form a community. And we use those real world networks to test OCDBTWD and the NMI results are shown in Table 2.

Table 2. Real world netwroks result

Real world networks	OCDBTWD	CPM(k=3)
karate network	0.37	0.17
US college football team network	0.61	0.24
dolphins network	0.37	0.33
Renren friends' relationship network	0.67	0.88

From Table 2, we can know that the OCDBTWD is better than CPM in karate, US college football team and dolphins networks. In Renren friends' network, CPM is better than OCDBTWD, and the reason is that the every stage classmates' subgraph is mostly a complete subgraph, so this is very beneficial for CPM. The result of real world networks also show that OCDBTWD is feasible and it effectively detects the overlapping community structures

6 Conclusion

In this paper, we divide the vesting relatioship between communities into three types: completely belong relation, completely not belong relation and incompletely belong relation. We use the positive domain, negative domain, and boundary domain in three-way decision to reflect these vesting relationships. The OCDBTWD defines the similarity between vertices, the similarity between communities, and membership ratio between vertex and community, and it uses the increment values of extended modularity to reflect the inclusion ratio thresholds, and then it uses three-way decision to decide the vesting relationship between communities. When the vesting relationship between two communities is completely belong relation, then merge them; when the vesting relationship is completely not belong relation, then do nothing; when the

vesting relationship is incompletely belong relation, then using the OVDA to detect the overlapping vertices. The OCDBTWD has been tested on computer-generated networks and real world networks, and compared with other overlapping community detection algorithms. The experiments show that the OCDBTWD is feasible and effectively detect the overlapping community structures.

However, there are still some problems to be solved, such as: in real world, the large-scale social networks like Facebook, Sina Weibo and Twitter have lots of vertices and edges, and how to use local information to improve the efficiency of OCDBTWD for large-scale social networks. This will be discussed in our future work.

Acknowledgments. This paper was supported by the "973" Project of China under Grant 2011 CB505300, by the National Natural Science Foundation of China under Grant 61021062 and 61105069, by the Technology Support Program of Jiangsu Province under Grant BE2011171 and BE2012161, also by the Innovation Fund of Nanjing University under Grant 2011CL07.

References

1. Watts, D.J., Strogatz, S.H.: Collective dynamic of 'small-world' network. Nature 393(6638), 440–442 (1998)
2. Albert, R., Jeong, H., Barabsi, A.L.: The internet's achille'heel: error and attack tolerance of complex networks. Nature 406(2115), 378–382 (2000)
3. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. PNAS 99(12), 7821–7826 (2002)
4. Chen, J.C., Yuan, B.: Detecting functional modules in the yeast protein–protein interaction network. Bioinformatics 22(18), 2276–2282 (2006)
5. Rives, A.W., Galitski, T.: Modular organization of cellular networks. In: Proceeding of the National Academy of Sciences, pp. 1128–1133 (2003)
6. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. In: Proceeding of the National Academy of Sciences, pp. 12123–12128 (2003)
7. Tang, L., Barbier, G., Liu, H., Zhang, J.: A social network analysis approach to detecting suspicious online financial activities. In: Chai, S.-K., Salerno, J.J., Mabry, P.L. (eds.) SBP 2010. LNCS, vol. 6007, pp. 390–397. Springer, Heidelberg (2010)
8. Gao, J., Liang, F., Fan, W., Wang, C., Sun, Y., Han, J.: On community outliers and their efficient detection in information networks. In: Process of KDD 2010, pp. 813–822 (2010)
9. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E 69(2), 1–15 (2004)
10. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. E 69(6), 066133 (2004)
11. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043), 814–818 (2005)
12. Lee, G., Reid, F., McDaid, A.: Detecting highly overlapping community structure by greedy clique expansion. In: The 4th SNA-KDD Workshop 2010 (SNA-KDD 2010), pp. 33–42 (2010)

13. Shen, H., Cheng, X., Cai, K., Hu, B.M.: Detect overlapping and hierarchical community structure in network. Physica A: Statistical Mechanics and its Applications 388(8), 1706–1712 (2009)
14. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics 11(3), 015–033 (2009)
15. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
16. Yao, Y.Y., Wong, S.K.M., Lingras, P.: A decision-theoretic rough set model. In: Proceedings of the 5th International Symposium on Methodologies for Intelligent Systems, pp. 17–24 (1990)
17. Yao, Y.Y., Zhao, Y.: Attribute reduction in decision-theoretic rough set models. Information Sciences 178(17), 3356–3373 (2008)
18. Yao, Y.Y.: Three-way decision: An interpretation of rules in rough set theory. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 642–649. Springer, Heidelberg (2009)
19. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180(3), 341–353 (2010)
20. Yao, Y.Y.: The superiority of three-way decisions in probabilistic rough set models. Information Sciences 181(6), 1086–1096 (2011)
21. Zhou, X.Z., Li, H.X.: A multi-view decision model based on decision-theoretic rough set. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 650–657. Springer, Heidelberg (2009)
22. Li, H.X., Zhou, X.Z.: Risk decision making based on decision-theoretic rough set: A multi-view decision model. International Journal of Computational Intelligence Systems 4(1), 1–11 (2011)
23. Liu, D., Yao, Y.Y., Li, T.R.: Three-way investment decisions with decision-theoretic rough sets. International Journal of Computational Intelligence System 4(1), 66–74 (2011)
24. Li, W., Miao, D.Q., Wang, W.L.: Hierarchical rough decision theoretic framework for text classification. In: 2010 9th IEEE International Conference on Cognitive Informatics, pp. 484–489 (2010)
25. Yao, J.T., Herbert, J.P.: Web-based support systems with rough set analysis. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 360–370. Springer, Heidelberg (2007)
26. Yu, H., Chu, S.S., Yang, D.C.: Autonomous knowledge-oriented clustering using decision-theoretic rough set theory. Fundamenta Informaticae 15(2), 141–156 (2012)
27. Yu, H., Liu, Z.G., Wang, G.Y.: Automatically determining the number of clusters using decision-theoretic rough set. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 504–513. Springer, Heidelberg (2011)
28. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithm. Physical Review E 78(4), 046110 (2008)
29. Zachary, W.W.: An information flow model for conict and fission in small groups. Journal of Anthropological Research 33(4), 452–473 (1977)
30. Lusseau, D.: The emergent properties of a dolphin social network. Proc. R. Soc. Lond. B (Suppl.), 186–188 (2003)

Three-Way Decisions in Dynamic Decision-Theoretic Rough Sets

Dun Liu[1,2], Tianrui Li[3], and Decui Liang[2]

[1] School of Economics and Management, Tsinghua University
Beijing 100084, P.R. China
liudun@sem.tsinghua.edu.cn
[2] School of Economics and Management, Southwest Jiaotong University
Chengdu 610031, P.R. China
decuiliang@126.com
[3] School of Information Science and Technology, Southwest Jiaotong University
Chengdu 610031, P.R. China
trli@swjtu.edu.cn

Abstract. In the previous decision-theoretic rough sets (DTRS), its loss function values are constant. This paper extends the constant values of loss functions to a more realistic dynamic environment. Considering the dynamic change of loss functions in DTRS with the time, an extension of DTRS, dynamic decision-theoretic rough sets (DDTRS) is proposed in this paper. An empirical study of climate policy making validates the reasonability and effectiveness of the proposed model.

Keywords: Decision-theoretic rough sets, loss functions, decision-making, dynamic.

1 Introduction

In some decision problems, one may make a decision with two choices of acceptance or rejection, especially one has sufficient confidence to do it. However, in many real scenarios, one can not make a decision immediately because of insufficient information, uncontrolled risks or lack of recognition. In this case, a third choice, deferment, is used to deal with these things which one do not have full understanding. The three types of choices of acceptance, rejection and deferment, denoted as three-way decisions [1, 2], have been used in many studies such as interval sets, three-valued logic, rough sets, fuzzy sets, shadowed sets, and others [3–6]. They have been applied in many disciplines, including medical clinic [7], email spam filtering [8], investment management [9], web support systems [10], products inspecting process [11], policy making [12], etc.

With respect to the three-way decisions using rough sets, the three regions generated by lower and upper approximations lead to three-way decision rules. Rules from the positive region are used for making a decision of acceptance, rules from the negative region for making a decision of rejection, and rules from the boundary region for making a decision of non-commitment or deferment [1, 2]. In

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 291–301, 2013.

classical rough sets, decisions of acceptance and rejection must be made without any error; in probabilistic rough sets, acceptance and rejection are associated with some tolerance levels of errors by using two parameters α and β [13–15]. Obviously, the three-way decisions of classical rough sets are of qualitative nature and called qualitative three-way decisions. Different from qualitative three-way decisions, quantitative three-way decisions require a semantic interpretation and computation of parameters α and β. Observed by this issue, Yao introduced loss functions to PRS and proposed decision-theoretic rough sets model (DTRS) with Bayesian decision procedures [16]. In DTRS, the two parameters α and β can be automatically computed by minimizing the expected overall risk function, and it gives a brief semantic explanation with minimum decision risks.

However, the loss functions in DTRS are precise and constant. The decision makers may hardly estimate the loss function values, especially when the decision procedure is complex and dynamic. As for the limitations of precise values, Liu et al. suggested to use some uncertain information (i.e. stochastic, vague or rough information) instead of precise ones in real decision procedure, and they further proposed stochastic decision-theoretic rough sets (SDTRS) [6], interval-valued decision-theoretic rough sets (IVDTRS) [4] and fuzzy decision-theoretic rough sets (FDTRS) [5], respectively. Liang et al. generalized a concept of the precise value of loss function to triangular fuzzy number, and proposed triangular fuzzy decision-theoretic rough sets (TFDTRS) [3]. Liu et al. introduced the fuzzy interval number to DTRS, and proposed a novel three-way decision model of fuzzy interval decision-theoretic rough sets (FIDTRS) [17]. Yao and Deng discussed the sequential three-way decisions with probabilistic rough sets, in which the cost of obtaining required evidence or information is considered [18]. Yao further presented a granular computing perspective on sequential three-way decisions in [19]. Li et al. investigated the cost-sensitive three-way decision with the sequential strategy [20]. In this paper, we mainly focus on investigating the situation that loss functions are dynamically varying under the dynamic decision environment, and a novel extended model of DTRS, dynamic decision-theoretic rough sets (DDTRS), is proposed.

The remainder of this paper is organized as follows: Section 2 provides the basic concepts of PRS and DTRS. DTRS model with dynamic loss function is proposed and its properties are analyzed in Section 3. Then, a case study of climate policy making problem is given to illustrate our approach in Section 4. Section 5 concludes the paper and outlines the future work.

2 Preliminaries

Basic concepts, notations and results of the PRS and DTRS are briefly reviewed in this section [1, 2, 13–16, 21–27].

Definition 1. *Let $S = (U, A, V, f)$ be an information system. $\forall x \in U, X \subseteq U$, let: $\mu_X(x) = Pr(X|[x]) = \frac{|[x] \bigcap X|}{|[x]|}$ be a rough membership function, where, $|\cdot|$ stands for the cardinality of a set, $Pr(X|[x])$ is is the conditional probability of*

an object in X given that the object is in $[x]$, estimated by using the cardinalities of sets.

A main result in PRS is parameterized probabilistic approximations, which is similar to the notion of α-cuts of fuzzy sets. This can be done by pair of parameters α and β with $\alpha > \beta$.

Definition 2. *Let $S = (U, A, V, f)$ be an information system. $\forall X \subseteq U$ and $0 \leq \beta < \alpha \leq 1$, the (α, β)-lower approximation, (α, β)-upper approximation are defined as follows:*

$$\begin{aligned} \underline{apr}_{(\alpha,\beta)}(X) &= \{x \in U | Pr(X|[x]) \geq \alpha\}; \\ \overline{apr}_{(\alpha,\beta)}(X) &= \{x \in U | Pr(X|[x]) > \beta\}. \end{aligned} \tag{1}$$

From the (α, β)-probabilistic lower and upper approximations, we can obtain the (α, β)-probabilistic positive, boundary and negative regions:

$$\begin{aligned} \mathrm{POS}_{(\alpha,\beta)}(X) &= \{x \in U \mid Pr(X|[x]) \geq \alpha\}, \\ \mathrm{BND}_{(\alpha,\beta)}(X) &= \{x \in U \mid \beta < Pr(X|[x]) < \alpha\}, \\ \mathrm{NEG}_{(\alpha,\beta)}(X) &= \{x \in U \mid Pr(X|[x]) \leq \beta\}. \end{aligned} \tag{2}$$

To acquire the values of the two parameters α and β, Yao et al. introduced Bayesian decision procedure into RST and proposed DTRS [16]. The DTRS model is composed of 2 states and 3 actions. The set of states is given by $\Omega = \{X, \neg X\}$ indicating that an object is in X and not in X, respectively. The set of actions is given by $\mathcal{A} = \{a_P, a_B, a_N\}$, where a_P, a_B, and a_N represent the three actions in classifying an object x, namely, deciding $x \in \mathrm{POS}(X)$, deciding x should be further investigated $x \in \mathrm{BND}(X)$, and deciding $x \in \mathrm{NEG}(X)$, respectively. The loss function λ regarding the risk or cost of actions in different states is given by the 3×2 matrix:

	X (P)	$\neg X$ (N)
a_P	λ_{PP}	λ_{PN}
a_B	λ_{BP}	λ_{BN}
a_N	λ_{NP}	λ_{NN}

In the matrix, λ_{PP}, λ_{BP} and λ_{NP} denote the losses incurred for taking actions of a_P, a_B and a_N, respectively, when an object belongs to X. Similarly, λ_{PN}, λ_{BN} and λ_{NN} denote the losses incurred for taking the same actions when the object belongs to $\neg X$. $Pr(X|[x])$ is the conditional probability of an object x belonging to X given that the object is described by its equivalence class $[x]$. For an object x, the expected loss $R(a_i|[x])$ associated with taking the individual actions can be expressed as:

$$\begin{aligned} R(a_P|[x]) &= \lambda_{PP} Pr(X|[x]) + \lambda_{PN} Pr(\neg X|[x]), \\ R(a_B|[x]) &= \lambda_{BP} Pr(X|[x]) + \lambda_{BN} Pr(\neg X|[x]), \\ R(a_N|[x]) &= \lambda_{NP} Pr(X|[x]) + \lambda_{NN} Pr(\neg X|[x]). \end{aligned}$$

The Bayesian decision procedure suggests the following minimum-cost decision rules:

(P) If $R(a_P|[x]) \leq R(a_B|[x])$ and $R(a_P|[x]) \leq R(a_N|[x])$, decide $x \in \text{POS}(X)$;
(B) If $R(a_B|[x]) \leq R(a_P|[x])$ and $R(a_B|[x]) \leq R(a_N|[x])$, decide $x \in \text{BND}(X)$;
(N) If $R(a_N|[x]) \leq R(a_P|[x])$ and $R(a_N|[x]) \leq R(a_B|[x])$, decide $x \in \text{NEG}(X)$.

Since $Pr(X|[x]) + Pr(\neg X|[x]) = 1$, we simplify the rules based only on the probability $Pr(X|[x])$ and the loss function. By considering a reasonable kind of loss functions with $\lambda_{PP} \leq \lambda_{BP} < \lambda_{NP}$ and $\lambda_{NN} \leq \lambda_{BN} < \lambda_{PN}$, the decision rules (P)-(N) can be expressed concisely as:

(P) If $Pr(X|[x]) \geq \alpha$ and $Pr(X|[x]) \geq \gamma$, decide $x \in \text{POS}(X)$;
(B) If $Pr(X|[x]) \leq \alpha$ and $Pr(X|[x]) \geq \beta$, decide $x \in \text{BND}(X)$;
(N) If $Pr(X|[x]) \leq \beta$ and $Pr(X|[x]) \leq \gamma$, decide $x \in \text{NEG}(X)$.

The thresholds values α, β, γ are given by:

$$\alpha = \frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})};$$
$$\beta = \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})};$$
$$\gamma = \frac{(\lambda_{PN} - \lambda_{NN})}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}. \tag{3}$$

In addition, as a well-defined boundary region, the conditions of rule (B) suggest that $\alpha > \beta$, that is, $\frac{(\lambda_{BP} - \lambda_{PP})}{(\lambda_{PN} - \lambda_{BN})} < \frac{(\lambda_{NP} - \lambda_{BP})}{(\lambda_{BN} - \lambda_{NN})}$. It implies $0 \leq \beta < \gamma < \alpha \leq 1$. To sum up, DTRS not only introduces the probabilistic rough set approximation of equation (2), but also provides semantical interpretation of the thresholds.

3 Dynamic Decision-Theoretic Rough Set Model

Our following discussions are motivated by an example of Savage's Omelet [28, 29]. In this example, Savage described a decision situation as follows: your wife has just broken five good eggs into a bowl when you come in and volunteer to finish making the omelet. The sixth egg, which for some reason must either be used for the omelet or wasted altogether, lies unbroken beside the bowl. You need decide what to do with this unbroken egg. Perhaps it is not too great an oversimplification to say that you must decide among three acts only, namely, to break it into the bowl containing the other five, to break it into a saucer for inspection, or to throw it away without inspection. Table 1 outlines the acts and states of this example.

In Table 1, the states of this problem is simply specified whether the sixth egg is good, and the three possible acts for you are discredited as "break the

Table 1. Savage's example illustrating acts and states [29]

Act	State	
	Good	Rotten
Break into bowl	six-egg omelet	no omelet, and five good eggs destroyed
Break into saucer	six-egg omelet, and a saucer to wash	five-egg omelet, and a saucer to wash
Throw away	five-egg omelet, and one good egg destroyed	five-egg omelet

egg into the bowl", "break the egg into the saucer" and "throw the egg away". Suppose that each good egg is no cost, a saucer to wash costs you 1 point and that each good egg destroyed costs you 2 points because of the reproaches of your wife [28]. The losses of Table 1 can be rewritten as:

	Good	Rotten
Break into bowl	0	10
Break into saucer	1	1
Throw away	2	0

Note that, the losses of the above matrix are constantly changing during the sequential decision process, *e.g.*, one should make continuous decisions from the first egg to the sixth egg. Observed by this issue, we consider the case where loss functions in DTRS are changing with time, and develop a dynamic decision-theoretic rough set model as follows.

In DDTRS, we also considers 2 states $\Omega = \{X, \neg X\}$ and 3 actions $\mathcal{A} = \{a_P, a_B, a_N\}$. Suppose λ^t_{PP}, λ^t_{BP} and λ^t_{NP} denote the losses incurred for taking actions of a_P, a_B and a_N at time t, when an object belongs to X. Similarly, λ^t_{PN}, λ^t_{BN} and λ^t_{NN} denote the losses incurred for taking the same actions at time t, when the object belongs to $\neg X$. We can rewrite the 3×2 matrix at time t as:

	$X\ (P)$	$\neg X\ (N)$
a_P	λ^t_{PP}	λ^t_{PN}
a_B	λ^t_{BP}	λ^t_{BN}
a_N	λ^t_{NP}	λ^t_{NN}

In the matrix, for each time t $(t = 1, 2, \cdots, n)$, we consider the following two factors. First, suppose t is a dependent variable, and $\lambda^t_{\bullet\bullet}$ $(\bullet = P, B, N)$ is directly depended on the variation of time t, *e.g.*, $\lambda^t_{\bullet\bullet} = 2t+1$. Second, suppose $\lambda^t_{\bullet\bullet}$ is affected by its former status, that is, $\lambda^t_{\bullet\bullet}$ is decided by $\lambda^{t-1}_{\bullet\bullet}, \lambda^{t-2}_{\bullet\bullet}, \cdots \lambda^{t-m}_{\bullet\bullet}$, $m \leq t$, *e.g.*, $\lambda^t_{\bullet\bullet} = 3\lambda^{t-1}_{\bullet\bullet} + 2\lambda^{t-2}_{\bullet\bullet}$. Obviously, by considering both combinations and semantics of $\lambda^t_{\bullet\bullet}$, there exist four scenarios: (1). consider two factors simultaneously; (2). only consider the first factor; (3). only consider the second factor; (4). neither consider the two factors, respectively.

Scenario 1: $\lambda^t_{\bullet\bullet}$ $(\bullet = P, B, N)$ is depended on two types of factors.

For simplicity, we only consider $\lambda^t_{\bullet\bullet}$ is depended on $\lambda^{t-1}_{\bullet\bullet}$ and the variation of time t. One form of the loss function $\lambda^t_{\bullet\bullet}$ can be expressed as: $\lambda^t_{\bullet\bullet} = f(\lambda^{t-1}_{\bullet\bullet}, t) = a_{\bullet\bullet} \cdot \lambda^{t-1}_{\bullet\bullet} + f(t), a_{\bullet\bullet} \neq 0$. $a_{\bullet\bullet}$ is the coefficient of $\lambda^t_{\bullet\bullet}$, and $f(t)$ is corresponding with the time t.

In this scenario, the relations between $\lambda^t_{\bullet\bullet}$ and $\lambda^1_{\bullet\bullet}$ can be calculated as:

$$\begin{aligned}
\lambda^t_{\bullet\bullet} &= a_{\bullet\bullet} \cdot \lambda^{t-1}_{\bullet\bullet} + f(t) \\
&= a_{\bullet\bullet} \cdot (a_{\bullet\bullet} \cdot \lambda^{t-2}_{\bullet\bullet} + f(t-1)) + f(t) \\
&= a_{\bullet\bullet} \cdot (a_{\bullet\bullet} \cdot (a_{\bullet\bullet} \cdot (\lambda^{t-3}_{\bullet\bullet} + f(t-2)) + f(t-1))) + f(t) \\
&\quad \cdots \\
&= a^{t-1}_{\bullet\bullet} \lambda^1_{\bullet\bullet} + \sum_{i=2}^{t} a^{t-i}_{\bullet\bullet} f(i)
\end{aligned}$$

Therefore, $\lambda^t_{\bullet\bullet}$ can be represented as:

$$\lambda^t_{\bullet\bullet} = \begin{cases} \lambda^1_{\bullet\bullet} & t = 1 \\ a^{t-1}\lambda^1_{\bullet\bullet} + \sum_{i=2}^{t} a^{t-i}_{\bullet\bullet} f(i) & t > 1 \end{cases}$$

On the basis of conditions in DTRS, it also requires the loss functions in each time t ($t = 1, 2, \cdots, n$) satisfy $\lambda^t_{PP} \leq \lambda^t_{BP} < \lambda^t_{NP}$ and $\lambda^t_{NN} \leq \lambda^t_{BN} < \lambda^t_{PN}$. Under the conditions, we can easily calculate the three thresholds values α, β, γ at time t as:

$$\begin{aligned}
\alpha^t &= \frac{(\lambda^t_{PN} - \lambda^t_{BN})}{(\lambda^t_{PN} - \lambda^t_{BN}) + (\lambda^t_{BP} - \lambda^t_{PP})}; \\
\beta^t &= \frac{(\lambda^t_{BN} - \lambda^t_{NN})}{(\lambda^t_{BN} - \lambda^t_{NN}) + (\lambda^t_{NP} - \lambda^t_{BP})}; \\
\gamma^t &= \frac{(\lambda^t_{PN} - \lambda^t_{NN})}{(\lambda^t_{PN} - \lambda^t_{NN}) + (\lambda^t_{NP} - \lambda^t_{PP})}.
\end{aligned} \tag{4}$$

where, $\lambda^t_{PP} = a^{t-1}_{PP}\lambda^1_{PP} + \sum_{i=2}^{t} a^{t-i}_{PP} f(i)$, $\lambda^t_{BP} = a^{t-1}_{BP}\lambda^1_{BP} + \sum_{i=2}^{t} a^{t-i}_{BP} f(i)$, $\lambda^t_{NP} = a^{t-1}_{NP}\lambda^1_{NP} + \sum_{i=2}^{t} a^{t-i}_{NP} f(i)$; $\lambda^t_{NN} = a^{t-1}_{NN}\lambda^1_{NN} + \sum_{i=2}^{t} a^{t-i}_{NN} f(i)$, $\lambda^t_{BN} = a^{t-1}_{BN}\lambda^1_{BN} + \sum_{i=2}^{t} a^{t-i}_{BN} f(i)$, $\lambda^t_{PN} = a^{t-1}_{PN}\lambda^1_{PN} + \sum_{i=2}^{t} a^{t-i}_{PN} f(i)$.

Scenario 2: $\lambda^t_{\bullet\bullet}$ ($\bullet = P, B, N$) is depended on the first factor.

In this scenario, one form of the loss function can be expressed as: $\lambda^t_{\bullet\bullet} = f(t) = b_{\bullet\bullet} \cdot t + c_{\bullet\bullet}$, $b_{\bullet\bullet} \neq 0$. $b_{\bullet\bullet}$ is the coefficient of $f(t)$, and $c_{\bullet\bullet}$ is a constant. Therefore, $\lambda^t_{\bullet\bullet}$ can be represented as: $b_{\bullet\bullet} \cdot t + c_{\bullet\bullet}$. Specially, when $t = 1$, $\lambda^1_{\bullet\bullet} = b_{\bullet\bullet} + c_{\bullet\bullet}$.

Similarly, under the basic conditions $\lambda^t_{PP} \leq \lambda^t_{BP} < \lambda^t_{NP}$ and $\lambda^t_{NN} \leq \lambda^t_{BN} < \lambda^t_{PN}$ ($t = 1, 2, \cdots, n$), we can get the three thresholds α^t, β^t and γ^t as: $\alpha^t = \frac{(\lambda^t_{PN} - \lambda^t_{BN})}{(\lambda^t_{PN} - \lambda^t_{BN}) + (\lambda^t_{BP} - \lambda^t_{PP})}$, $\beta^t = \frac{(\lambda^t_{BN} - \lambda^t_{NN})}{(\lambda^t_{BN} - \lambda^t_{NN}) + (\lambda^t_{NP} - \lambda^t_{BP})}$ and $\gamma^t = \frac{(\lambda^t_{PN} - \lambda^t_{NN})}{(\lambda^t_{PN} - \lambda^t_{NN}) + (\lambda^t_{NP} - \lambda^t_{PP})}$. where, $\lambda^t_{PP} = b_{PP} \cdot t + c_{PP}$, $\lambda^t_{BP} = b_{BP} \cdot t + c_{BP}$,

$\lambda_{NP}^t = b_{NP} \cdot t + c_{NP}$; $\lambda_{NN}^t = b_{NN} \cdot t + c_{NN}$, $\lambda_{BN}^t = b_{BN} \cdot t + c_{BN}$, $\lambda_{PN}^t = b_{PN} \cdot t + c_{PN}$.

Scenario 3: $\lambda_{\bullet\bullet}^t$ ($\bullet = P, B, N$) is depended on the second factor.

In this scenario, one form of the loss function can be expressed as: $\lambda_{\bullet\bullet}^t = f(\lambda_{\bullet\bullet}^{t-1}) = a \cdot \lambda_{\bullet\bullet}^{t-1} + c_{\bullet\bullet}$, $a_{\bullet\bullet} \neq 0$. $a_{\bullet\bullet}$ is the coefficient of $\lambda_{\bullet\bullet}^{t-1}$, and $c_{\bullet\bullet}$ is a constant. The relations between $\lambda_{\bullet\bullet}^t$ and $\lambda_{\bullet\bullet}^1$ can be calculated as:

$$\begin{aligned}\lambda_{\bullet\bullet}^t &= a \cdot \lambda_{\bullet\bullet}^{t-1} + c_{\bullet\bullet} \\ &= a \cdot (a \cdot \lambda_{\bullet\bullet}^{t-2} + c_{\bullet\bullet}) + c_{\bullet\bullet} \\ &= a \cdot (a \cdot (a \cdot \lambda_{\bullet\bullet}^{t-3} + c_{\bullet\bullet}) + c_{\bullet\bullet}) + c_{\bullet\bullet} \\ &\quad \cdots \\ &= a_{\bullet\bullet}^{t-1} \cdot \lambda_{\bullet\bullet}^1 + \frac{c_{\bullet\bullet} \cdot (1 - a_{\bullet\bullet}^{t-1})}{1 - a_{\bullet\bullet}}.\end{aligned}$$

Therefore, $\lambda_{\bullet\bullet}^t$ can be represented as:

$$\lambda_{\bullet\bullet}^t = \begin{cases} \lambda_{\bullet\bullet}^1 & t = 1 \\ a_{\bullet\bullet}^{t-1} \cdot \lambda_{\bullet\bullet}^1 + \frac{c_{\bullet\bullet} \cdot (1 - a_{\bullet\bullet}^{t-1})}{1 - a_{\bullet\bullet}} & t > 1 \end{cases}$$

Similarly, under the basic conditions $\lambda_{PP}^t \leq \lambda_{BP}^t < \lambda_{NP}^t$ and $\lambda_{NN}^t \leq \lambda_{BN}^t < \lambda_{PN}^t$ ($t = 1, 2, \cdots, n$), we can get the three thresholds α^t, β^t and γ^t as: $\alpha^t = \frac{(\lambda_{PN}^t - \lambda_{BN}^t)}{(\lambda_{PN}^t - \lambda_{BN}^t) + (\lambda_{BP}^t - \lambda_{PP}^t)}$, $\beta^t = \frac{(\lambda_{BN}^t - \lambda_{NN}^t)}{(\lambda_{BN}^t - \lambda_{NN}^t) + (\lambda_{NP}^t - \lambda_{BP}^t)}$ and $\gamma^t = \frac{(\lambda_{PN}^t - \lambda_{NN}^t)}{(\lambda_{PN}^t - \lambda_{NN}^t) + (\lambda_{NP}^t - \lambda_{PP}^t)}$. where, $\lambda_{PP}^t = a_{PP}^{t-1} \cdot \lambda_{PP}^1 + \frac{c_{PP} \cdot (1 - a_{PP}^{t-1})}{1 - a_{PP}}$, $\lambda_{BP}^t = a_{BP}^{t-1} \cdot \lambda_{BP}^1 + \frac{c_{BP} \cdot (1 - a_{BP}^{t-1})}{1 - a_{BP}}$, $\lambda_{NP}^t = a_{NP}^{t-1} \cdot \lambda_{NP}^1 + \frac{c_{NP} \cdot (1 - a_{NP}^{t-1})}{1 - a_{NP}}$; $\lambda_{NN}^t = a_{NN}^{t-1} \cdot \lambda_{NN}^1 + \frac{c_{NN} \cdot (1 - a_{NN}^{t-1})}{1 - a_{NN}}$, $\lambda_{BN}^t = a_{BN}^{t-1} \cdot \lambda_{BN}^1 + \frac{c_{BN} \cdot (1 - a_{BN}^{t-1})}{1 - a_{BN}}$, $\lambda_{PN}^t = a_{PN}^{t-1} \cdot \lambda_{PN}^1 + \frac{c_{PN} \cdot (1 - a_{PN}^{t-1})}{1 - a_{PN}}$.

Scenario 4: $\lambda_{\bullet\bullet}^t$ ($\bullet = P, B, N$) is not depended on two factors

In this scenario, the value of $\lambda_{\bullet\bullet}^t$ is constant and don't change with the variation of t, and $\lambda_{\bullet\bullet}^t = \lambda_{\bullet\bullet}^{t-1} = \cdots = \lambda_{\bullet\bullet}^1 = c_{\bullet\bullet}$.

Under the basic conditions $\lambda_{PP}^t \leq \lambda_{BP}^t < \lambda_{NP}^t$ and $\lambda_{NN}^t \leq \lambda_{BN}^t < \lambda_{PN}^t$ for ($t = 1, 2, \cdots, n$), we can calculate the three thresholds α, β and γ at time t as: $\alpha^t = \frac{(c_{PN} - c_{BN})}{(c_{PN} - c_{BN}) + (c_{BP} - c_{PP})}$, $\beta^t = \frac{(c_{BN} - c_{NN})}{(c_{BN} - c_{NN}) + (c_{NP} - c_{BP})}$, $\gamma^t = \frac{(c_{PN} - c_{NN})}{(c_{PN} - c_{NN}) + (c_{NP} - c_{PP})}$.

Obviously, the three parameters have the same presentations with (3) in Section 2, and the DDTRS model converts to the classical DTRS model under the conditions in scenario 4.

4 An Illustration

In this section, we illustrate the extended model by an example of decision in climate policy. The debate over a policy response to global climate change has

been and continues to be deadlocked by two aspects: (1). The view that the impacts of climate change are too uncertain and the response to the policy should be delayed until we learn more. (2). We cannot wait to resolve the uncertainty because climate change is irreversible so we must take precautionary measure now [30]. By considering the change of global climate are depended on some uncertain characteristics: a long time horizon, large uncertainties in both the scientific basis and the potential economic costs of addressed it, uneven distribution of costs and damages, possible irreversible effects, and collective action required for an effective response [30]. The respond to climate change is not a decision that must be made now and set in stone for all time.

In our following discussions, we suppose the process of climate change last three periods, from time $t = 1$ to time $t = 3$. With insightful gain from DTRS, the climate policy making procedure may lead to three kinds of actions: executed, need further investigated and do not executed, respectively. As well, the states of a climate policy are described as {good policy, bad policy} according to a series of carefully explorations and appraisal with the characteristics of the global climate change issue. The two states are given by $\Omega = \{X, \neg X\}$ and the three actions of the decisions are given by $\mathcal{A} = \{a_P, a_B, a_N\}$. Suppose λ^t_{PP}, λ^t_{BP} and λ^t_{NP} denote the loss incurred for taking actions of executing, need further investigated and non-execute, respectively, when the climate policy is good at time t; whereas, λ^t_{PN}, λ^t_{BN} and λ^t_{NN} denote the loss incurred for taking actions of executing, need further investigated and non-execute, respectively, when the climate policy is bad at time t. For simplicity, we consider 4 types of climate policies $PO = \{po_1, po_2, po_3, po_4\}$, the loss functions for 4 types of climate policies are outlined in Table 2.

Table 2. The loss functions of 4 types of climate policies

PO	λ^t_{PP}	λ^t_{BP}	λ^t_{NP}	λ^t_{PN}	λ^t_{BN}	λ^t_{NN}
po_1	$\lambda^{t-1}_{PP}+t+1$	$2\lambda^{t-1}_{BP}+2t+2$	$4\lambda^{t-1}_{NP}+3t+3$	$5\lambda^{t-1}_{PN}+5t+2$	$3\lambda^{t-1}_{BN}+3t+1.5$	$2\lambda^{t-1}_{NN}+2t+0.5$
po_2	$t+1$	$2t+2$	$4t+3$	$5t+2$	$3t+1.5$	$2t+0.5$
po_3	$\lambda^{t-1}_{PP}+1$	$2\lambda^{t-1}_{BP}+2$	$4\lambda^{t-1}_{NP}+3$	$5\lambda^{t-1}_{PN}+2$	$3\lambda^{t-1}_{BN}+1.5$	$2\lambda^{t-1}_{NN}+0.5$
po_4	1	2	4	5	3	2

where, $\lambda^0_{PP} = \lambda^0_{BP} = \lambda^0_{NP} = \lambda^0_{PN} = \lambda^0_{BN} = \lambda^0_{NN} = 1$.

In Table 2, po_1 corresponds to the scenario 1, po_2 corresponds to the scenario 2, po_3 corresponds to the scenario 3, po_4 corresponds to the scenario 4. According to our analysis in Section 3, the loss values for $\{po_1, po_2, po_3, po_4\}$ and their corresponding α^t, β^t and γ^t $(t = 1, 2, 3)$ are calculated in Table 3.

In Table 3, the three parameters α^t, β^t and γ^t are changing with the increasing of t for po_1, po_2 and po_3. For simplicity, we set $Pr^t(X|po_i) = 0.76$ (t=1,2,3; i=1,2,3,4). With the decision criteria of (P), (B) and (N), we can obtain the decision regions for 4 types of climate policies in different periods by comparing the thresholds in Table 3 and $Pr(X|po_i)$, which are listed in Table 4.

Table 3. The loss values and three parameters for 4 types of climate policies

PO	λ^1_{PP}	λ^1_{BP}	λ^1_{NP}	λ^1_{PN}	λ^1_{BN}	λ^1_{NN}	α^1	β^1	γ^1
po_1	3	6	11	12	7.5	4.5	0.6000	0.3750	0.4839
po_2	2	4	7	7	4.5	2.5	0.5556	0.4000	0.4737
po_3	2	4	7	7	4.5	2.5	0.5556	0.4000	0.4737
po_4	1	2	4	5	3	2	0.6667	0.3333	0.5000
PO	λ^2_{PP}	λ^2_{BP}	λ^2_{NP}	λ^2_{PN}	λ^2_{BN}	λ^2_{NN}	α^2	β^2	γ^2
po_1	6	18	55	72	30	13.5	0.7778	0.3084	0.5442
po_2	3	6	11	12	7.5	4.5	0.6000	0.3750	0.4839
po_3	3	10	31	37	15	5.5	0.7586	0.3115	0.5294
po_4	1	2	4	5	3	2	0.6667	0.3333	0.5000
PO	λ^3_{PP}	λ^3_{BP}	λ^3_{NP}	λ^3_{PN}	λ^3_{BN}	λ^3_{NN}	α^3	β^3	γ^3
po_1	10	44	235	377	100.5	33.5	0.8905	0.2597	0.6042
po_2	4	8	15	17	10.5	6.5	0.6190	0.3636	0.4884
po_3	4	22	127	187	46.5	11.5	0.8864	0.2500	0.5879
po_4	1	2	4	5	3	2	0.6667	0.3333	0.5000

Table 4. The decision regions for the 4 types of climate policies in different periods

$Time$	(α, β)	$POS(X)$	$BND(X)$	$NEG(X)$
$t = 1$	(α^1, β^1)	$\{po_1, po_2, po_3, po_4\}$	$\emptyset$	$\emptyset$
$t = 2$	(α^2, β^2)	$\{po_2, po_3, po_4\}$	$\{po_1\}$	$\emptyset$
$t = 3$	(α^3, β^3)	$\{po_2, po_4\}$	$\{po_1, po_3\}$	$\emptyset$

In Table 4, the decision regions for the 4 climate policies are changing with different time t, *e.g.*, po_1 should be executed at $t = 1$, and need further investigated at $t = 2$ and $t = 3$; po_3 should be executed at $t = 1$ and $t = 2$, and need further investigated at $t = 3$. These variations in Table 4 indicate that one should adjust his/her decisions timely with the variations of external decision environment, that is, one policy suits for today, may not makes sense for tomorrow. In a short, one can directly make a decision by comparing the relation between (α^t, β^t) and $Pr^t(X|po_i)$ at one time t, and our proposed model brings an intuitive way to achieve the goal.

5 Conclusions

As an extension of constant numerical values, we introduce dynamic loss function into DTRS to deal with the variations of decisions in practical decision procedure. With respect to the minimum Bayesian expected risk, a model of DDTRS is built. We carefully investigated four scenarios of DDTRS. Furthermore, the corresponding decision criteria of DDTRS under four scenarios are discussed. An example of climate policy making is given to illustrate the proposed model in applications. However, this paper only presents some preliminary ideas on dynamic three-way decision and focuses on the linear cases. By considering the

action in DDTRS would depend on the actions in the previous times and the current situation, the general case of DDTRS need further investigated. Our future research work will focus on developing the DTRS models under sequential decision process and Markov decision process. The attribute reduction based on the DDTRS will be another future work.

Acknowledgements. This work is partially supported by the National Science Foundation of China (Nos. 71201133, 61175047, 71201076), the Youth Social Science Foundation of the Chinese Education Commission (No. 11YJC630127), the Research Fund for the Doctoral Program of Higher Education of China (No. 20120184120028), the China Postdoctoral Science Foundation (Nos. 2012M520310, 2013T60132) and the Fundamental Research Funds for the Central Universities of China (No. SWJTU12CX117).

References

1. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180, 341–353 (2010)
2. Yao, Y.Y.: The superiority of three-way decision in probabilistic rough set models. Information Sciences 181, 1080–1096 (2011)
3. Liang, D.C., Liu, D., Pedrycz, W., Hu, P.: Triangular fuzzy decision-theoretic rough sets. International Journal of Approximate Reasoning 54, 1087–1106 (2013)
4. Liu, D., Li, T.R., Liang, D.C.: Interval-valued decision-theoretic rough sets. Computer Science 39(7), 178–181+214 (2012) (in Chinese)
5. Liu, D., Li, T.R., Li, H.X.: Fuzzy decision-theoretic rough sets. Computer Science 39(12), 25–29 (2012) (in Chinese)
6. Liu, D., Li, T.R., Liang, D.C.: Decision-theoretic rough sets with probabilistic distribution. 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. 389–398. Springer, Heidelberg (2012)
7. Pauker, S., Kassirer, J.: The threshold approach to clinical decision making. The New England Journal of Medicine 302, 1109–1117 (1980)
8. Zhou, B., Yao, Y.Y., Luo, J.G.: A three-way decision approach to email spam filtering. In: Farzindar, A., Kešelj, V. (eds.) Canadian AI 2010. LNCS, vol. 6085, pp. 28–39. Springer, Heidelberg (2010)
9. Liu, D., Yao, Y.Y., Li, T.R.: Three-way investment decisions with decision-theoretic rough sets. International Journal of Computational Intelligence Systems 4, 66–74 (2011)
10. Yao, J.T., Herbert, J.P.: Web-based support systems with rough set analysis. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 360–370. Springer, Heidelberg (2007)
11. Woodward, P.W., Naylor, J.C.: An application of bayesian methods in SPC. The Statistician 42, 461–469 (1993)
12. Liu, D., Li, T.R., Liang, D.C.: Three-way government decision analysis with decision-theoretic rough sets, International Journal of Uncertainty. Fuzziness and Knowledge-Based Systems 20(suppl. 1), 119–132 (2012)

13. Ślęzak, D.: Rough sets and bayes factor. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 202–229. Springer, Heidelberg (2005)
14. Yao, Y.Y.: Probabilistic rough set approximations. International Journal of Approximate Reasoning 49, 255–271 (2008)
15. Ziarko, W.: Probabilistic approach to rough set. International Journal of Approximate Reasoning 49, 272–284 (2008)
16. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximating concepts. Int. J. of Man-Machine Studies 37(6), 793–809 (1992)
17. Liu, D., Li, T.R., Liang, D.C.: Fuzzy interval decision-theoretic rough sets. In: Proceeding of 2013 IFSA World Congress NAFIPS Annual Meeting (accepted, 2013)
18. Yao, Y.Y., Deng, X.F.: Sequential three-way decisions with probabilistic rough sets. In: Proceeding of 10th ICCI, pp. 120–125 (2011)
19. Yao, Y.Y.: Granular computing and sequential three-way decisions. Accepted by this proceeding
20. Li, H.X., Zhou, X.Z., Zhao, J.B., Liu, D.: Cost-sensitive three-way decision: A sequential strategy. Accepted by this proceeding
21. Herbert, J.P., Yao, J.T.: Game-theoretic rough sets. Fundamenta Informaticae 108, 267–286 (2011)
22. Li, H.X., Zhou, X.Z.: Risk decision making based on decision-theoretic rough set: A three-way view decision model. International Journal of Computational Intelligence Systems 4, 1–11 (2011)
23. Liu, D., Li, H.X., Zhou, X.Z.: Two decades' research on decision-theoretic rough sets. In: Proceeding of 9th ICCI, pp. 968–973 (2010)
24. Liu, D., Li, T.R., Ruan, D.: Probabilistic model criteria with decision-theoretic rough sets. Information Sciences 181, 3709–3722 (2011)
25. Liu, D., Li, T.R., Liang, D.C.: Incorporating logistic regression to decision-theoretic rough sets for classifications. International Journal of Approximate Reasoning (2013), doi:10.1016/j.ijar.2013.02.013.
26. Pawlak, Z.: Rough sets. International Journal of Computer and Information Science 11, 341–356 (1982)
27. Yao, Y.Y.: Two semantic issues in a probabilistic rough set model. Fundamenta Informaticae 108, 249–265 (2011)
28. Jeffrey, R.: Savage's omelet. In: Proceedings of the Biennial Meeting of the Philosophy of Science Association. Symposia and Invited Papers, vol. 2, pp. 361–371 (1976)
29. Savage, L.J.: The Foundations of Statistics. Dover Publications, New York (1972)
30. Webster, M.D.: Uncertainty and learning in sequential decision-making: the case of climatie policy. Doctor thesis at Massachusetts Institute of Technology (2000)

A Cluster Ensemble Framework Based on Three-Way Decisions

Hong Yu and Qingfeng Zhou

Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing, 400065, P.R. China
yuhong@cqupt.edu.cn

Abstract. Cluster ensembles can combine the outcomes of several clusterings to a single clustering that agrees as much as possible with the input clusterings. However, little attention has been paid to the development of approaches to deal with consolidating the outcomes of both soft and hard clustering systems into a single final partition. For this reason, this paper proposes a cluster ensemble framework based on three-way decisions, and the interval sets used here to represent the cluster which is described by three regions according to the lower and upper bound of the cluster. In addition, this paper also devises a plurality voting-based consensus function which can consolidate the outcomes of multiple clustering systems whatever the systems are soft clustering systems or hard clustering systems. The proposed consensus function has been evaluated both in the quality of consensus partitions and in the running time.

Keywords: cluster ensemble, three-way decisions, voting-based consensus, interval sets.

1 Introduction

As one of the important branches of multiple classifier systems, the cluster ensemble approach has a strong capability to integrate multiple partitions from different data sources, which has been widely used as a powerful tool to reveal underlying patterns in many areas such as data mining, web mining, geographical data processing, medicine and so on [1]. Compared to single clustering approaches, the cluster ensemble approach has advantages such as robustness, novelty, stability and parallelism.

A recent trend in the field of unsupervised classification is the combination of the outcomes of multiple clustering systems into a single consolidated partition, known as consensus functions [2]. Fred et al. [1] proposed a new clustering algorithm - voting-k-means which can find consistent clusters in data partitions. Zhou and Tang [3] proposed four weighted-voting methods to build ensembles of k-means clusters. Wang et al. [4] proposed a nonparametric Bayesian clustering ensemble (NBCE) method, which can discover the number of clusters in the consensus clustering. Zhou et al. [5] proposed a spectral clustering ensemble

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 302–312, 2013.

method which not only made use of the advantages of spectral clustering dealing with arbitrary distribution data set, but also utilized the good robustness and generalization ability of cluster ensemble.

Despite there are lots of achievements on cluster ensembles, there exist relatively few approaches to consensus clustering specifically oriented to combine the outcomes of multiple soft unsupervised classifiers into a single of consensus partition. Anyway, there are also some scholars who have studied on it using fuzzy sets theory. For example, Sevillano et al. [6] proposed a set of of fuzzy consensus functions using positional and confidence voting techniques, which can combine multiple soft clustering results into a final soft clustering result represented by the membership matrix. Punera and Ghosh [7] proposed a several consensus algorithms that can be applied to soft clusterings by extending the relatively hard clustering methods. Avogadri and Valentini [8] proposed an unsupervised fuzzy ensemble clustering approach, where the fuzzy-set theory was used to express the uncertainty of the data ownership, and other fuzzy tools was used to transform the soft clusterings into hard clusterings. Soft clustering techniques are widely needed in a variety of important applications such as network structure analysis, wireless sensor networks and biological information. The objective of this paper is to propose a cluster ensemble approach that allows to obtain the final consensus clustering result both in hard and soft formats.

On the other hand, a theory of three-way decisions is constructed on the notions of acceptance, rejection and noncommitment. It is an extension of the commonly used binary-decision model with an added third option. Three-way decisions play a key role in everyday decision-making and have been widely used in many fields and disciplines [9]. In fact, considering the relation between an object and a cluster, the object does belong to the cluster certainly, the object does not belong to the cluster certainly, and the object might belong to the cluster. Obviously, it is a typical three-way decisions processing to decide the relation between an object and a cluster. This inspires us to solve clustering using three-way decisions. Furthermore, we had proposed a three-way decision strategy for overlapping clustering based on the decision-theoretic rough set model in [10], where each cluster is described by an interval set that is defined by a pair of sets called the lower and upper bounds, and the clustering method is effective to overlapping clustering.

In many data mining applications, using interval sets to represent clusters can be more appropriate than using crisp representations. Objects in a lower bound are definitely part of the cluster, and objects in a upper bound are possibly part of that cluster and potentially belong to another clusters. The interval sets make it possible to describe ambiguity in categorizing some of the objects. Thus, Lingras and Yan [11] introduced interval sets to represent clusters. Lingras and West [12] proposed an interval set clustering method with Rough K-Means for mining clusters of web visitors. Yao et al. [13] had represented each cluster by an interval set instead of a single set as the representation of a cluster. Chen and Miao [14] studied the clustering method represented as interval sets, wherein the rough k-means clustering method was combined.

In order to obtain a single "consensus" clustering solution from the outcomes of multiple unsupervised classifiers, this paper proposes a cluster ensemble framework based on three-way decisions, which can lead to a final consensus clustering result both in hard and soft formats since the interval sets are used to represent clusters. Furthermore, a voting-based consensus function is proposed, where one object is decided to belong to the positive region of a cluster or belong to a boundary region by traversing the voting matrix first, according to consensus rules designed in this paper.

2 Define Clustering Using Interval Sets

The goal of cluster analysis is to group objects in a universe so that objects in the same cluster are more similar to each other and objects in different clusters are dissimilar.

To define our framework, let a universe be $U = \{\mathbf{x}_1, \cdots, \mathbf{x}_n, \cdots, \mathbf{x}_N\}$, and the clustering result is $\mathbf{C} = \{C^1, \cdots, C^k, \cdots, C^K\}$, which is a family of clusters of the universe. The $\mathbf{x}_n$ is an object which has D attributes, namely, $\mathbf{x}_n = (x_n^1, \cdots, x_n^d, \cdots, x_n^D)$. The x_n^d denotes the value of the d attribute of the object $\mathbf{x}_n$, where $n \in \{1, \cdots, N\}$, and $d \in \{1, \cdots, D\}$.

As we have discussed, using interval sets to represent clusters can be more appropriate than crisp representations, which directly leads to an interpretation in three-way decisions for clustering. Let's review some basic concepts of clustering using interval sets [10].

Use an interval set to represent a cluster in $\mathbf{C}$, namely, C^k is represented by an interval set $[\underline{A}(C^k), \overline{A}(C^k)]$, where $\underline{A}(C^k)$ is the lower bound of the cluster C^k, $\overline{A}(C^k)$ is the upper bound of the cluster C^k, and $\underline{A}(C^k) \subseteq \overline{A}(C^k)$.

The objects in $\underline{A}(C^k)$ may represent typical objects of the cluster C^k, objects in $\overline{A}(C^k) - \underline{A}(C^k)$ may represent fringe objects, and objects in $U - \overline{A}(C^k)$ may represent the negative objects. In other words, the sets $\underline{A}(C^k)$, $\overline{A}(C^k) - \underline{A}(C^k)$ and $U - \overline{A}(C^k)$ are equivalent to the three regions of the cluster C^k as positive region, boundary region and negative region, respectively, which are described as follows.

$$\begin{aligned} POS(C^k) &= \underline{A}(C^k) \\ BND(C^k) &= \overline{A}(C^k) - \underline{A}(C^k) \\ NEG(C^k) &= U - \overline{A}(C^k) \end{aligned} \tag{1}$$

With respect to the family of clusters $\mathbf{C} = \{C^1, \cdots, C^k, \cdots, C^K\}$, we have the following family of clusters formulated by interval sets:

$$\mathbf{C} = \{[\underline{A}(C^1), \overline{A}(C^1)], \ldots, [\underline{A}(C^k), \overline{A}(C^k)], \ldots, [\underline{A}(C^K), \overline{A}(C^K)]\} \tag{2}$$

We adopt the following properties for a cluster in the form of interval set:

$$(i)\ \underline{A}(C^k) \neq \emptyset, 0 \leq k \leq K; \quad (ii)\ \bigcup \overline{A}(C^k) = U.$$

Property (i) implies each cluster cannot be empty. In order to make sure that a cluster is physically meaningful, Property (ii) states that any object of U belongs to the upper bound of a cluster, which ensures that every object belongs to at least one cluster.

According to Eq.(1), the family of clusters $\mathbf{C}$ give a three-way decision clustering result. Namely, objects in $POS(C^k)$ belong to the cluster C^k definitely, objects in $NEG(C^k)$ don't belong to the cluster C^k definitely, and objects in the region $BND(C^k)$ might belong to the cluster or not. The $BND(C^k) \neq \emptyset$ means we need more information to help making decisions.

When $k \neq t$, if $\underline{A}(C^k) \cap \underline{A}(C^t) \neq \emptyset$, or $BND(C^k) \cap BND(C^t) \neq \emptyset$, that means there exists at least one object belonging to more than one cluster and it is a soft clustering. Otherwise, it is a hard clustering.

3 The Cluster Ensemble Framework

In this section, we propose a cluster ensemble framework based on three-way decisions. Figure 1 shows the basic setup of the cluster ensemble.

Let $U = \{\mathbf{x}_1, \cdots, \mathbf{x}_n, \cdots, \mathbf{x}_N\}$ denote a set of objects/samples/points. In order to increase diversity of ensemble clusterings, we use sampling method on the data set U firstly. Then, H samples are obtained, and let U' be the family of samples, namely, $U' = \{U_1, U_2, \cdots, U_h, \cdots, U_H\}$, and $U_h \subseteq U$, and $h \in \{1, \cdots, H\}$.

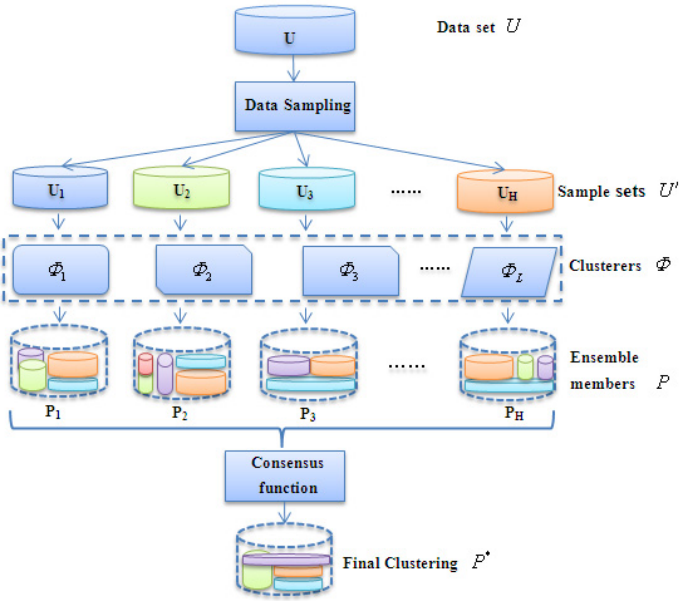


Fig. 1. The framework of the cluster ensemble

Let Φ be a clusterer selection function from a sample to a clusterer. Usually, the Φ is composed by a set of clustering algorithms and denoted as $\Phi = \{\Phi_1, \cdots, \Phi_l, \cdots, \Phi_L\}$, and a clusterer Φ_l is selected randomly or according to a priori knowledge of the data set.

Then, the clustering result of a clusterer Φ_l is called a clustering, also called an ensemble member, denoted by P_h. Let $P = \{P_h | \Phi_l(U_h) \to P_h\}$ be the set of clusterings, where $h \in \{1, \cdots, H\}, l \in \{1, \cdots, L\}$. Assume K^h is the number of clusters of the clustering P_h, the P_h is described as: $P_h = \{C_h^1, \cdots, C_h^k, \cdots, C_h^{K^h}\} = \{[\underline{A}(C_h^1), \overline{A}(C_h^1)], \cdots, [\underline{A}(C_h^k), \overline{A}(C_h^k)], \cdots, [\underline{A}(C_h^{K^h}), \overline{A}(C_h^{K^h})]\}$.

Finally, the labeling clusterings are combined into a single labeling clustering P^* using a consensus function $F : F(P_1, P_2, \cdots, P_{H'}) \to P^*$. Here, $H' \leq H$, because when combining clusterings into a final clustering, we may combine all ensemble members or some of ensemble members. P^* is represented as: $P^* = \{C^1, \cdots, C^k, \cdots, C^{K^*}\} = \{[\underline{A}(C^1), \overline{A}(C^1)], \cdots, [\underline{A}(C^k), \overline{A}(C^k)], \cdots, [\underline{A}(C^{K^*}), \overline{A}(C^{K^*})]\}$.

Obviously, the lower bounds and upper bounds of the clustering satisfy Property (i) and (ii). In other words, the framework of cluster ensemble based on three-way decisions not only can represent the soft clustering, but also can represent the hard clustering.

4 Voting-Based Consensus Function

This section introduces a consensus function which combines the outcomes of multiple clustering systems into a single consolidated partition. The advantage of the proposed consensus function is that it can consolidate the ensemble members whenever the members are soft or hard clusterings since the cluster is defined by an interval set.

The consensus function decides an object to a cluster based on the plurality voting system. The process of partitioning is regarded as an election. Each of the clusterers is a voter and casts a vote for an object to a cluster when the object is assigned to the cluster. Therefore, if an object is assigned into the same cluster by most of the voters, the consensus clustering process should respect that decision; that is, the object is decided to belong to the cluster by the consensus function. Obviously, consensus functions based on voting strategies must include a cluster disambiguation process prior to voting proper, and the cluster disambiguation process is conducted by using the method in reference [15] in our experiments.

Thus, the input of the consensus function is the set of ensemble members P, and the output is the final clustering result P^*. Now we consider a specious case, the number of clusters of all ensemble members and the final clustering is K, which is the real number of clusters of the data set. Every ensemble member includes K clusters and every cluster consists of three regions: the positive region, the boundary region and the negative region. When an object is partitioned into one region of a cluster in an ensemble member system, the object gets a vote from the ensemble member system to the region of the cluster. Thus, the H ensemble members give expression to all votes in the K clusters for all objects.

In order to express the votes, two matrixes are used here such as: the positive voting matrix $\mathbf{V\_Pos}_{N\times K}$ and the boundary voting matrix $\mathbf{V\_Bnd}_{N\times K}$, where $k \in \{1, 2, \cdots, K\}$, $n \in \{1, 2, \cdots, N\}$. The value $V\_Pos_n^k$ of matrix **V_Pos** means the votes for the object $\mathbf{x}_n$ in the positive region of cluster C^k; the value $V\_Bnd_n^k$ of matrix **V_Bnd** means the votes for the object $\mathbf{x}_n$ in the boundary region of cluster C^k. Set $\mathbf{Z} = \{Z_1, \cdots, Z_n, \cdots, Z_N\}$ be the total votes for all objects, namely, the value Z_n means how many ensemble members (or voters) cast their votes for the object $\mathbf{x}_n$.

We can scan the H ensemble members and get votes for all objects in the K clusters, then the positive voting matrix **V_Pos**, the boundary voting matrix **V_Bnd** and the total votes vector **Z** are constructed. The next, the consensus function can make decisions for building the final clustering P^*. The basic idea is that to scan every object in voting matrixes and assign it to corresponding region (positive or boundary) of a cluster according to the votes in the K clusters.

For every object $\mathbf{x}_n$, the specific decision rules are described as follows.

Case 1: $if\ \exists k(V\_Pos_n^k > Z_n/2), then\ decide\ \mathbf{x}_n \in Pos(C^k)$;

When votes for $\mathbf{x}_n$ in the positive region of a cluster are greater than a half of its total votes, that means there is at most one cluster which $\mathbf{x}_n$ can belong to, then we decide the $\mathbf{x}_n$ to belong to the positive region of the cluster.

Case 2: $if\ \exists k(V\_Pos_n^k = Z_n/2)$

When votes for $\mathbf{x}_n$ in the positive region of clusters are equal to a half of its total votes, there are at most two clusters which $\mathbf{x}_n$ can belong to, so we randomly select one cluster to consider. Now there are two other cases:

1) $if\ (V\_Pos_n^k > V\_Bnd_n^k\ and\ V\_Bnd_n^k \neq 0),\ then\ \mathbf{x}_n \in Pos(C^k)$;

When votes for $\mathbf{x}_n$ in the positive region of cluster are greater than its votes in the boundary region of the cluster and its votes in the boundary region of cluster is not 0, we decide that the $\mathbf{x}_n$ belongs to the positive region of the cluster.

2) $if\ (V\_Pos_n^k = V\_Bnd_n^k\ or\ V\_Bnd_n^k = 0),\ then\ \mathbf{x}_n \in Bnd(C^k)$;

When votes for $\mathbf{x}_n$ in the positive region of cluster are equal to its votes in the boundary region of cluster or its votes in the boundary region of cluster is 0, we decide that the $\mathbf{x}_n$ belongs to the boundary region of the cluster.

Case 3: $if\ \forall k(V\_Pos_n^k < Z_n/2)\ and\ A \neq \emptyset$,
then decide $\mathbf{x}_n$ *to belong to the boundary of clusters in the set* A;

Set $A = \{C^l | V\_Bnd_n^l \geq Z_n/2,\ l \in \{1, 2, \cdots, K\}\}$, which is a set of clusters, where the votes for $\mathbf{x}_n$, in the boundary region of cluster in A, are no less than a half of its total votes. When votes for $\mathbf{x}_n$ in the positive region of the cluster are less than a half of its total votes, and there exists an A, then we decide that $\mathbf{x}_n$ belongs to the boundary of clusters in the set A.

Case 4: $if\ \forall k(V\_Pos_n^k < Z_n/2)\ and\ \forall k(V\_Bnd_n^k < Z_n/2)$

Set $B = \{C^l | \mathbf{x}_n \in Bnd(C^l),\ l \in \{1, 2, \cdots, K\}\}$ be a set of clusters where $\mathbf{x}_n$ belongs to the boundary region of these clusters. When votes for $\mathbf{x}_n$ in the positive region of every cluster are less than a half of its total votes, and votes for $\mathbf{x}_n$ in the boundary region of every cluster are also less than a half of its total votes, there are also two other cases:

1) $if\ (V\_Pos_n^k \geq Z_n/4), then\ \mathbf{x}_n \in Bnd(C^k)$;

When votes for $\mathbf{x}_n$ in the positive regions of clusters reach a certain number, $Z_n/4$ used in experiments, we decide that the $\mathbf{x}_n$ belongs to the boundary region of clusters in the set B.

2) $if(V\_Pos_n < Z_n/4)), then\, \mathbf{x}_n\, belongs\, to\, the\, boundary\, of\, clusters\, in\, the\, B$;

When votes for $\mathbf{x}_n$ in the positive region of clusters do not reach a certain number, $Z_n/4$ used in experiments, we decide that the $\mathbf{x}_n$ belongs to the boundary region of clusters in the set B.

Through the study of the four cases, the voting-based consensus function can combine multiple clusterings to a final clustering which may be a hard clustering or a soft clustering.

5 Experiments

This section will describe some experimental results to show the performance of the proposed plurality voting-based consensus function. The accuracy is used to evaluate the clusterings in the experiments, and we only conclude objects in the positive region of clusters when calculating the accuracy of clusterings because there might be some objects in the boundaries of clusters. We considered a special case that all samples were equal to the original data set, namely we don't use any sampling methods in our experiments.

Experiment 1. The first experiment is on a synthetic data set to show the main idea of the method, which can process the fuzzy boundaries of clusters. The data set MD1 is illustrated in Figure 2, which consists of 323 objects and three clusters where there are some indistinct objects between the two clusters.

The RK-Means algorithm [12] is used as clusterers in the experiment, the number of ensemble members are 20 and the threshold used by the RK-Means algorithm is 1.0. Figure 3 depicts the final clustering of the MD1, where each cluster is described by two regions as the positive and the boundary. Obviously, the nine objects on the boundary of $C1$ and $C3$ are found out and assigned into the boundary objects for the two clusters by the method, which accords with the fact.

In order to further explain the performance of the consensus function, we observe the every ensemble members and find that the accuracy of the best

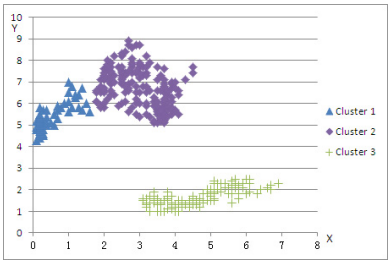


Fig. 2. The original data set MD1

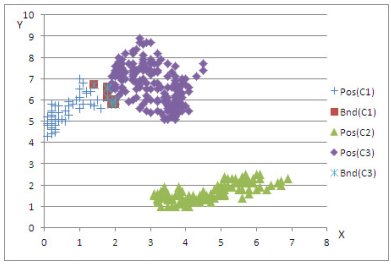


Fig. 3. The final clustering of the MD1

ensemble member clustering is equal to the final clustering. In additional, Figure 4 and Figure 5 show the results of two ensemble members. The two ensemble members partition objects in the upper part of the original MD1 into one cluster and in the lower part of MD1 into two clusters. Observe the results in Figure 4, the original $C3$ is divided into two clusters. The boundaries of $C1$ and $C3$ in Figure 5 are big and are the same region. Fortunately, we obtain a good final clustering result after consensus combining in Figure 3.

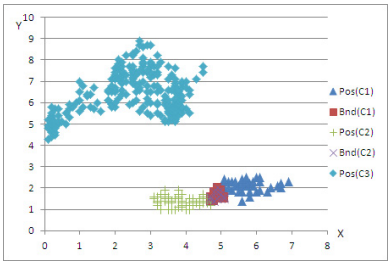


Fig. 4. The results of one ensemble member on the MD1

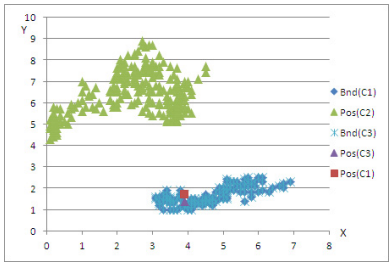


Fig. 5. The results of another ensemble member on the MD1

Experiment 2 The experiments have been conducted on 13 publicly available data collections obtained from the UCI Machine Learning Repository [16] which are commonly employed as benchmarks in the pattern recognition and machine learning fields. The detail description of UCI data sets shows in Table 2. The Pendigits1234 and PenDigits1469 data set are subsets of the Pendigits data set. They both consist of 4 clusters, where the former contains digits 1 to 4 and the latter contains digits 1, 4, 6 and 9.

Table 1. Different Clusterers Φ

Clusterers Φ	Algorithms in the Clusterers
$\Phi 1$	{K-means}
$\Phi 2$	{K-means, K-medoids, Ward}
$\Phi 3$	{RK-means, K-means, K-medoids, Complete Link}
$\Phi 4$	{K-means, K-medoids, Average Link}
$\Phi 5$	{K-means, K-medoids, Single Link}
$\Phi 6$	{K-means, K-medoids}

We use soft clustering algorithms and hard clustering algorithms as clusterers in our experiments. That is, the RK-Means algorithm [12], K-means algorithm, K-medoids algorithm, and some hierarchical clustering algorithms such as Single Link method, Average Link method, Complete Link method and Ward method [17], are used as clusterers in experiments. We can choose some of the clustering

algorithms as clusterers randomly, or just choose appropriate clustering algorithms according to the prior knowledge. In this paper, the random selection method is used, and the ten-fold cross validation method is also used in each test where there are 60 ensemble members generated.

Table 1 shows the different clustering algorithms used in the different clusterers Φ. The results are shown in Table 2. The column Φ describes the clusterers used in each test for the data set. N, D and K denotes the number of objects in the data set, the number of attributes and the number of clusters of the data set, respectively. $AVG(Accuracy(P^*))$ denotes the average of the accuracies of the final clustering P^* for all tests, and $AVG(Accuracy(P))$ denotes the average of the accuracies of all ensemble members P, where $\pm$ means the standard deviations. The time(s) denotes the running time of the consensus function and the second is a unit of time.

Observe the results, for all data sets except for the first and tenth data sets, the accuracy of the final clustering P^* obtained by the proposed consensus function is higher than or equal to the accuracy of corresponding ensemble members P. For the first data set, the accuracy of the final clustering P^* obtained by the proposed consensus function is higher than the accuracy of corresponding

Table 2. Results of the consensus function on the UCI data sets

NO.	Dataset	N	D	K	Φ	$AVG(Accuracy(P^*))$	$AVG(Accuracy(P))$	time(s)
1	Zoo	101	17	7	$\Phi 1$	0.75 ± 0.06	0.70 ± 0.01	0.00 ± 0.01
					$\Phi 2$	0.61 ± 0.06	0.63 ± 0.01	0.00 ± 0.01
2	Wine	178	13	3	$\Phi 1$	0.70 ± 0.00	0.66 ± 0.01	0.00 ± 0.00
					$\Phi 2$	0.70 ± 0.01	0.68 ± 0.01	0.00 ± 0.01
3	IRIS	150	4	3	$\Phi 1$	0.89 ± 0.00	0.80 ± 0.03	0.00 ± 0.01
					$\Phi 2$	0.87 ± 0.04	0.81 ± 0.01	0.00 ± 0.01
					$\Phi 3$	0.88 ± 0.03	0.83 ± 0.01	0.01 ± 0.01
4	LiverDisorders	345	6	2	$\Phi 1$	0.55 ± 0.00	0.55 ± 0.00	0.01 ± 0.01
					$\Phi 4$	0.55 ± 0.01	0.55 ± 0.00	0.00 ± 0.00
5	Ionoshpere	351	34	2	$\Phi 1$	0.71 ± 0.00	0.71 ± 0.00	0.00 ± 0.01
					$\Phi 4$	0.70 ± 0.03	0.67 ± 0.01	0.00 ± 0.00
6	WDBC	569	30	2	$\Phi 1$	0.85 ± 0.00	0.85 ± 0.00	0.00 ± 0.00
					$\Phi 5$	0.85 ± 0.00	0.78 ± 0.00	0.00 ± 0.01
7	Image Segmentation	2310	19	7	$\Phi 1$	0.61 ± 0.02	0.52 ± 0.00	0.00 ± 0.01
					$\Phi 6$	0.60 ± 0.04	0.52 ± 0.01	0.00 ± 0.01
8	PenDigits1469	4398	16	4	$\Phi 1$	0.83 ± 0.07	0.83 ± 0.01	0.01 ± 0.01
					$\Phi 6$	0.87 ± 0.05	0.79 ± 0.02	0.01 ± 0.01
9	PenDigits1234	4486	16	4	$\Phi 1$	0.87 ± 0.00	0.80 ± 0.01	0.01 ± 0.01
					$\Phi 6$	0.86 ± 0.03	0.79 ± 0.01	0.00 ± 0.01
10	Waveform-21	5000	21	3	$\Phi 1$	0.39 ± 0.00	0.40 ± 0.01	0.01 ± 0.01
					$\Phi 6$	0.70 ± 0.01	0.68 ± 0.01	0.00 ± 0.01
11	PageBlocks	5473	10	5	$\Phi 1$	0.73 ± 0.00	0.73 ± 0.00	0.01 ± 0.01
					$\Phi 6$	0.69 ± 0.07	0.56 ± 0.03	0.01 ± 0.01
12	Landsat	6435	36	6	$\Phi 1$	0.68 ± 0.00	0.64 ± 0.01	0.01 ± 0.01
13	PenDigits	11092	16	10	$\Phi 1$	0.79 ± 0.03	0.69 ± 0.01	0.02 ± 0.00

ensemble members P in the first experiment.However the accuracy of the final clustering P^* was slightly worse to the accuracy of corresponding ensemble members P in the second experiment. Besides, for the tenth data set, the accuracy of the final clustering P^* obtained by the proposed consensus function is higher than the accuracy of corresponding ensemble members P in the second experiment. And the accuracy of the final clustering P^* was slightly worse to the accuracy of corresponding ensemble members P in the first experiment. In addition, the accuracy of the second experiment was higher than the first experiment. Selecting different clusterers was more effective in the second experiment of the tenth data set. In sum, the results after combining by consensus function is much better than the single clustering. Furthermore, computing consensus function spend little time, which is helpful to cluster on big data sets.

6 Conclusion

Cluster ensembles can combine the outcomes of several clusterings to a single clustering. In order to develop an approach to deal with consolidating the outcomes of both soft and hard clustering systems into a single final partition, this paper proposes a cluster ensemble framework based on three-way decisions, and the interval sets used here to represent the cluster have advantages to describe both hard clustering and soft clustering. According to the lower and upper bound of the cluster, a cluster is described by three regions: objects in the positive region belong to the cluster certainly, objects in the negative region do not belong to the cluster certainly, and objects in the boundary region defer decisions since the information is not sufficient. Besides, this paper also proposes a plurality voting-based consensus function, which can consolidate the outcomes of cluster ensemble systems, and the systems contain only soft clustering systems or only hard clustering systems or both soft and hard clustering systems. The experimental results show that the method is effective both in the quality of the consensus partitions and in the running time. However, how to determine the number of ensemble members, how to delete the "bad" votes in ensemble members, and how to improve the time complexity of the consensus function to be available to big data sets are still problems needed to study in the further work.

Acknowledgments. This work was supported in part by the China NSFC grant (No.61272060), and the Chongqing CSTC grant (no.cstc2013jcyjA40063).

References

1. Fred, A.: Finding consistent clusters in data partitions. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 309–318. Springer, Heidelberg (2001)
2. Strehl, A., Ghosh, J.: Cluster ensembles - A knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research 3, 583–617 (2003)

3. Zhou, Z.H., Tang, W.: Clusterer ensemble. Knowledge-Based Systems 19(1), 77–83 (2006)
4. Wang, H.J., Shan, H.H.: Bayesian cluster ensembles. Statistical Analysis and Data Mining 4(1), 54–70 (2011)
5. Zhou, L., Ping, X.J., Xu, S., Zhang, T.: Cluster ensemble based on spectral clustering. Acta Automatica Sinica 38(8), 1335–1342 (2012) (in Chinese)
6. Sevillano, X., Alías, F., Socoró, J.C.: Positional and confidence voting-based consensus functions for fuzzy cluster ensembles. Fuzzy Sets and Systems 193, 1–32 (2012)
7. Punera, K., Ghosh, J.: Consensus-based ensembles of soft clusterings. Applied Artificial Intelligence 22(7-8), 780–810 (2008)
8. Avogadri, R., Valentini, G.: Ensemble clustering with a fuzzy approach. In: Okun, O., Valentini, G. (eds.) Supervised and Unsupervised Ensemble Methods and their Applications 2008. SCI, pp. 49–69. Springer, Heidelberg (2008)
9. Yao, Y.Y.: An outline of a theory of three-way decisions. In: Yao, J.T., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 1–17. Springer, Heidelberg (2012)
10. Yu, H., Wang, Y.: Three-way decisions method for overlapping clustering. In: Yao, J.T., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 277–286. Springer, Heidelberg (2012)
11. Lingras, P., Yan, R.: Interval clustering using fuzzy and rough set theory. In: Proceeding IEEE Annual Meeting of the Fuzzy Information 2004, Banff, Alberta, vol. 2, pp. 780–784 (2004)
12. Lingras, P., West, C.: Interval set clustering of web users with rough k-means. Journal of Intelligent Information Systems 23(1), 5–16 (2004)
13. Yao, Y.Y., Lingras, P., Wang, R.Z., Miao, D.Q.: Interval set cluster analysis: A reformulation. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS, vol. 5908, pp. 398–405. Springer, Heidelberg (2009)
14. Chen, M., Miao, D.Q.: Interval set clustering. Expert Systems with Applications 38(4), 2923–2932 (2011)
15. Tang, W., Zhou, Z.H.: Bagging-based selective clusterer ensemble. Journal of Software 16(4), 496–502 (2005) (in Chinese)
16. UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml`
17. DeSá, J.P.M.: Pattern Recognition: Concepts, Methods, and Applications. Springer, Heidelberg (2001)

Multistage Email Spam Filtering Based on Three-Way Decisions

Jianlin Li[1], Xiaofei Deng[2], and Yiyu Yao[2]

[1] School of Computer and Software
Nanjing College of Information Technology
Nanjing, China, 210023
lijl@njcit.cn
[2] Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada, S4S 0A2
{deng200x,yyao}@cs.uregina.ca

Abstract. A ternary, three-way decision strategy to email spam filtering divides incoming emails into three folders, namely, a mail folder consisting of emails that we *accept* as being legitimate, a spam folder consisting of emails that we *reject* as being legitimate, and a third folder consisting of emails that we cannot accept nor reject based on available information. The introduction of the third folder enables us to reduce both acceptance and rejection errors. Many existing ternary approaches are essentially a single-stage process. In this paper, we propose a model of multistage three-way email spam filtering based on principles of granular computing and rough sets.

1 Introduction

An email spam filtering system automatically processes incoming emails according to certain criteria and classifies and organizes emails into several folders. Many studies [1, 3, 12, 15, 16] treat spam filtering as a binary, two-way decision/classification so that approaches from machine learning can be conveniently applied. For a two-way decision, one can either accept a message as being legitimate, or reject the message as being legitimate (i.e., spam) by using a single threshold. A trade-off between incorrect acceptance and incorrect rejection is introduced by the single threshold. For example, a larger threshold on the probability of legitimacy of emails typically leads to a lower rate of incorrect acceptance but a higher rate of incorrect rejection; the reverse is true for a smaller threshold. As discussed in [4], for two-way decisions one can not decrease incorrect acceptance and incorrect rejection errors simultaneously. Therefore, the idea of ternary, three-way decision/classification arises and attracts attention from many authors [2, 7–11, 14, 22–25].

Existing three-way decision approaches to email spam filtering are a single-stage process and mainly focus on two fundamental issues, namely, constructing a function that estimates the legitimacy of emails [25] and determining an optimal pair of thresholds. Based on a framework of sequential three-way decisions

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 313–324, 2013.

[20, 21], this paper introduces a multistage three-way email spam filtering model by taking advantages of multiple levels of granularity existed in emails.

Human beings usually make effective decisions based on available information and search for more evidence when it is impossible to make a decision. This observation implies that we make decisions in multiple steps/stages. To formally describe such a decision-making process, a sequential three-way decision model, an extension of probabilistic rough sets, is proposed and studied [20, 21]. The multistage three-way email spam filtering method of this paper is an application of the model of sequential three-way decisions. We construct multiple representations of the same email at different levels of granularity by adding additional information, starting from sender and moving to subject, to main text, and to attachments. We can make either an acceptance or a rejection decision at an abstract higher level with a coarser granulation when we are confident enough; otherwise, we make a non-commitment decision and move to a lower level with more detailed information.

The rest of the paper is organized as follows. Section 2 examines and analyzes existing single-stage two-way and three-way decision approaches. Section 3 proposes a multistage three-way decision model. Section 4 introduces an approach to construct multilevel granulations in support of the multistage three-way decision-making.

2 Single-Stage Email Spam Filtering

Email spam filtering can be formulated as a classification problem. Suppose U is a finite non-empty set of objects, called the universe, and each object in U represents an email message. We can use a feature vector $\boldsymbol{x} = \{x_1, x_2, \cdots, x_n\}$ to describe an object $x \in U$, where x_i with $1 \leq x_i \leq n$ is the i-th feature or attribute of an email. For example, x_1 represents the sender of a message, x_2 represents the subject of the message. According to [19], we assume each object x has two states, legitimate, represented by $x \in C$, or spam, represented by $x \in C^c$. A classifier predicts state of objects by using either a discriminate function [25] or an estimation function [19], denoted as $v(C|x)$. This section introduces binary, two-way decisions and ternary, three-way decisions to email spam filtering.

2.1 Email Spam Filtering as a Binary, Two-Way Decision

By introducing a single threshold $0 \leq \gamma \leq 1$, we can formulate two-way decision regions as follows:

$$\begin{aligned} \mathrm{POS}_\gamma(C) &= \{x \in U \mid v(C|x) \geq \gamma\}, \\ \mathrm{NEG}_\gamma(C) &= \{x \in U \mid v(C|x) < \gamma\}. \end{aligned} \tag{1}$$

The positive region $\mathrm{POS}_\gamma(C)$ can be considered as the folder consisting of emails that we accept as being legitimate, while the negative region $\mathrm{NEG}_\gamma(C)$ can be considered as another folder consisting of emails that we reject as being legitimate (i.e., being treated as a spam).

For each object $x \in U$, we can use the following two-way decision rules:

Acceptance: If $v(C|x) \geq \gamma$, accept x as being legitimate, i.e., $x \in C$;

Rejection: If $v(C|x) < \gamma$, reject x as being legitimate, i.e., $x \in C^c$.

Tie-breaking rules can be applied to these two rules. A normalized function $v(C|x)$ provides an estimated value for the two-way decisions. According to different applications, one may interpret and construct different estimation or discriminate functions. For example, a SVM classifier [3] uses a measure of distance between the given email and the decision hyperplane, while a naive Bayes classifier [12, 15] uses probabilistic classification techniques.

2.2 Email Spam Filtering as a Ternary, Three-Way Decision

The three-way decisions [19] allow an additional decision option, namely, non-commitment, when the support information is insufficient for either an acceptance nor a rejection decision. By introducing a pair of thresholds (α, β) with $0 \leq \beta < \alpha \leq 1$, we can define the positive, negative and boundary three-way decision regions as follows:

$$\begin{aligned} \mathrm{POS}_{(\alpha,\cdot)}(C) &= \{x \in U \mid v(C|x) \geq \alpha\}, \\ \mathrm{NEG}_{(\cdot,\beta)}(C) &= \{x \in U \mid v(C|x) \leq \beta\}, \\ \mathrm{BND}_{(\alpha,\beta)}(C) &= \{x \in U \mid \beta < v(C|x) < \alpha\}, \end{aligned} \tag{2}$$

where we use an '$\cdot$' to represent an irrelevant threshold and $v(C|x)$ is a normalized estimation function. For objects in the boundary region $\mathrm{BND}_{(\alpha,\beta)}(C)$, we need to obtain more evidence to make a definite decision, i.e., either an acceptance or a rejection. The cardinality of the boundary region depends on the pair of thresholds (α, β), while the positive and negative regions depend on α and β, respectively.

For each object $x \in U$, we use the following three-way decision rules:

Acceptance: If $v(C|x) \geq \alpha$, accept x as being legitimate, i.e., $x \in C$;

Rejection: If $v(C|x) \leq \beta$, reject x as being legitimate, i.e., $x \in C^c$;

Non-commitment: If $\beta < v(C|x) < \alpha$, neither accept nor reject x as being legitimate, instead, opt for a non-commitment decision.

The decision of an email depends on the acceptable level of confidence. For example, an email $x \in U$ is accepted as being legitimate if its estimated value $v(C|x)$ is at or above α level, rejected if the value is at or below the β level, and neither accepted nor rejected if the evidence is insufficient, i.e., $\beta < v(C|x) < \alpha$.

Ternary, three-way decision approach to email spam filtering receives attention from many researchers. For example, Robinson [14] proposes a measure of spamminess and construct an estimation function of each email based on probability and spamminess. A ternary classifier is used to classify an email as spam if the estimated value is near 1, is classified as legitimate if it is near 0 and is classified as uncertain if it is near 0.5. Zhao and Zhang [24] propose a three-way

decision approach using the genetic algorithm and rough set theory. They classify incoming emails into three categories, namely, spam, non-spam and suspicious. Zhou et al. [25] propose a cost-sensitive three-way decision approach based on the well-established Bayesian decision theory.

2.3 Motivation of the Multistage Email Spam Filtering

In order to illustrate the purpose of introducing the multistage three-way decisions, we discuss several issues of the single-stage decision-makings in terms of a special case of three-way decisions, i.e., probabilistic rough sets [17]. The following discussion draws from our earlier work [4].

Suppose C is the legitimate email class and $v(C|x) = Pr(C|x)$ is the conditional probability of C given x. We use $Pr(C|x)$ as the estimation function for single-stage decision-makings. Based on Equation (1), the incorrect classification errors of acceptance and rejection for two-way decisions can be respectively defined by: for $0 \leq \gamma \leq 1$,

$$\begin{cases} \text{Incorrect-Acceptance Error}: & \text{IAE}(C, \text{POS}_\gamma(C)) = \frac{|C^c \cap \text{POS}_\gamma(C)|}{|\text{POS}_\gamma(C)|}, \\ \text{Incorrect-Rejection Error}: & \text{IRE}(C, \text{NEG}_\gamma(C)) = \frac{|C \cap \text{NEG}_\gamma(C)|}{|\text{NEG}_\gamma(C)|}. \end{cases} \tag{3}$$

Intuitively, we tend to make a binary decision with minimum errors. That is, we should decrease both $\text{IAE}(C, \text{POS}_\gamma(C))$ and $\text{IRE}(C, \text{NEG}_\gamma(C))$. Unfortunately, this is not always possible due to the following monotonicity for binary, two-way classifications [4]: for $\gamma_1, \gamma_2 \in [0, 1]$,

$$\begin{aligned} &\text{(M1)} \quad \gamma_1 \geq \gamma_2 \Longrightarrow \text{IAE}(C, \text{POS}_{\gamma_1}(C)) \leq \text{IAE}(C, \text{POS}_{\gamma_2}(C)); \\ &\text{(M2)} \quad \gamma_1 \geq \gamma_2 \Longrightarrow \text{IRE}(C, \text{NEG}_{\gamma_1}(C)) \geq \text{IRE}(C, \text{NEG}_{\gamma_2}(C)). \end{aligned}$$

Properties (M1) and (M2) confirm that we can not reduce the incorrect classification errors of the acceptance and rejection simultaneously.

Three-way decisions solve this issue by providing a non-commitment option. The following monotonic properties hold [4]: for $0 \leq \beta < \alpha \leq 1$,

$$\begin{aligned} &\text{(M3)} \quad \alpha_1 \geq \alpha_2 \Longrightarrow \text{IAE}(C, \text{POS}_{(\alpha_1, \cdot)}(C)) \leq \text{IAE}(C, \text{POS}_{(\alpha_2, \cdot)}(C)), \\ &\text{(M4)} \quad \beta_1 \geq \beta_2 \Longrightarrow \text{IRE}(C, \text{NEG}_{(\cdot, \beta_1)}(C)) \geq \text{IRE}(C, \text{NEG}_{C, (\cdot, \beta_2)}(C)); \end{aligned}$$

According to properties (M3) and (M4), one can reduce the incorrect classification errors of acceptance and rejection by adjusting the α and β thresholds at the same time. However, the non-commitment decision leaves the issue of classifying uncertain emails to the boundary region. Whenever we are not sure about the email, we put it into the boundary region. An email filtering system needs to explore sufficient information and finally make either an acceptance or a rejection for those emails. A single-stage three-way decision model does not provide any solution for this issue, as a result, we need a model of multistage three-way decisions.

3 A Model of Multistage Email Spam Filtering

Using the framework of sequential three-way decisions [20], we introduce a model of multistage three-way decisions for email spam filtering.

3.1 Email Spam Filtering Based on a Sequence of Attributes

We assume an email message contains the following set of attributes:

$$\begin{aligned} AT = \{ \; & sender, receiver, subject, date\ of\ sending, date\ of\ receiving, \\ & length, body\ (content), attachment \ldots\}. \end{aligned} \tag{4}$$

By selecting different subsets of attributes from AT, we can make a sequence of subsets of attributes as follow:

$$P_1 \subset P_2 \subset \cdots \subset P_m \subseteq AT, \tag{5}$$

where, for example, we have:

$$\begin{aligned} P_1 &= \{sender\}, \\ P_2 &= \{sender, subject\}, \\ P_3 &= \{sender, subject, length\}, \\ P_4 &= \{sender, subject, length, date\ of\ sending\}, \\ &\cdots \\ P_m &= \{sender, subject, length, date\ of\ sending, \cdots, body\}. \end{aligned} \tag{6}$$

Based on Equation (5), one can make a sequence of descriptions of $x \in U$ satisfying the following condition:

$$N_a(\mathrm{Des}_{P_1}(x)) \leq N_a(\mathrm{Des}_{P_2}(x)) \leq \cdots \leq N_a(\mathrm{Des}_{P_m}(x)), \tag{7}$$

where $\mathrm{Des}_{P_i}(x)$ is a description of x based on $P_i \subseteq AT$, $N_a(\cdot)$ is the number of attributes used in $\mathrm{Des}_{P_i}(x)$. The more attributes we use, the more evidence the description provides. Thus, we can form multiple levels of information granularity. At a higher level we have more abstract information, while at a lower level we use more detailed information.

3.2 Non-monotonicity of Estimations

An exploration of an estimation function may help to make appropriate decisions. Intuitively, the distribution of objects and the legitimate class C determine whether a new piece evidence is valuable. Decision monotonicity probably does not hold for estimation functions at different levels. In [21], we discuss the non-monotonicity of estimations in terms of conditional probabilities. As a

generalization of the conditional probability, the following three scenarios may happen to an estimation function $v(\mathrm{Des}_P(x))$: if $P_1 \subset P_2 \subseteq AT$,

$$\begin{aligned} &v(\mathrm{Des}_{P_2}(x)) > v(\mathrm{Des}_{P_1}(x)) \\ &v(\mathrm{Des}_{P_2}(x)) = v(\mathrm{Des}_{P_1}(x)) \\ &v(\mathrm{Des}_{P_2}(x)) < v(\mathrm{Des}_{P_1}(x)). \end{aligned} \tag{8}$$

The new evidence may support, be neutral, and refutes C, although P_2 contains more attributes than P_1. The observation reveals that we may make incorrect decisions at a higher level, and need to revise our decisions at a lower level.

We prefer to avoid revisions made at a higher level and only allows small chances of revisions at lower levels. Since the three-way decision-makings are based on acceptable levels of incorrect classification errors, one can adjust thresholds at higher levels and only allow low rates of incorrect classification errors. Therefore, we suggest the following conditions of thresholds:

$$\begin{aligned} &0 \leq \beta_i < \alpha_i \leq 1, \quad 1 \leq i \leq m, \\ &\beta_1 \leq \beta_2 \leq \ldots \leq \beta_m < \alpha_m \leq \ldots \leq \alpha_2 \leq \alpha_1. \end{aligned} \tag{9}$$

With high α and low β values, decisions made at higher levels are biased towards the non-commitment option; while, at lower levels, we can make more accurate decisions with the support of more evidence. One can use the decision-theoretic approach [19, 20] to determine the optimal pair of thresholds (α_i, β_i) at each decision stage. There are many discussions on this topic, for example, see [2, 4, 7, 9, 19, 20].

3.3 Multistage Three-Way Decisions

Given a sequence of descriptions $\{\mathrm{Des}_{P_i}(x)\}$ and a sequence of thresholds $\{(\alpha_i, \beta_i)\}$, the multistage three-way decision regions can be recursively defined by:

1. The initialization:

$$\mathrm{MPOS}_0(C) = \emptyset,\ \mathrm{MNEG}_0(C) = \emptyset,\ \mathrm{MBND}_0(C) = U \tag{10}$$

2. Decision stages: suppose (α_i, β_i) for the i-th stage,

$$\begin{aligned} \mathrm{MPOS}_i(C) &= \mathrm{MPOS}_{i-1}(C) \cup \{x \in \mathrm{MBND}_{i-1}(C) \mid v(\mathrm{Des}_{P_i}(x)) \geq \alpha_i\}, \\ \mathrm{MNEG}_i(C) &= \mathrm{MNEG}_{i-1}(C) \cup \{x \in \mathrm{MBND}_{i-1}(C) \mid v(\mathrm{Des}_{P_i}(x)) \leq \beta_i\}, \\ \mathrm{MBND}_i(C) &= \{x \in \mathrm{MBND}_{i-1}(C) \mid \beta_i < v(\mathrm{Des}_{P_i}(x)) < \alpha_i\}, \end{aligned} \tag{11}$$

where $\mathrm{MBND}_{i-1}(C)$ denotes the boundary region at the $(i-1)$-th stage. At the final stage, if there are emails in the boundary region, we can either use a binary classification [20] or leave the job to the user. The workflow of multistage email spam filtering is explained by Figure 1.

As shown in Figure 1, we would like to make acceptance and rejection decisions at a higher level and proceed into a lower level (i.e., the next three-way decision stage) if more information is required. For each stage, for an object we make an acceptance, a rejection or an non-commitment decision. At the final stage, we would like to use either a binary, two-way decision or leave the job to users.

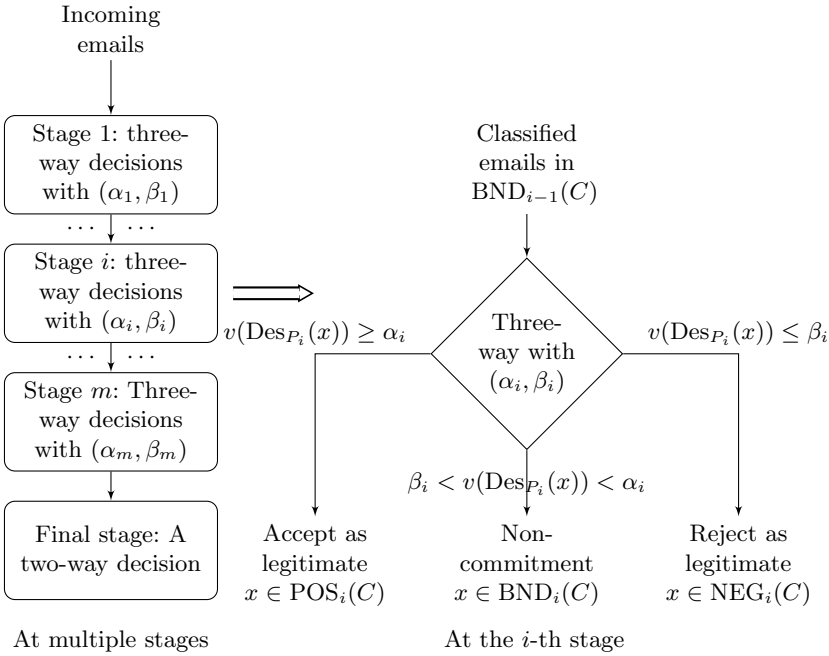


Fig. 1. Workflow of multistage three-way email spam filtering

3.4 A Multistage Three-Way Email Spam Filtering Algorithm

Given a sequence of descriptions of objects, one can make multistage three-way decisions. Since descriptions of objects are defined by using subsets of attributes, one can use a fitness function to select appropriate subsets of attributes. The algorithm in Figure 2 illustrates this approach to multistage three-way decisions in detail.

The algorithm stops and completes the ternary classification when the boundary region is empty, i.e., $\mathrm{MBND}_{i-1}(C) = \emptyset$, or when all available information is used, i.e., $P_{i-1} = AT$. The results of multistage three-way decisions are the three decision regions, i.e., $\mathrm{MPOS}(C)$, $\mathrm{MNEG}(C)$ and $\mathrm{MBND}(C)$. We can either do a binary classification or let the user make the decision when $\mathrm{MBND}(C) \neq \emptyset$.

4 Constructing Multilevel Granular Structures

A fundamental issue of multistage three-way decisions is the construction of the sequence of different levels of representations of the same email. Based on the theory of rough sets and granular computing [13], we formulate and construct such a sequence of multilevel granular structures by adding new information at different levels.

Input: A set of emails U described by a set of attributes AT;
A set of legitimate emails $C \subseteq U$;
A sequence of thresholds $\{(\alpha_i, \beta_i)\}$ satisfying conditions in (9);
A fitness function δ;
Output: MPOS(C), MNEG(C) and MBND(C);
begin
$\text{MPOS}_0(C) = \emptyset, \text{MNEG}_0(C) = \emptyset, \text{MBND}_0(C) = U$;
$i = 1$, $P_0 = \emptyset$;
while $(P_{i-1} \subset AT) \wedge (\text{BND}_{i-1}(C) \neq \emptyset)$ **do**
Use the finiteness function δ to select a subset of attributes P_i;
Produce the description $\text{Des}_{P_i}(x)$ for stage i;
$\text{MPOS}_i(C) = \text{MPOS}_{i-1}(C) \cup \{x \in \text{MBND}_{i-1}(C) \mid v(\text{Des}_{P_i}(x)) \geq \alpha_i\}$;
$\text{MNEG}_i(C) = \text{MNEG}_{i-1}(C) \cup \{x \in \text{MBND}_{i-1}(C) \mid v(\text{Des}_{P_i}(x)) \leq \beta_i\}$;
$\text{MBND}_{i-1}(C) = \{x \in \text{MBND}_{i-1}(C) \mid \beta_i < v(\text{Des}_{P_i}(x)) < \alpha_i\}$;
$i = i + 1$;
end
return $\text{MPOS}(C) = \text{MPOS}_i(C)$, $\text{MNEG}(C) = \text{MNEG}_i(C)$,
$\text{MBND}(C) = \text{MBND}_i(C)$;
end

Fig. 2. Multistage three-way decisions to email spam filtering

4.1 An Information Table

In order to organize and express meaningful information of emails, we adopt the notion of an information table.

Let $S = (U, AT, \{V_a \mid a \in AT\}, \{I_a \mid a \in AT\})$ denote an information table, where U is a finite non-empty set of objects called the universe, AT is a finite non-empty set of attributes, V_a is the domain of attribute $a \in AT$ and $I_a : U \longrightarrow V_a$ is an information function that maps an object $x \in U$ to a particular value $v \in V_a$. We can define a binary relation based on S, called the equivalence relation $E \subseteq U \times U$, which is reflexive ($\forall x \in U, xEx$), symmetric ($\forall x, y \in U, xEy \Longrightarrow yEx$) and transitive ($\forall x, y, z \in U, xEy \wedge yEz \Longrightarrow xEz$). In an information table S, we can define an equivalence relation E_P by using a subset of attributes $P \subseteq AT$,

$$xE_Py \Longleftrightarrow \forall a \in P\ (I_a(x) = I_a(y)). \tag{12}$$

Equivalence classes of E_P containing $x \in U$ is given by $[x]_{E_P} = [x]_P = [x] = \{y \in U \mid xE_Py\}$. Objects in $[x]_P$ share the same description denoted by $\text{Des}_P(x)$. A family of all equivalence classes forms a partition of the universe, called the quotient set, denoted by $U/E_P = U/P = \{[x]_{E_P} \mid x \in U\}$. In order to precisely define the description $\text{Des}_P(x)$, we can define a decision logic based on the information table.

4.2 Describing Objects by a Decision Logic Language

A decision logic language can be recursively defined as follows:

1. Atomic formula: for $a \in AT$, $v \in V_a$, the pair $(a = v)$ is an atomic formula;
2. Composite formula: if ϕ_1 and ϕ_2 are formulas, $\phi_1 \wedge \phi_2$ is a formula,

where a is an attribute in AT, v is a value of a in the domain V_a and $(a = v)$ is an attribute-value pair [5]. In this language, we only discuss the conjunction operator $\wedge$, therefore, an object $x \in U$ can be described as a conjunction of atomic formulas with respect to a subset of attributes $P \subseteq AT$,

$$\mathrm{Des}_P(x) = \bigwedge_{a \in P} (I_a(x) = v), \tag{13}$$

where a is an attribute in the subset P.

Suppose DL is the set of all formulas of the decision logic language. The meaning of a formula $\phi \in DL$ is a subset of objects that can be recursively defined by:

1. Atomic formula: if $\phi = (a = v)$, then $m(a = v) = \{x \in U \mid I_a(x) = v\}$,
2. Composite formula: if $\phi = \phi_1 \wedge \phi_2$, then $m(\phi_1 \wedge \phi_2) = m(\phi_1) \cap m(\phi_2)$.

Grzymala-Busse [5, 6] refers to the meaning of a formula as a block. We can use the pair $(\mathrm{Des}_P(x), m(\mathrm{Des}_P(x)) = [x]_P)$ to describe the equivalence class induced by x [18].

4.3 Formulating Multilevel Granulations

Based on the notions of information table and decision logic language, we can formulate the sequence of multiple representations of the same email. Using the terminology of granular computing, a set of objects can be considered as a granule and a family of granules as a granulation.

A granule g can be defined as a pair [18]:

$$(\mathrm{Des}(g), m(g)), \tag{14}$$

where g is the name of a granule, $\mathrm{Des}(g)$ is the description of g and $m(g)$ is the meaning set of the description. We can form granules based on an information table. For an object $x \in U$, the equivalence class containing x can be defined as a granule:

$$(\mathrm{Des}_P(x), [x]_P). \tag{15}$$

Consider two subsets of attributes $P_1, P_2 \subseteq AT$ with $P_1 \subset P_2 \subseteq AT$. For an object $x \in U$, two granules $(\mathrm{Des}_{P_1}(x), [x]_{P_1})$ and $(\mathrm{Des}_{P_2}(x), [x]_{P_2})$ satisfy the following properties:

$$\begin{aligned} &(\mathrm{C1}) \quad N_a(\mathrm{Des}_{P_2}(x)) \geq N_a(\mathrm{Des}_{P_1}(x)), \\ &(\mathrm{C2}) \quad [x]_{P_2} \subseteq [x]_{P_1}, \end{aligned}$$

where $N_a(\cdot)$ denotes the number of attributes used in a description. Intuitively, values on additional attributes $P_2 - P_1$ can be viewed as new evidence or support

information [21]. For example, in medical diagnosis, $P_2 - P_1$ represents a new set of medical tests. Property (C1) shows that we change a coarser description $\mathrm{Des}_{P_1}(x)$ into a finer description $\mathrm{Des}_{P_2}(x)$ by adding new pieces of information:

$$\mathrm{Des}_{P_2-P_1}(x) = \bigwedge_{a \in P_2-P_1} (a = I_a(x)). \tag{16}$$

It is obvious that $\mathrm{Des}_{P_1}(x) = \mathrm{Des}_{P_1}(x) \wedge \mathrm{Des}_{P_2-P_1}(x)$. In terms of meaning sets, more details or support information results in a finer granule, i.e., $[x]_{P_2} \subseteq [x]_{P_1}$, as shown by Property (C2).

A family of granules forms a granulation which can be defined by:

$$G = \{g_1, g_2, \cdots, g_k\}, \tag{17}$$

where g_i with $1 \leq i \leq k$ is a granule in G. For two granulations G_1 and G_2, a refinement-coarsening relation $\preceq$ can be established by:

$$G_1 \preceq G_2 \Longleftrightarrow \forall g_i \in G_1 \exists g_j \in G_2 \ (m(g_i) \subseteq m(g_j)). \tag{18}$$

That is, for each granule g_i in G_1, if we can find a granule g_j in G_2 such that $m(g_i) \subseteq m(g_j)$, then the granulation G_1 is finer than G_2 and G_2 is coarser than G_1. A quotient set U/P defined using an information table is a typical example of granulations, which can be re-expressed in terms of granules:

$$U/P = \{(\mathrm{Des}_P(x), [x]_P) \mid x \in U\}. \tag{19}$$

For two subsets of attributes $P_1 \subset P_2 \subseteq AT$, we have the following property: if $P_1 \subset P_2 \subseteq AT$,

$$\text{(C3)} \quad U/P_2 \preceq U/P_1.$$

This can be easily verified by the definition of refinement-coarsening relation.

According to Property (C3), we can construct a sequence of different levels of granulations by using a nested sequence of subsets of attributes:

$$U/P_m \preceq U/P_{m-1} \preceq \ldots \preceq U/P_2 \preceq U/P_1. \tag{20}$$

A finer granulation contains smaller granules with more detailed descriptions, and vice-versa. For multistage three-way decisions, we use a coarser granulation at a higher level of decision stage, while use a finer granulation at a lower level.

5 Conclusion

An analysis of existing single-stage two-way and three-way decision approaches to email spam filtering points out the trade-off between correct and incorrect classifications in two-way decisions and the problem of three-way decisions with insufficient information. In order to solve these issues, we introduce a multistage three-way decision model to email spam filtering by extending the framework of sequential three-way decisions, and propose an approach to construct multilevel

granular structure to represent the same email at different levels of granularity. We have performed some preliminary experiments by using a small data set and the results were very encouraging. As future work, we will focus on the interpretation and construction of the sequence of pairs of thresholds for multistage three-way decision-makings and fully evaluate our model by using more data sets.

Acknowledgements. This work is partially supported by a Discovery Grant from NSERC Canada.

References

1. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C.D., Stamatopoulos, P.: Learning to filter spam e-mail: A comparison of a naive Bayesian and a memory-based approach. In: Proceedings of PKDD 2000, pp. 1–13 (2000)
2. Azam, N., Yao, J.T.: Multiple criteria decision analysis with game-theoretic rough sets. In: Li, T., Nguyen, H.S., Wang, G., Grzymala-Busse, J.W., Janicki, R., Hassanien, A.E., Yu, H. (eds.) RSKT 2012. LNCS (LNAI), vol. 7414, pp. 399–408. Springer, Heidelberg (2012)
3. Cristianini, N., Shawe-Taylor, I.: An Introduction to Support Vector Machines and Other Kernel-base Learning Methods. Cambridge University Press, Cambridge (2000)
4. Deng, X.F., Yao, Y.Y.: A multifaceted analysis of probabilistic three-way decisions (manuscript, 2013)
5. Grzymala-Busse, J.W.: LERS - A system for learning from examples based on rough sets. In: Słowiński, R. (ed.) Intelligent Decision Support, pp. 3–18. Kluwer Academic Publishers, Boston (1992)
6. Grzymala-Busse, J.W.: A local version of the MLEM2 algorithm for rule induction. Fundamenta Informaticae 100, 99–116 (2010)
7. Jia, X.Y., Li, W.W., Shang, L., Chen, J.J.: An optimization viewpoint of decision-theoretic rough set model. In: Yao, J.T., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS (LNAI), vol. 6954, pp. 457–465. Springer, Heidelberg (2011)
8. Jia, X.Y., Zheng, K., Li, W.W., Liu, T.T., Shang, L.: Three-way decisions solution to filter spam email: An empirical study. In: Yao, J.T., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS (LNAI), vol. 7413, pp. 287–296. Springer, Heidelberg (2012)
9. Li, H.X., Zhou, X.Z.: Risk decision making based on decision-theoretic rough set: A three-way view decision model. International Journal of Computational Intelligence Systems 4, 1–11 (2011)
10. Liu, D., Li, T.R., Li, H.X.: A multiple-category classification approach with decision-theoretic rough sets. Fundamenta Informaticae 115, 173–188 (2012)
11. Liu, D., Li, T.R., Liang, D.C.: A three-way government decision analysis with decision-theoretic rough sets. International Journal of Uncertainty and Knowledge-based Systems 20, 119–132 (2012)
12. Pantel, P., Lin, D.K.: SpamCop: A spam classification & organization program. In: AAAI Workshop on Learning for Text Categorization. AAAI Technical Report WS-98-05, pp. 95–98 (1998)

13. Pedrycz, W.: Granular Computing: Analysis and Design of Intelligent Systems. CRC Press/Francis Taylor, Boca Raton (2013)
14. Robinson, G.: A statistical approach to the spam problem, spam detection. Linux Journal (107) (2003), http://www.linuxjournal.com/article/6467 (retrieved on April 25, 2013)
15. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian approach to filtering junk e-mail. In: AAAI workshop on learning for text categorization. AAAI Technical Report WS-98-05, Madison, Wisconsin (1998)
16. Schapire, E., Singer, Y.: BoosTexter: A boosting-based system for text categorization. Machine Learning 39, 135–168 (2000)
17. Yao, Y.Y.: Probabilistic rough set approximations. International Journal of Approximate Reasoning 49, 255–271 (2008)
18. Yao, Y.Y.: Information granulation and rough set approximation. International Journal of Intelligent Systems 16, 87–104 (2001)
19. Yao, Y.Y.: An outline of a theory of three-way decisions. In: Yao, J.T., Yang, Y., Słowiński, R., Greco, S., Li, H.X., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 1–17. Springer, Heidelberg (2012)
20. Yao, Y.Y.: Granular computing and sequential three-way decisions. In: Lingras, P., Wolski, M., Cornelis, C., Mitra, S., Wasilewski, P. (eds.) RSKT 2013. LNCS (LNAI), vol. 8171, pp. 16–27. Springer, Heidelberg (2013)
21. Yao, Y.Y., Deng, X.F.: Sequential three-way decisions with probabilistic rough sets. In: Proceedings of the 10th IEEE International Conference on Cognitive Informatics and Cognitive Computing, pp. 120–125 (2011)
22. Yih, W.T., McCann, R., Kolcz, A.: Improving spam filtering by detecting gray mail. In: Proceedings of the 4th Conference on Email and Anti-Spam, CEAS 2007 (2007)
23. Yu, H., Chu, S.S., Yang, D.C.: Autonomous knowledge-oriented clustering using decision-theoretic rough set theory. Fundamenta Informaticae 115, 141–156 (2012)
24. Zhao, W., Zhang, Z.: An email classification model based on rough set theory. In: Proceedings of the International Conference on Active Media Technology, pp. 403–408 (2005)
25. Zhou, B., Yao, Y.Y., Luo, J.G.: Cost-sensitive three-way email spam filtering. Journal of Intelligent Information Systems (2013), doi:10.1007/s10844-013-0254-7

Cost-Sensitive Three-Way Decision: A Sequential Strategy

Huaxiong Li[1], Xianzhong Zhou[1], Bing Huang[2], and Dun Liu[3]

[1] School of Management and Engineering, Nanjing University, Nanjing, Jiangsu, 210093, P.R. China
[2] School of Information Science, Nanjing Audit University, Nanjing, Jiangsu, 211815, P.R. China
[3] School of Economics and Management, Southwest Jiaotong University, Chengdu, Sichuan, 610031, P.R. China
{huaxiongli,zhouxz}@nju.edu.cn, hbhuangbing@126.com, newton83@163.com

Abstract. Three-way decision model is an extension of two-way decision model, in which boundary region decision is regarded as a new feasible decision choice when precise decision can not be immediately made due to lack of available information. In this paper, a cost-sensitive sequential three-way decision model is presented, which simulate a gradual decision process from rough granule to precise granule. At the beginning of the sequential decision process, the decision results have a high decision cost and many instances are decided as boundary region due to lack of information. With the increasing of the decision steps, the decision cost decrease and more instances are precisely decided. Eventually the decision cost achieve at a satisfying value and the boundary region disappears. The paper presents both theoretic analysis and experimental validation on this proposed model.

Keywords: three-way decision, cost-sensitive, sequential decision, decision-theoretic rough sets.

1 Introduction

Three-way decision theory, proposed by Yao in [19], is an extension decision theory of two-way decision theory, in which the positive decision, negative decision and boundary decision are considered as three optional actions in the process of decision [18,20]. In traditional two-way decision theory, there are basically two choices for the decision: one is positive decision, and the other is negative decision, which requires the decision makers to make immediately decision actions. The two-way decision strategy many result in wrong decisions when the information used for decisions is limited while the decision result should be immediately made. In this situation, it is a reasonable choice to take three-way decision strategy [18]. The main superiority of three-way decision compared to two-way decision is the utility of the boundary decision. In three-way decision

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 325–337, 2013.

theory, the boundary decision is regarded as a feasible choice of decision when the available information for decision is too limited to make a proper decision, which is similar to the human decision strategy in the practical decision problems.

Three-way decision theory originates from the researches on decision-theoretic rough set model (DTRS) [17,22,23], which presents a semantics explanation on how to classify an instance into positive, negative and boundary region based on cost-sensitive classification strategy. In recent years, three-way decision theory and DTRS have received more and more attention, and many examples of theoretical research and applications of the three-way decision DTRS are frequently mentioned in literatures [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,24,25]. Previous three-way decision and DTRS researches mainly focus on a static minimum risk decision result induced from a certain static known information. For example, we can make three-way decision based on the entire attribute set of a decision table, and we can also make three-way decision based on a reduced test attribute set, i.e., reduct of the attribute set. The three-way decision is determined when the test attribute set is given. However, in real world decision problem, the available information for decision is always limited, and there will be some costs in the process of acquiring the attribute values. In this case, we may take a sequential decision strategy: the three-way decisions are sequentially made according to gradually acquired information which takes some certain test cost, until the decision results are satisfied. Recently, Yao. et al. proposes a framework of sequential three-way decisions with probabilistic rough sets [21]. In this paper, we will further discuss this kind of sequential three-way decision strategy in detail and present a new cost-sensitive sequential three-way decision model.

2 Cost-Sensitive Three-Way Decision

In this section, we will review some basic notions of cost-sensitive three-way decision [17,18,19,20,22,23], which forms a theoretical basis for the proposed cost-sensitive sequential three-way decision model.

Let us consider on a binary decision or classification problem. The set of actual states is given by $\Omega = \{X, \neg X\} = \{X_P, X_N\}$ indicating that the actual state of each instance for decision is either labeled by X (X_P) or labeled by $\neg X$ (X_N). If we take two-way decision strategy, the decision actions include only two choices: deciding X or $\neg X$. Considering a dilemma situation when the available information to make a precise decision is limited, we should add a third choice for decision, i.e., delay decision, which means we need to collect more information for further precise decision. The three-way decision presents such a dilemma decision action. In three-way decision model, decision actions are given by $\mathcal{A} = \{a_P, a_N, a_B\}$, representing $POS(X)$, $NEG(X)$ and $BND(X)$ decisions respectively. Table 1 presents all costs for three-way decisions. The cost λ_{ij} forms a matrix denoted as $(\lambda_{ij})_{2\times 3}$, where $i \in \{P, B, N\}$, and $j \in \{P, N\}$.

Normally, the costs of right decision are less than that of wrong decision, and we have $\lambda_{PP} \leq \lambda_{BP} \leq \lambda_{NP}$ and $\lambda_{NN} \leq \lambda_{BN} \leq \lambda_{PN}$. Moreover, a reasonable assumption is that the costs of right decision are equal to zero, then we get a simplified decision cost matrix with only four parameters including $\lambda_{PN}, \lambda_{NP}, \lambda_{BP}$

Table 1. Decision cost matrix

Actual States	Decide $POS(X)$	Decide $BND(X)$	Decide $NEG(X)$
X (X_P)	λ_{PP}	λ_{BP}	λ_{NP}
$\neg X$ (X_N)	λ_{PN}	λ_{BN}	λ_{NN}

and λ_{BN}. By introducing the cost-sensitive classification and learning methods, we compare all decision costs of $\mathcal{A} = \{a_P, a_N, a_B\}$ and select out the optimal action which has the minimum expected decision cost. We denote the data set as a decision information table [23]: $S = (U, At = C \cup D, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$. Given a set of attributes $\tilde{C} \subseteq C$, the expected decision cost $R(a_i|[x]_{\tilde{C}})$ for taking the each action can be expressed as follows:

$$\begin{aligned} R(a_P|[x]_{\tilde{C}}) &= \sum_{j\in\{P,N\}} \lambda_{Pj} P(X_j|[x]_{\tilde{C}}) = \lambda_{PN} P(X_N|[x]_{\tilde{C}}), \\ R(a_N|[x]_{\tilde{C}}) &= \sum_{j\in\{P,N\}} \lambda_{Nj} P(X_j|[x]_{\tilde{C}}) = \lambda_{NP} P(X_P|[x]_{\tilde{C}}), \\ R(a_B|[x]_{\tilde{C}}) &= \sum_{j\in\{P,N\}} \lambda_{Bj} P(X_j|[x]_{\tilde{C}}) = \lambda_{BP} P(X_P|[x]_{\tilde{C}}) + \lambda_{BN} P(X_N|[x]_{\tilde{C}}), \end{aligned} \tag{1}$$

where $[x]_{\tilde{C}}$ denotes the equivalence class of x under relation $\tilde{C}$. Based on cost-sensitive classification strategy, we compute all three decision cost $R(a_P|[x]_{\tilde{C}})$, $R(a_N|[x]_{\tilde{C}})$ and $R(a_B|[x]_{\tilde{C}})$ to find out the minimum decision cost, then the optimal three-way decision $\phi^*([x]_{\tilde{C}})$ will be made, which is presented as follows:

$$\phi^*([x]_{\tilde{C}}) = \underset{\mathcal{D}\in\{a_P,a_N,a_B\}}{\operatorname{argmin}} R(\mathcal{D}|[x]_{\tilde{C}}). \tag{2}$$

Based on the properties of DTRS, the optimal three-way decision results can be enumerated as formula (3) under the condition that $\lambda_{PP} \leq \lambda_{BP} \leq \lambda_{NP}$, $\lambda_{NN} \leq \lambda_{BN} \leq \lambda_{PN}$ and $(\lambda_{PN} - \lambda_{BN})(\lambda_{NP} - \lambda_{BP}) > (\lambda_{BP} - \lambda_{PP})(\lambda_{BN} - \lambda_{NN})$ [17]:

$$\phi^*([x]_B) = \begin{cases} a_P, & , \quad \text{if } P(X|[x]_{\tilde{C}}) \geq \alpha, \\ a_N, & , \quad \text{if } P(X|[x]_{\tilde{C}}) \leq \beta, \\ a_B, & , \quad \text{if } \beta < P(X|[x]_{\tilde{C}}) < \alpha, \end{cases} \tag{3}$$

where

$$\begin{aligned} \alpha &= \frac{\lambda_{PN} - \lambda_{BN}}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}, \\ \beta &= \frac{\lambda_{BN} - \lambda_{NN}}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}, \end{aligned} \tag{4}$$

are thresholds determined by cost matrix $(\lambda_{ij})_{2\times 3}$.

3 Sequential Three-Way Decision Model

In real world decision problems, the available information used for decision is usually limited, and there will be some costs when acquiring the available information. For example in medical diagnose decision problem, it costs money to make physical examinations including X-rays tests, gastroscopy test, and magnetic resonance imaging test. Some tests may be very expensive, therefore, doctors will sequentially make the test. Initially, some cheaper tests will be taken and the diagnose is made based on these available information. The final decision will be made if currently collected information is sufficient for precise decision. However, if currently collected information is too limited to make a precise decision, then the doctor may delay the decision, and take next step test to collect more information for further decision. Such decision strategies are frequently used in human decision process, which form a sequential three-way decision model. In this section, we will discuss this decision model in detail.

Definition 1. *Let* $S = (U, At = C \cup D, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$ *be a decision table, where* U *denotes the set of objects, and* At *is a set of attributes including a condition attributes set* C *and a decision attributes set* D. V_a *is a set of values of* $a \in At$, *and* $I_a : U \to V_a$ *is an information function.* x *is an instance for decision.* $M = \{\lambda_{ij}\}_{3\times 2}$ $(i \in \{P, N, B\}, j \in \{P, N\})$ *is a decision cost matrix, and* $|C| = m$. *A sequential three-way decision series is defined as:*

$$\begin{aligned} &SD = (SD_1, SD_2, SD_3, \cdots, SD_m) \\ &= (\phi^*([x]_{\{c_{i_1}\}}), \phi^*([x]_{\{c_{i_1}, c_{i_2}\}}), \phi^*([x]_{\{c_{i_1}, c_{i_2}, c_{i_3}\}}), \cdots, \phi^*([x]_{\{c_{i_1}, c_{i_2}, \cdots, c_{i_m}\}}), \end{aligned} \tag{5}$$

where $\phi^*([x]_{\{c_{i_1}, c_{i_2}, \cdots, c_{i_k}\}})$ *is the optimal three-way decision presented in formula (2), i.e.,* $\phi^*([x]_{\{c_{i_1}, c_{i_2}, \cdots, c_{i_k}\}}) = arg\ min_{\mathcal{D} \in \{a_P, a_N, a_B\}} R(\mathcal{D}|[x]_{\{c_{i_1}, c_{i_2}, \cdots, c_{i_k}\}})$. *The attributes series for* SD *is denoted as:* $SA_m = (c_{i_1}, c_{i_2}, \cdots, c_{i_m})$.

Considering a single k-th step decision SD_k in SD, we compute the decision cost of SD_k based on formula (2), then we have:

$$Cost(x, SD_k) = \min_{i \in \{P,N,B\}} \Big(\sum_{j \in \{P,N\}} \lambda_{iP} P(X_j | [x]_{\{c_{i_1}, c_{i_2}, \cdots, c_{i_k}\}}) \Big). \tag{6}$$

With the increasing of the decision steps, one may intuitively conclude that the decision cost decreases since the decision precision are gradually improved. Such conclusion is correct in most cases but it is not always true. We present two theorems for explanation.

Theorem 1. *Let* $SD = (SD_1, SD_2, SD_3, \cdots, SD_m)$ *be a sequential three-way decision series presented in Definition 1, and* SD_k *be the* k*-th decision series* $(1 \le k \le m)$. *Suppose* SD_l *is a successive decision of* SD_k, *i.e.,* $l > k$, *and* $P(X_P|[x]_{C_k}) \ge \alpha$, *where* α *is determined by formula (4), then the following proposition holds:*

If $P(X_P|[x]_{C_l}) \ge P(X_P|[x]_{C_k})$, *then* $Cost(x, SD_l) \le Cost(x, SD_k)$;
If $\alpha \le P(X_P|[x]_{C_l}) < P(X_P|[x]_{C_k})$, *then* $Cost(x, SD_l) \ge Cost(x, SD_k)$.

Proof: Suppose the attribute set associated with SD_k is $\{c_{i_1}, c_{i_2}, \cdots, c_{i_k}\}$, then we have $SD_k = \phi^*([x]_{\{c_{i_1}, c_{i_2}, \cdots, c_{i_k}\}})$. According to formula (6), we compute the $Cost(x, SD_k)$ as follows:

$$Cost(x, SD_k) = \min_{i \in \{P,N,B\}} \Big(\sum_{j \in \{P,N\}} \lambda_{iP} P(X_j | [x]_{\{c_{i_1}, c_{i_2}, \cdots, c_{i_k}\}}) \Big)$$

$$= \begin{cases} \lambda_{PN} P(X_N | [x]_{C_k}), & , \quad \text{if } P(X|[x]_{C_k}) \geq \alpha, \\ \lambda_{NP} P(X_P | [x]_{C_k}), & , \quad \text{if } P(X|[x]_{C_k}) \leq \beta, \\ \lambda_{BP} P(X_P | [x]_{C_k}) + \lambda_{BN} P(X_N | [x]_{C_k}), & , \quad \text{if } \beta < P(X|[x]_{C_k}) < \alpha, \end{cases} \tag{7}$$

Firstly, if $P(X_P|[x]_{C_l}) \geq P(X_P|[x]_{C_k})$, then we have $P(X_P|[x]_{C_l}) \geq P(X_P|[x]_{C_k}) \geq \alpha$. According to formula (7), $Cost(x, SD_k) = \lambda_{PN} P(X_N|[x]_{C_k}) = \lambda_{PN} - \lambda_{PN} P(X_P|[x]_{C_k})$ and $Cost(x, SD_l) = \lambda_{PN} P(X_N|[x]_{C_l}) = \lambda_{PN}(1 - P(X_P|[x]_{C_l})) = \lambda_{PN} - \lambda_{PN} P(X_P|[x]_{C_l}) \leq \lambda_{PN} - \lambda_{PN} P(X_P|[x]_{C_k}) = Cost(x, SD_k)$, thus $Cost(x, SD_l) \leq Cost(x, SD_k)$. Secondly, if $\alpha \leq P(X_P|[x]_{C_l}) < P(X_P|[x]_{C_k})$, then according to formula (3) and (7) we have $\phi^*([x]_{C_l}) = a_P$, and $Cost(x, SD_l) = \lambda_{PN} P(X_N|[x]_{C_l}) = \lambda_{PN} - \lambda_{PN} P(X|[x]_{C_l}) \geq \lambda_{PN} - \lambda_{PN} P(X|[x]_{C_k}) = Cost(x, SD_k)$, thus $Cost(x, SD_l) \geq Cost(x, SD_k)$. □

Remark: It can be similarly proved that the decision cost may increase or decrease with the increase of the decision steps under the condition $\beta < P(X_P|[x]_{C_k}) < \alpha$ or the condition $P(X_P|[x]_{C_k}) \leq \beta$. Moreover, according to literatures [6] and [8], $P(X_P|[x]_{C_l}) > P(X_P|[x]_{C_k})$ and $P(X_P|[x]_{C_l}) < P(X_P|[x]_{C_k})$ are two possible cases when $l > k$. It implies that the decision cost is non-monotonic with regard to the decision step, which is inconsistent with our intuitions. However, if we take a global view, we may find that the global trend of the decision cost will decrease with the increasing of the decision steps.

Definition 2. *Let $S = (U, At = C \cup D, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$ be a decision table. S is called a consistent decision table if and only if $POS_C(D) = U$, where $POS_C(D) = \bigcup_{X \in U/D} \underline{apr}_C(X)$, and $\underline{apr}_C(X) = \{x | [x]_C \subseteq X\}$.*

In general, decision consistency assumption is mostly true in real decision problem. For example in medical diagnose problems, patients who have all same symptoms should be diagnosed as the same illness, otherwise we may think that the medical examination is wrong or the related medical data is not sufficient. Under the decision consistency assumption, we have following Theorem 2.

Theorem 2. *Let $S = (U, At = C \cup D, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$ be a consistent decision table, $x \in U$ is an instance for decision, and $SD = (SD_1, SD_2, SD_3, \cdots, SD_m)(m = |C|)$ be a sequential three-way decision series presented in Definition 1. It concludes that there exist a step $k(1 \leq k \leq m)$ satisfying $Cost(x, SD_k) = 0$.*

Proof: S is a consistent decision table, therefore $POS_C(D) = U$. For $x \in U$, we have $[x]_C \subseteq X_P$ or $[x]_C \subseteq X_N$ since $U/D = X_P, X_N$. If $[x]_C \subseteq X_P$, then for any $0 \leq \alpha \leq 1$, $P(X_P|[x]_C) = 1 \geq \alpha$. According to formula (7),

$Cost(x, SD_m) = \lambda_{PN} - \lambda_{PN}P(X_P|[x]_C)$=0, therefore we have the worse case that $Cost(x, SD_k) = 0$ holds when k reach the maximum value m. Similarly, if $[x]_C \subseteq X_N$, then for any $0 \leq \beta \leq 1$, $P(X_P|[x]_C) = 0 \leq \beta$, According to formula (7), $Cost(x, SD_m) = \lambda_{PN} - \lambda_{PN}P(X_P|[x]_C)$=0, therefore we have the worse case that $Cost(x, SD_k) = 0$ holds when k reach the maximum value m. Thus in both cases $[x]_C \subseteq X_P$ and $[x]_C \subseteq X_N$, we can find the worse case of $k = m$ satisfying $Cost(x, SD_k) = 0$. Normally in practice, we may usually find that there exist a $k < m$ decision step which reduces the decision cost to zero. The reason is that $[x]_{\{c_{i_1}\}} \supseteq [x]_{\{c_{i_1},c_{i_2}\}} \supseteq [x]_{\{c_{i_1},c_{i_2},c_{i_3}\}} \supseteq \cdots \supseteq [x]_{\{c_{i_1},c_{i_2},\cdots,c_{i_k}\}}$. With the increasing of decision steps, the equivalence class of x reduces and it has higher possibility to be precisely decided as X_P or X_N, and the decision cost will reduce to zero when the instance x is precisely decided. □

Remark: Theorem 1 and Theorem 2 present two views on the trend of decision cost when decision steps increase. Theorem 1 takes a local view: the decision cost may locally increase even the decision steps increase, while Theorem 2 takes a global view: the decision cost will eventually reduce to zero regardless of the local changes of the decision cost in the series decision process.

Another property should be concerned on cost-sensitive sequential three-way decision model is the variation trend of the boundary region. The following Theorem 3 presents a theoretic result on this issue.

Theorem 3. *Let* $S = (U, At = C \cup D, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$ *be a consistent decision table,* $U/D = \{X_P, X_N\}$*, and* $x \in U$ *is an instance for decision.* $SD = (SD_1, SD_2, SD_3, \cdots, SD_m) = (\phi^*([x]_{\{c_{i_1}\}}), \phi^*([x]_{\{c_{i_1},c_{i_2}\}}), \cdots, \phi^*([x]_{\{c_{i_1},c_{i_2},\cdots,c_{i_m}\}})(m = |C|)$ *is a sequential three-way decision series presented in Definition 1. For a single decision step* $1 \leq k \leq m$*, the positive region, negative region and boundary region with regard to a pair of threshold* (α, β) *are denoted as* $POS^{\alpha}_{\{c_{i_1},c_{i_2},\cdots,c_{i_k}\}}(X_P)$*,* $NEG^{\beta}_{\{c_{i_1},c_{i_2},\cdots,c_{i_k}\}}(X_P)$ *and* $BND^{(\alpha,\beta)}_{\{c_{i_1},c_{i_2},\cdots,c_{i_k}\}}(X_P)$ *respectively. It concludes that there exist a step* $k(1 \leq k \leq m)$ *satisfying* $BND^{(\alpha,\beta)}_{\{c_{i_1},c_{i_2},\cdots,c_{i_k}\}}(X_P) = \emptyset$.

Proof: It can be proved that the boundary region will reduce to empty set when the decision step k reach m at most. In the worse case $k = m$, according to the decision consistent assumption, $POS_C(D) = U$. For any $x \in U$, we have $[x]_C \subseteq X_P$ or $[x]_C \subseteq X_N$ since $U/D = X_P, X_N$. If $[x]_C \subseteq X_P$, then $P(X_P|[x]_C) = 1 \geq \alpha$, i.e., x is decided as positive region. Otherwise, $[x]_C \subseteq X_N$, then $P(X_N|[x]_C) = 1$, $P(X_P|[x]_C) = 1 - P(X_N|[x]_C) = 0 \leq \beta$, i.e., x is decided as in negative region. Therefore, for any $x \in U$, in the worse case $k = m$, it is decided as either in positive region or in negative region, and it will not be decided as boundary region, i.e., for any $x \in U$, $x \notin BND^{(\alpha,\beta)}_{\{c_{i_1},c_{i_2},\cdots,c_{i_k}\}}(X_P)$, thus $BND^{(\alpha,\beta)}_{\{c_{i_1},c_{i_2},\cdots,c_{i_k}\}}(X_P) = \emptyset$. □

According to Theorems 1 to 3, we may conclude that the global variation trend of both decision cost and boundary region are decreased in the sequential decision process. Therefore, a series of decisions with more steps will have lower decision cost and more accurate decision results. However, there will be some test

costs to acquire the unknown attribute values in the sequential decision process. The more steps will lead to a higher test cost. Therefore, we should balance the decision cost and test cost. A feasible choice is to set a upper bound of decision cost in the sequential decision process, and gradually increase decision steps until the decision cost are under the designated upper bound of decision cost. First, we introduce how to set the test costs of the attributes for a decision table.

In real world database, some data sets are naturally associated with test costs on attributes. However, many data sets have not been naturally associated with test costs. In this case, we assume that the test cost of an attribute is proportional to the classification ability of the attribute. For example, nuclear magnetic resonance test has higher diagnose efficiency than X-rays test, therefore, the former has a higher test cost than the latter. An appropriate measure for evaluate the classification ability of an attribute is the conditional entropy. We present a conditional-entropy-based method to evaluate the test cost.

Definition 3. *Let* $S = (U, At = C \cup D, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$ *be a decision table. For a single attribute* $c_l \in C$*, denote* $U/\{c_l\} = \{X_1, X_2, \cdots, X_{\tilde{N}}\}$*,* $U/D = \{Y_1, Y_2, \cdots, Y_{\tilde{M}}\}$*, then the test cost of* c_l *is defined as:* $T(c_l) = 1 - H(D|\{c_l\}) = 1 + \sum_{i=1}^{\tilde{N}} P(X_i) \sum_{i=1}^{\tilde{M}} P(Y_j|X_i) \log P(Y_j|X_i)/\log(\tilde{M})$*. For a l-step sequential decision* SD_l*, the attributes series set is denoted as* $SA_l = (c_{i_1}, c_{i_2}, \cdots, c_{i_l})$*, and the test cost of* SA_l *is defined as the summation of the test costs of* c_{i_j}*:* $Test(SA_l) = \sum_{j=1}^{l} T(c_{i_j})$.

Remark: The conditional entropy $H(D|\{c_l\})$ represents the correlation degree between $\{c_l\}$ and D. If $H(D|\{c_l\})$ equals to a lower value, then $\{c_l\}$ has a higher correlation with D, which indicates that $\{c_l\}$ has a higher ability for classification w.r.t. D. It can be proved that $0 \leq H(D|\{c_l\}) \leq 1$, then we take $1-H(D|\{c_l\})$ as the test cost of a single attribute c_l. The test cost of an attribute set is defined as the summation of all test costs of attributes in the set.

Based on the Theorems 1-3 and the definition of the test cost, we present a cost-sensitive sequential three-way decision algorithm to simulate the human sequential decision process: a sequential decision process from rough granule to precise granule. In the beginning, a part of attributes are used, which leads to a high decision cost and a rough granule decision, and many instances are decided as boundary region. Then some new attributes are selected for decision according to some certain order and strategy, so that the available information increase and the decision cost decrease. Besides, some instances previously decided as boundary are transferred to positive or negative region, thus the rough granule decision result transfers to a precise decision result. The Cost-Sensitive Sequential Decision (CSSD) algorithm is presented in Figure 1.

In the proposed algorithm, the order to add attributes includes two strategies. One is the Test Cost ascend strategy (TCA), and the other is the Test Cost Descend strategy (TCD). The former adds the attributes from low test cost to high test cost, and the latter adds the attributes from high test cost to low cost.

ALGORITHM: Cost-Sensitive Sequential Decision (TCA/TCD)
INPUT: A consistent decision table $S = (U, C \cup D\}$, a decision cost matrix $M = \{\lambda_{ij}\}$, an instance x for classification, a decision cost threshold $Cost^*$;
OUTPUT: A sequential decision $SD = (SD_1, SD_2, \cdots, SD_k)$,
A sequential decision attribute series $SA_k = (c_{i_1}, c_{i_2}, \cdots, c_{i_k})$.
Decision cost $Cost$; Test cost $Tcost$.
PROCESS:
Compute the partition of U by D: $U/D = \{Y_1, Y_2, \cdots, Y_{\tilde{M}}\}$;
For any $c_l \in C$
$U/\{c_l\} = \{X_1, X_2, \cdots, X_{\tilde{N}}\}$;
$T(c_l) = 1 + \sum_{i=1}^{\tilde{N}} P(X_i) \sum_{i=1}^{\tilde{M}} P(Y_j|X_i) \log P(Y_j|X_i) / \log(\tilde{M})$;
End of For
$C' \leftarrow$Sorted $c_l \in C$ in ascending(TCA)/descending(TCD) order of $T(c_l)$;
Set decision attribute series $SA \leftarrow \emptyset$, Set candidate attribute set $B \leftarrow C'$;
$k = 1$; $b \leftarrow B_1$ (the first attribute in B); $SA \leftarrow (b)$;
$SD_k \leftarrow \phi^*([x]_b) = \text{argmin}_{\mathcal{D} \in \{a_P, a_N, a_B\}} R(\mathcal{D}|[x]_b)$;
$Cost(x, SD_k) = \min_{i \in \{P,N,B\}} (\sum_{j \in \{P,N\}} \lambda_{iP} P(X_j|[x]_{SA})$;
$Cost \leftarrow Cost(x, SD_k)$;$Tcost \leftarrow T(b)$;
While $Cost > Cost^*$ **Do**
$k \leftarrow k + 1$; $b \leftarrow B_1$; $SA \leftarrow SA \cup \{b\}$; $B \leftarrow B - \{b\}$;
$SD_k \leftarrow \phi^*([x]_{SA}) = \text{argmin}_{\mathcal{D} \in \{a_P, a_N, a_B\}} R(\mathcal{D}|[x]_{SA})$;
$Cost(x, SD_k) = \min_{i \in \{P,N,B\}} (\sum_{j \in \{P,N\}} \lambda_{iP} P(X_j|[x]_{SA})$;
Update $Cost \leftarrow Cost(x, SD_k)$; $Tcost \leftarrow Tcost + T(b)$;
End of While
$SD \leftarrow (SD_1, SD_2, \cdots, SD_k)$; $SA_k \leftarrow (c_{i_1}, c_{i_2}, \cdots, c_{i_k})$;
OUTPUT: SD; SA_k; $Cost$; $Tcost$.

Fig. 1. The Cost-Sensitive Sequential Decision Algorithm (CSSD)

These two strategies are consistent with human decision process in reality. For example in medical diagnose problem, some patients prefer low test cost rather than low decision cost due to lack of money. Initially, they may take some low cost tests for diagnoses decision. The tests will terminated if the available information is sufficient for diagnoses. Otherwise, they may concern some more low cost tests. They do not select high cost tests unless all low cost tests information are not sufficient for a reliable diagnoses decision. Therefore, they take the Test Cost Ascend strategy. On the other hand, some patients prefer low decision cost rather than low test cost. They prefer tests that may support a precise diagnoses, regardless of test cost. Therefore, they take the Test Cost Descend strategy.

4 Experimental Analysis

In this section, we presents an experimental analysis on the proposed cost-sensitive sequential three-way decision strategy. Experiments are performed on four UCI data sets listed in Table 2. In the four data sets, Mushroom, Breast-cancer-wisconsin, and Hepatitis contain missing values, and we delete those instances with missing values in data sets Mushroom, Breast-cancer-wisconsin, and

fill in missing values with most common values for data set Hepatitis. Considering that the proposed algorithm is designed to deal with binary classification problem, we convert the four classes in the data set Car to two classes by merging the class labeled with "good" and "vgood" into the class labeled with "acc". Among the four data sets, there are three decision consistent data (Mushroom, Breast and Car) and one decision inconsistent data (Hepatitis). The decision cost matrix is set as: $\lambda_{PN} = 16, \lambda_{NP} = 20, \lambda_{BP} = 5$, and $\lambda_{BN} = 4$.

Table 2. Experimental data sets from UCI machine learning repository

ID	Data	Classes	Attributes	Raw size	New size
1	Mushroom	2	22	8124	5644
2	Breast-cancer-wisconsin	2	9	699	683
3	Hepatitis	2	12	155	155
4	Car	2	6	1728	1728

Firstly, we test the variation trends of decision cost based on the proposed algorithm CSSD, which are analyzed in Theorem 1 and Theorem 2. Based on the CSSD, we compute the average decision costs of two sets of instances with regard to the decision steps k. One set is the entire set of instances, and the other is a part of instances from each data set (we take the first 5 instances here). The former is used to validate the global trend of decision cost (described in Theorem 2), and the latter is used to validate the local trend of decision cost (described in Theorem 1). The order to add attributes includes both TCA and TCD strategies. The experiment results are presented in Fig. 2 to Fig. 9, where Fig. 2,4,6, and 8 take the TCA strategy, and Fig. 3,5,7, and 9 take the TCD strategy. From Fig. 2-9, we can obtain the following conclusions: (1) The decision cost is non-monotonic with regard to the decision steps. Decision cost may not decrease when decision steps increase. In some situations, decision cost will abnormally rise even the decision steps increase, which are reflected in the hollow circle curves of Fig. 2,3,5 and 8. (2) For a consistent decision data, the global trend of decision cost is monotonic with regard to the decision steps, and the decision cost will eventually reduce to zero (see the solid circle curves of Fig. 4-9). For an inconsistent decision data, the conclusion is almost the same, but the decision cost may not reduce to zero due to inconsistency of the decision in the data (see the solid circle curves of Fig. 2-3). (3) Compared to TCA, the TCD strategy presents a faster process to decrease the decision cost in most cases, but it is associated with a higher test cost in the sequential decision process.

Secondly, we test the changes of the boundary region when decision steps increase based on CSSD algorithm. In the experiments, we take mushroom data set as an example to test the global trend of the boundary region. The Fig. 10 presents the global trend of instances numbers of positive, boundary and negative regions based on CSSD algorithm with TCA strategy. We can conclude that the boundary region will globally decrease during the sequential decision process. Eventually, the boundary region disappear due to the sufficient available information for precise decision.

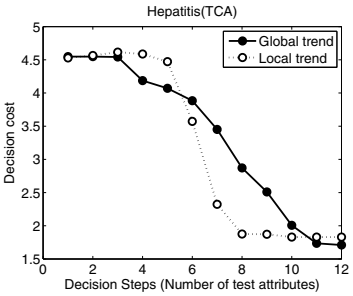


Fig. 2. Decision cost – Hepatitis (TCA)

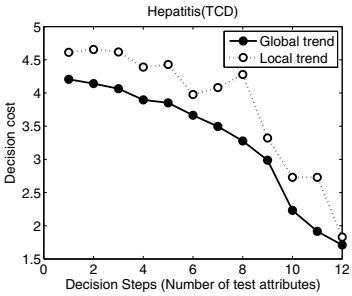


Fig. 3. Decision cost – Hepatitis (TCD)

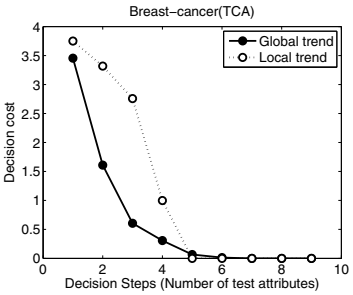


Fig. 4. Decision cost – Breast (TCA)

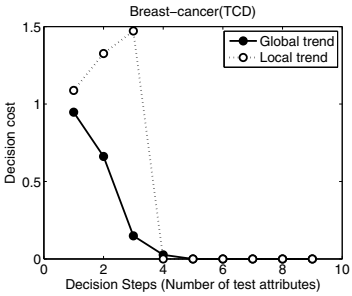


Fig. 5. Decision cost – Breast (TCD)

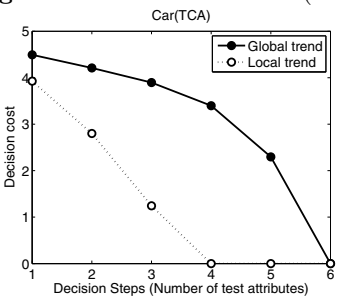


Fig. 6. Decision cost – Car (TCA)

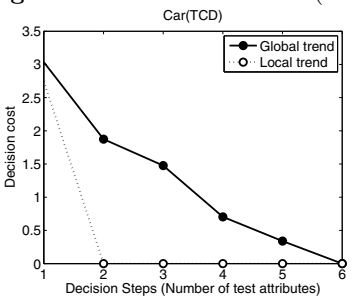


Fig. 7. Decision cost – Car (TCD)

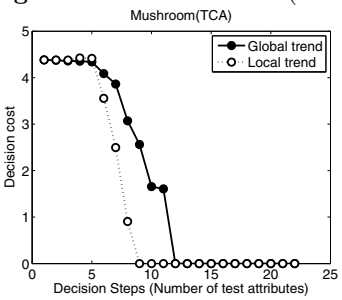


Fig. 8. Decision cost – Mushroom (TCA)

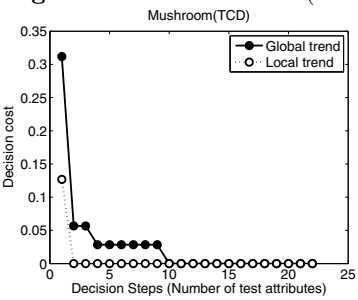


Fig. 9. Decision cost – Mushroom (TCD)

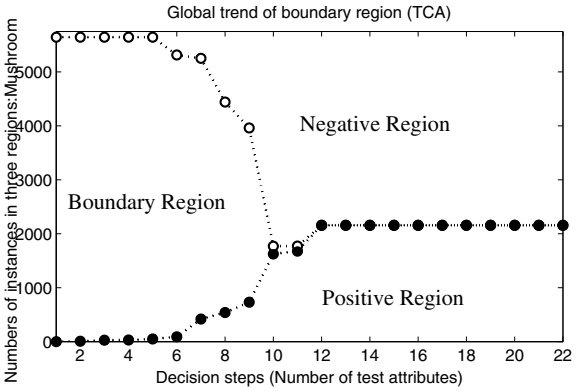


Fig. 10. Variation trends of positive, boundary, negative regions (Mushroom)

5 Conclusion

The objective of this paper is to simulate a sequential decision process which takes the strategy from rough granule to precise granule. A cost-sensitive sequential three-way decision model is presented in the paper. In the beginning, only a few available information can be used for decision, and the decision results have a higher decision cost with a rough granule view. With the increasing of the decision steps, the decision cost decreases and the boundary region reduces globally. Eventually, the decision cost reach a satisfying threshold and all instances are precisely decided. The theoretic analysis on the proposed decision model are presented and the experimental analysis validate the related propositions. In future work, we will further investigate the attribute selection methods which represent different sequential decision strategies.

Acknowledgments. This research is supported by the National Natural Science Foundation of China under grant No. 70971062, 71201076, 61170105, 71201133, the Natural Science Foundation of Jiangsu, China (BK2011564), and the Ph.D. Programs Foundation of Ministry of Education of China (20120091120004).

References

1. Herbert, J.P., Yao, J.T.: Game-theoretic risk analysis in decision-theoretic rough sets. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 132–139. Springer, Heidelberg (2008)
2. Herbert, J.P., Yao, J.T.: Game-theoretic rough sets. Fundamenta Informaticae (3-4), 267–286 (2011)

3. Jia, X.Y., Shang, L., Chen, J.J.: Attribute reduction based on minimum decision cost. Journal of Frontiers of Computer Science and Technology 5, 155–160 (2011) (in Chinese)
4. Li, H.X., Liu, D., Zhou, X.Z.: Survey on decision-theoretic rough set model. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition) 22, 624–630 (2010) (in Chinese)
5. Li, H.X., Zhou, X.Z., Zhao, J.B., Huang, B.: Cost-Sensitive classification based on decision-theoretic rough set model. 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. 379–388. Springer, Heidelberg (2012)
6. Li, H.X., Zhou, X.Z., Zhao, J.B., Liu, D.: Attribute reduction in decision-theoretic rough set model: A further investigation. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 466–475. Springer, Heidelberg (2011)
7. Li, H.X., Zhou, X.Z.: Risk decision making based on decision-theoretic rough set: A three-way view decision model. International Journal of Computational Intelligence Systems 4, 1–11 (2011)
8. Li, H.X., Zhou, X.Z., Li, T.R., Wang, G.Y., Miao, D.Q., Yao, Y.Y. (eds.): Decision-Theoretic Rough Sets Theory and Recent Research. Science Press, Beijing (2011) (in Chinese)
9. Li, W., Miao, D.Q., Wang, W.L., Zhang, N.: Hierarchical rough decision theoretic framework for text classification. In: Proceedings of ICCI 2010, pp. 484–489. IEEE Press (2010)
10. Liu, D., Li, H.X., Zhou, X.Z.: Two decades' research on decision-theoretic rough sets. In: Proceedings of ICCI 2010, pp. 968–973. IEEE Press (2010)
11. Liu, D., Li, T.R., Hu, P., Li, H.X.: Multiple-category classification with decision-theoretic rough sets. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS, vol. 6401, pp. 703–710. Springer, Heidelberg (2010)
12. Liu, D., Li, T.R., Li, H.X.: A multiple-category classification approach with decision-theoretic rough sets. Fundamenta Informaticae 115, 173–188 (2012)
13. Liu, D., Li, T.R., Ruan, D.: Probabilistic model criteria with decision-theoretic rough sets. Information Sciences 181, 3709–3722 (2011)
14. Min, F., Liu, Q.H.: A hierarchical model for test-cost-sensitive decision systems. Information Sciences 179, 2442–2452 (2009)
15. Min, F., He, H.P., Qian, Y.H., Zhu, W.: Test-cost-sensitive attribute reduction. Information Sciences 181, 4928–4942 (2011)
16. Yao, J.T., Herbert, J.P.: Web-based support systems with rough set analysis. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 360–370. Springer, Heidelberg (2007)
17. Yao, Y.Y.: Decision-theoretic rough set models. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 1–12. Springer, Heidelberg (2007)
18. Yao, Y.Y.: The superiority of three-way decision in probabilistic rough set models. Information Sciences 181, 1080–1096 (2011)
19. Yao, Y.Y.: Three-way decision: An interpretation of rules in rough set theory. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 642–649. Springer, Heidelberg (2009)
20. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180, 341–353 (2010)
21. Yao, Y.Y., Deng, X.F.: Sequential three-way decisions with probabilistic rough sets. In: Proceedings of ICCI*CC 2011, pp. 120–125. IEEE Press (2011)

22. Yao, Y.Y., Wong, S.K.M., Lingras, P.: A decision-theoretic rough set model. In: Methodologies for Intelligent Systems, vol. 5, pp. 17–24. North-Holland, New York (1990)
23. Yao, Y.Y., Zhao, Y.: Attribute reduction in decision-teoretic rough set models. Information Sciences 178, 3356–3373 (2008)
24. Yu, H., Chu, S.S., Yang, D.C.: Autonomous knowledge-oriented clustering using decision-theoretic rough set theory. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS, vol. 6401, pp. 687–694. Springer, Heidelberg (2010)
25. Zhou, X.Z., Li, H.X.: A multi-view decision model based on decision-theoretic rough set. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 650–657. Springer, Heidelberg (2009)

Two-Phase Classification Based on Three-Way Decisions

Weiwei Li[1], Zhiqiu Huang[1], and Xiuyi Jia[2]

[1] College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics, Nanjing, China, 210016
{liweiwei,zqhuang}@nuaa.edu.cn
[2] School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing, China, 210094
jiaxy@njust.edu.cn

Abstract. A two-phase classification method is proposed based on three-way decisions. In the first phase, all objects are classified into three different regions by three-way decisions. A positive rule makes a decision of acceptance, a negative rule makes a decision of rejection, and a boundary rule makes a decision of abstaining. The positive region contains those objects that have been assigned a class label with a high level of confidence. The boundary and negative regions contain those objects that have not been assigned class labels. In the second phase, a simple ensemble learning approach to determine the class labels of objects in the boundary or negative regions. Experiments are performed to compare the proposed two-phase classification approach and a classical classification approach. The results show that our method can produce a better classification accuracy than the classical model.

Keywords: Two-phase classification, decision-theoretic rough set model, ensemble learning, three-way decisions.

1 Introduction

In the last few years, many researchers [3, 5, 6] have concentrated on the study of decision-theoretic rough set model. Decision-theoretic rough set model [12] makes two main contributions to rough set theory. One is to provide a sound theoretic framework for calculating the thresholds required in probabilistic rough set models. It can derive several probabilistic rough set models when proper cost functions are used, such as Pawlak rough set model [8], 0.5 probabilistic rough set model [9], variable precision rough set model [18] and Bayesian rough set model [10]. The other is to give the semantic interpretation of the positive boundary and negative regions which are commonly used in all rough set models. The notion of three-way decisions, consisting of positive, boundary and negative rules, comes closer to the philosophy of the rough set theory, namely, representing a concept by using three regions [9].

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 338–345, 2013.

Yao [13–15] analyzed the superiority of three-way decisions in probabilistic rough set models from a theoretic perspective. Several researchers [4, 17] applied three-way decisions in spam filtering. Liu et al. [7] proposed a framework for three-way investment decisions with decision-theoretic rough set model. Li et al. [5] proposed a hierarchical framework for text classification based on decision-theoretic rough set model. Li et al. [6] combined misclassification cost and test cost to determine the classification result based on decision-theoretic rough set model.

For classification problem, an advantage of using three-way decisions over two-way decisions is that three-way decisions can classify some potentially misclassified objects into the boundary region for a further-exam, which may lead to lower misclassification error and lower misclassification cost. In the framework of three-way decisions, it is better to defer assigning a definite class label to objects in the boundary region. An intuitive and reasonable interpretation is that available information is not enough to classify these objects. For a binary classification problem or a multiple classification problem, the negative region of the decision table also contains objects that cannot be assigned a definite class label. For many applications, users want to get a definite result without requiring any additional information, which requires us to give a definite mechanism to handle the boundary region and the negative region. By reviewing current research, we found that not many studies focus on making further classification on objects needing further examination.

In this paper, we study how to apply three-way decisions to classical classification problem. We propose a two-phase classification scheme. In the first phase, a specific classifier will be chosen as the base classifier to compute the probability distribution of each object. After comparing to the thresholds, all objects will be classified into three regions. For all objects in the positive region, they will be assigned the corresponding class labels. In the second phase, for the remaining objects, several classifiers will be combined to vote for the final class labels. The experimental results show that the two-phase method can produce a higher classification accuracy than the classical method.

2 Three-Way Decisions with Decision-Theoretic Rough Sets

Decision-theoretic rough set model was proposed by Yao et al. [12] based on Bayesian decision theory. The basic ideas of the theory [13] are reviewed in this section.

A decision table is the following tuple:

$$S = (U, At = C \cup D, \{V_a | a \in At\}, \{I_a | a \in At\}), \tag{1}$$

where U is a finite nonempty set of objects, At is a finite nonempty set of attributes, C is a set of condition attributes describing the objects, and D is a set of decision attributes that indicates the classes of objects. V_a is a nonempty

set of values of $a \in At$, and $I_a : U \to V_a$ is an information function that maps an object in U to exactly one value in V_a.

In rough set theory [8], a set X is approximated by three regions. The positive region $\mathrm{POS}(X)$ contains objects that surely belong to X, the boundary region $\mathrm{BND}(X)$ contains objects that possibly belong to X and X^c at the same time, and the negative region $\mathrm{NEG}(X)$ contains objects that do not belong to X.

With respect to these three regions, the set of state is given by $\Omega = \{X, X^c\}$, and the set of actions is given by $\mathcal{A} = \{a_P, a_B, a_N\}$, where a_P, a_B and a_N represent the three actions in classifying an object x, namely, deciding $x \in \mathrm{POS}(X)$, deciding $x \in \mathrm{BND}(X)$, and deciding $x \in \mathrm{NEG}(X)$. Let λ_{PP}, λ_{BP} and λ_{NP} denote the costs incurred for taking actions a_P, a_B, a_N, respectively, when an object belongs to X, and let λ_{PN}, λ_{BN} and λ_{NN} denote the costs incurred for taking these actions when the object does not belong to X. Let $p(X|x)$ be the conditional probability of an object x being in state X.

The Bayesian decision procedure suggests the following minimum-cost decision rules [13]:

(P) If $p(X|x) \geq \alpha$ and $p(X|x) \geq \gamma$, decide $x \in \mathrm{POS}(X)$,
(B) If $p(X|x) \leq \alpha$ and $p(X|x) \geq \beta$, decide $x \in \mathrm{BND}(X)$,
(N) If $p(X|x) \leq \beta$ and $p(X|x) \leq \gamma$, decide $x \in \mathrm{NEG}(X)$,

where

$$\alpha = \frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})},$$
$$\beta = \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})},$$
$$\gamma = \frac{(\lambda_{PN} - \lambda_{NN})}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}. \tag{2}$$

Each rule is defined by two out of the three parameters. The conditions of rule (B) suggest that $\alpha > \beta$ may be a reasonable constraint; it will ensure a well-defined boundary region. If the cost functions satisfy the following condition [13]:

$$\frac{(\lambda_{NP} - \lambda_{BP})}{(\lambda_{BN} - \lambda_{NN})} > \frac{\lambda_{BP} - \lambda_{PP}}{(\lambda_{PN} - \lambda_{BN})}, \tag{3}$$

then $0 \leq \beta < \gamma < \alpha \leq 1$. In this case, after tie-breaking, the following simplified rules are obtained:

(P1) If $p(X|x) > \alpha$, decide $x \in \mathrm{POS}(X)$;
(B1) If $\beta \leq p(X|x) \leq \alpha$, decide $x \in \mathrm{BND(X)}$;
(N1) If $p(X|x) < \beta$, decide $x \in \mathrm{NEG}(X)$.

The threshold parameters are systematically calculated from cost functions based on the Bayesian decision theory.

3 Two-Phase Classification Mechanism

Based on Bayesian decision theory, decision-theoretic rough set model provides a three-way decisions scheme for the classification problem. Compared to a

two-way decisions method, the classification result given by a three-way decisions method may have a smaller classification cost. For most situations, a three-way decisions method also has a lower misclassification error rate. However, a lower misclassification error rate does not follow by a higher classification accuracy because their sum may not be 1 in a three-way decisions method. Usually, rejection rate can be defined to evaluate the size of the boundary region in three-way decisions. By introducing rejection rate, both misclassification error rate and classification accuracy decrease in most cases.

In classical classification problem, a desirable result is a high classification accuracy and a low misclassification error rate with no unclassified objects. To reach this goal, how to deal with the boundary and negative regions of the decision table will be crucial. We will introduce a two-phase classification method and adopt an ensemble strategy to deal with the boundary and negative regions.

First, a classifier is selected as the base classifier and train it using WEKA [2]. Then the trained classifier can provide the probability distribution $p(D_i|x)$ for each object x. Given by experts, all cost functions can be used to compute the thresholds (α, β). Then, an object x will be classified into one of three regions based on the thresholds and its probability:

$$\begin{aligned}
&\mathrm{POS}_{\alpha,\beta}(\pi_D) = \{x \in U | p(D_{max}(x)|x) > \alpha\},\\
&\mathrm{BND}_{\alpha,\beta}(\pi_D) = \{x \in U | \beta \leq p(D_{max}(x)|x) \leq \alpha\},\\
&\mathrm{NEG}_{\alpha,\beta}(\pi_D) = \{x \in U | p(D_{max}(x)|x) < \beta\},
\end{aligned} \tag{4}$$

where $D_{max}(x)$ is a dominant decision class of the object x, i.e., $D_{max}(x) = \arg\max_{D_i}\{p(D_i|x)\}$. By introducing the probability of the dominant decision class, we can deal with multi-class classification problem directly. In this stage, all objects in the positive region will be assigned a class label $D_{max}(x)$. For objects in the boundary and negative regions, we cannot give them labels as their probabilities are less than the threshold α.

In the second phase, we will focus on classifying the objects from the boundary and negative regions. In decision-theoretic rough set model, assigning these objects with labels will bring more misclassification cost as they are classified indistinctly under current thresholds. It can be understood that the base classifier is a kind of *weak learner* or *weak classifier*. We need some *strong learners* to overcome the weakness of the *weak learner* on the objects in the boundary and negative regions. Ensemble learning will be an intuitive and reasonable approach because the generalization ability of an ensemble is usually much stronger than that of a single learner [16]. We use multiple classifiers to vote for the final class labels, and it can be seen as a typical implementation of Stacking algorithm [11]. In the voting stage, the most frequent class label appeared will be the final result. If there exists a tie situation, for example, each class label appeared once, select one randomly as the final result. The detail of the two-phase classification algorithm is presented in Fig. 1.

INPUT: Training data set $\mathcal{D}$;
A cost matrix λ_{ij};
Multiple classifiers $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_T$;
An object x.
OUTPUT: predicted class label $H(x)$.
Begin:
Compute α, β from λ_{ij};
For $t = 1, \ldots, T$:
$h_t = \mathcal{C}_t(\mathcal{D})$; % Train a learner h_t by applying $\mathcal{C}_t$ to training data $\mathcal{D}$
End For;
$p_b(D_{max}(x)|x) = p(h_b(x))$; % Use h_b as the base learner to get the probability
% of the dominant decision class
If $p_b(D_{max}(x)|x) > \alpha$ **Then**
$H(x) = D_{max}(x)$; %Get the class label
Else
For $t = 1, \ldots, T$:
$L_t = h_t(x)$; % Using each learner to compute the class label L_t
End For;
$H(x) = vote\{L_1, L_2, \ldots, L_T\}$; % Select the most frequently class label,
% pick one randomly if tie situation exists
End If
End Begin

Fig. 1. The two-phase classification algorithm

4 Experiments

Experimental results are reported and analyzed to support the effectiveness of the two-phase classification algorithm.

4.1 Experiments' Settings

Five classical classifiers are selected as the multiple classifiers in experiments, which are NB (Naive Bayesian), C4.5, KNN, SVM and RBF [1]. All classifiers are implemented in WEKA (version 3.5) [2], and default values are used for all parameters in these classifiers. Several data sets from UCI [19] containing two classes data and multiple classes data are used in experiments. The details of all data sets are summarized in Table 1.

In the experiments, 10-fold cross validation is employed, and average results are recorded. In our experiments, we assume $\lambda_{PN} = 10$, $\lambda_{BP} = 2$, $\lambda_{BN} = 4$, $\lambda_{NP} = 100$, then $\alpha = 0.75$, $\beta = 0.039$ is gotten and will be used in our experiments.

Two classification approaches are compared in our experiments. One is two-phase classification algorithm, denoted by TPC. The other one is a classical single classifier approach, denoted by SC.

Table 1. Data sets from UCI machine learning repository

Data	Database description	Instances	Classes
wdbc	Breast Cancer Wisconsin (Prognostic)	569	2
wpbc	Breast Cancer Wisconsin (Diagnostic)	198	2
crx	Credit Approval	690	2
hepatitis	Hepatitis	155	2
house-votes-84	Congressional Voting Records	435	2
clean1	Musk (Version 1)	476	2
transfusion	Blood Transfusion Service Center	748	2
ionosphere	Ionosphere	351	2
bands	Cylinder Bands	541	2
hayes-roth	Hayes-Roth	160	3
iris	Iris	150	3
glass	Glass Identification	214	7
breastTissue	Breast Tissue	106	6
movement_libras	Libras Movement	360	15
CTG	Cardiotocography	2126	3
car	Car Evaluation	1728	4
anneal	Annealing	798	6
balance-scale	Balance-Scale	625	3

4.2 Experimental Results

The experimental results are presented in Table 2, Table 3 and Table 4. Table 2 gives the average classification accuracy of applying each classifier as the base classifier on different data sets. Table 3 shows the detailed accuracy values on data set *crx*. Table 4 shows the average classification accuracy of all classifiers on all data sets with respect to different values of threshold α.

From the results, we can see that TPC can get a better accuracy on all data sets except *iris*. The following conclusions can be drawn from the experimental results:

- Two-phase classification method is a feasible classification approach, which provides an effective mechanism for dealing with objects that need further examination in a three-way decision-theoretic rough set model.
- Two-phase classification method is independent of any classifier. From the average result shown in Table 2 and one detailed result shown in Table 3, we find that our method can produce a better accuracy no matter which classifier is used.
- We also test some other values of threshold α in our experiments. As the threshold α determines the positive region directly, $\alpha > 0.5$ is a rational constraint to avoid an object be classified into two different positive regions. In our experiment, five different values are tested and the experimental results are shown in Table 4. Our method can have a higher accuracy based on these threshold values. The accuracy is improved by the increase of value of α.

Table 2. Average classification accuracy on different data sets

Data	SC	TPC
wdbc	0.9371	**0.9389**
wpbc	0.7048	**0.7199**
crx	0.8443	**0.8542**
hepatitis	0.8283	**0.8293**
house-votes-84	0.9370	**0.9409**
clean1	0.8850	**0.8892**
transfusion	0.7676	**0.7687**
ionosphere	0.8928	**0.8997**
bands	0.7362	**0.7743**
hayes-roth	0.6771	**0.7413**
iris	**0.9560**	0.9533
glass	0.6259	**0.6924**
breastTissue	0.6500	**0.6842**
movement_libras	0.7239	**0.7533**
CTG	0.9467	**0.9603**
car	0.9068	**0.9335**
anneal	0.8040	**0.8670**
balance-scale	0.8576	**0.8742**
average	0.8156	**0.8375**

Table 3. Classification accuracy on crx

base classifier	SC	TPC
NB	0.8449	**0.8638**
C4.5	0.8478	**0.8507**
KNN	0.8246	**0.8449**
SVM	**0.8522**	**0.8522**
RBF	0.8522	**0.8594**

Table 4. Average classification accuracy on all data sets based on different α

α	SC	TPC
0.55	0.8156	**0.8257**
0.65	0.8156	**0.8317**
0.75	0.8156	**0.8375**
0.85	0.8156	**0.8415**
0.95	0.8156	**0.8470**

Two-phase classification method can get a better accuracy on reasonable thresholds than classical classification method.

5 Conclusions

How to deal with objects in the boundary and negative regions in three-way decision-theoretic rough sets is still an open problem. We introduced a two-phase classification mechanism based on three-way decisions to deal with this problem. In the first phase, objects in the positive region of the decision table are assigned the corresponding class labels. In the second phase, for the unlabelled objects which are classified into the boundary and negative regions, ensemble learning is applied to vote for the final class label. We tested five classifiers and different values of threshold α on several data sets, the experimental results show that the two-phase classification method can have a better classification accuracy than the classical classification method.

Acknowledgments. The authors thank Prof. Yiyu Yao, Prof. Masahiro Inuiguchi and other reviewers for the constructive comments and suggestions. This research is supported by the Fundamental Research Funds for the Central Universities under Grant No. NS2012129, the China Postdoctoral Science Foundation under Grant No. 2013M530259, and Postdoctoral Science Foundation of Jiangsu Province under Grant No. 1202021C.

References

1. Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)
2. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1), 10–18 (2009)
3. Jia, X.Y., Liao, W.H., Tang, Z.M., Shang, L.: Minimum cost attribute reduction in decision-theoretic rough set models. Information Sciences 219, 151–167 (2013)
4. Jia, X.Y., Zheng, K., Li, W.W., Liu, T.T., Shang, L.: Three-way decisions solution to filter spam email: An empirical study. In: Yao, J., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 287–296. Springer, Heidelberg (2012)
5. Li, W., Miao, D.Q., Wang, W.L., Zhang, N.: Hierarchical rough decision theoretic framework for text classification. In: Proceedings of the 9th International Conference on Cognitive Informatics, pp. 484–489 (2010)
6. Li, H.X., Zhou, X.Z., Zhao, J.B., Huang, B.: Cost-sensitive classification based on decision-theoretic rough set model. In: Li, T., Nguyen, H., Wang, G., Grzymała-Busse, J., Janicki, R., Hassanien, A., Yu, H. (eds.) RSKT 2012. LNCS, vol. 7414, pp. 379–388. Springer, Heidelberg (2012)
7. Liu, D., Yao, Y.Y., Li, T.R.: Three-way investment decisions with decision-theoretic rough sets. International Journal of Computational Intelligence Systems 4, 66–74 (2011)
8. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
9. Pawlak, Z.: Rough Sets, Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
10. Ślęzak, D., Ziarko, W.: The investigation of the Bayesian rough set model. International Journal of Approximate Reasoning 40, 81–91 (2005)
11. Wolpert, D.H.: Stacked generalization. Neural Networks 5(2), 241–260 (1992)
12. Yao, Y.Y., Wong, S.K.M., Lingras, P.: A decision-theoretic rough set model. Methodologies for Intelligent Systems 5, 17–24 (1990)
13. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180, 341–353 (2010)
14. Yao, Y.Y.: The superiority of three-way decisions in probabilistic rough set models. Information Sciences 181, 1080–1096 (2011)
15. Yao, Y.Y.: An outline of a theory of three-way decisions. In: Yao, J., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 1–17. Springer, Heidelberg (2012)
16. Zhou, Z.H.: Ensemble learning. In: Li, S.Z. (ed.) Encyclopedia of Biometrics, pp. 270–273. Springer, Heidelberg (2009)
17. Zhou, B., Yao, Y.Y., Luo, J.G.: Cost-sensitive three-way email spam filtering. Journal of Intelligent Information Systems (in press, 2013), doi:10.1007/s10844-013-0254-7
18. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Science 46, 39–59 (1993)
19. UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/`

A Three-Way Decisions Model Based on Constructive Covering Algorithm

Yanping Zhang[1,⋆], Hang Xing[1], Huijin Zou[1], Shu Zhao[1,⋆⋆], and Xiangyang Wang[2]

[1] School of Computer Science and Technology, Anhui University Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Hefei, Anhui Province, 230601, P.R. China
[2] Anhui Electrical Engineering Professional Technique College, Hefei, Anhui Province, 230051, P.R. China

Abstract. The three-way decisions model divides the universe into three regions, i.e., positive region (POS), boundary region (BND) and negative region (NEG) according to two thresholds. A challenge of the three-way decisions model is how to compute the thresholds that generally rely on the experience of experts. In this paper, we propose a novel three-way decisions model based on Constructive Covering Algorithm(CCA). The new model produces three regions automatically according to the samples and does not need any given parameters. We give a method for constructing coverings from which the three regions are formed. We can classify samples based on the three regions. The experimental results show that the proposed model has great advantage on the classification efficiency and provides a new method to form three regions automatically for the theory of three-way decisions.

Keywords: Constructive Covering Algorithm, three-way decisions, DTRSM, three regions, parameters.

1 Introduction

The theory of three-way decisions is proposed by Yao to interpret the sematic of three regions in rough set [1][2]. The three-way decisions model plays a key role in everyday decision-making and has been widely used in many fields and disciplines. Nowdays, the three-way decisions model is mainly based on rough set, i.e., Decision Theoretic of Rough Set Model (DTRSM). DTRSM is a typical probabilistic rough set model [3], in which two thresholds can be directly calculated from given loss functions based on the experience of experts. It divides the universe into three regions, i.e., positive region (POS), boundary region (BND) and negative region (NEG) based on the two thresholds. DTRSM is applied to many studies and applications [4][5]. In most applications, the thresholds are computed according to the given loss functions, i.e., the experience of experts. For example,

⋆ Yanping Zhang, Professor, Anhui University, main research in machine learning, artificial neural network and data mining.
⋆⋆ Corresponding author, Shu Zhao, email to: zhaoshuzs2002@hotmail.com

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 346–353, 2013.

Zhou and Li proposed a multi-view decision model based on DTRSM [6][7], in which optimistic decision, pessimistic decision and rational decision are proposed based on the cost of misclassification, namely, loss function. The parameters are according to the decision-maker. Li and Miao studied a rough decision theoretic framework for text classification [8]. Yu, Yang and Chu studied clustering algorithms based on DTRS [9][10]. Yao and Herbert studied Web-based support with rough set analysis [11]. However, loss functions are subjective and unreliable in some cases. It is difficult to give the loss functions precisely.

It is a challenge to calculate the thresholds, i.e., how to form the three regions. Some researchers have studied on this. Herbert and Yao used game-theoretic approach to calculate the two thresholds [12], in which tolerance values are provided to ensure correct approximation region size. Jia and Li proposed an adaptive learning parameters algorithm in three-way-decision-theoretic rough set [13], in which a pair of optimum parameters are calculated.

In this paper, we propose a three-way decisions model based on constructive covering algorithm (CCA). The formation of the three regions in the proposed model is based on CCA. CCA produces three regions automatically according to the distribution of samples and dose not need any parameters. We introduce to form the covers. Three regions are formed according to these covers and we can classify samples based on the three regions. We introduce CCA to the three-way decisions procedure. The proposed model can produce three regions automatically and does not need any parameters.

The rest of this paper is organized as follows. In section 2, we describe CCA briefly. In section 3, we present a new model based on CCA. Section 4 gives the experimental results. Finally, section 5 concludes the paper with a brief summary and further study.

2 Brief Description of CCA

Given a training samples set $X=\{(\boldsymbol{x}_1,\ y_1),\ (\boldsymbol{x}_2,\ y_2),\ldots,(\boldsymbol{x}_p,\ y_p)\}$, where $\boldsymbol{x}_i=(x_i^1,x_i^2,\ldots,x_i^n)$ $(i=1,2,\ldots,p)$ represents n-dimensional characteristic attribute of the ith sample. $\boldsymbol{x}_i$ can be regarded as an input vector, and y_i is the decision attribute, i.e., category [14][15].

2.1 Formation of Covers

The formation of the three regions is according to the covers. Firstly, we describe the formation of the covers as Algorithm 1.

Algorithm 1: formation of the covers.

Step 1: Map X to $(n+1)$-dimensional sphere S^{n+1} according to following formula:

$$T: X \to S^{n+1}, T(\boldsymbol{x}) = (\boldsymbol{x}, \sqrt{R^2 - |\boldsymbol{x}|^2}) \tag{1}$$

where $R \geqslant \max\{|\boldsymbol{x}|,\ \boldsymbol{x} \in \mathrm{X}\}$. Assume that the domain of input vectors X is a bounded set D of n-dimensional space. In most cases, the length of each sample

is not equal. As described in paper [16], we project them to an *n+1*-dimensional sphere, and make the length of each sample equal.

Step 2: Select a sample $\boldsymbol{x}_k$ as the center of a cover randomly.

Step 3: Compute the cover radius θ. Please see section 2.2 for details.

Step 4: Form a cover. Based on the center and the radius,we get a cover on S^{n+1}.

Step 5: Go back to Step 2 until all samples are covered.

To the end, we get a set of covers C=$\{C_1^1, C_1^2, \ldots, C_1^{n_1}, C_2^1, C_2^2, \ldots, C_2^{n_2}, \ldots, C_m^1, \ldots, C_m^{n_m}\}$, where C_i^j represents the jth cover of the ith category. We assume $C_i = \bigcup C_i^j$, j=1,2,. . . n_i. C_i represents all covers of the ith category samples.

2.2 Compute the Radius θ

The paper proposes a method to obtain cover radius θ. We compute θ described as the following three steps.

Step 1: Compute minimum radius θ_1

$$d_1(k) = \min dist(\boldsymbol{x}_k, \boldsymbol{x}_i), y_k \neq y_i, i \in \{1, 2, ...p\} \tag{2}$$

$$d_2(k) = \max\{dist(\boldsymbol{x}_k, \boldsymbol{x}_i) | dist(\boldsymbol{x}_k, \boldsymbol{x}_i) < d_1(k)\}, y_k = y_i, i \in \{1, 2, ...p\} \tag{3}$$

$$\theta_1 = d_2(k) \tag{4}$$

The minimum radius regards the max distance between the center and the similar points as the radius where the boundary does not have any dissimilar points as shown in Fig. 1.

Step 2: Compute maximum radius θ_2

$$d_1(k) = \min dist(\boldsymbol{x}_k, \boldsymbol{x}_i), y_k \neq y_i, i \in \{1, 2, ...p\} \tag{5}$$

$$\theta_2 = d_1(k) \tag{6}$$

The maximum radius regards the minimum distance between dissimilar points and the center as the radius as shown in Fig. 2.

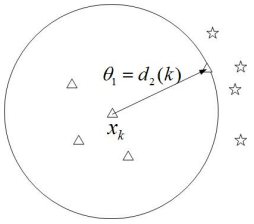


Fig. 1. Minimum radius θ_1

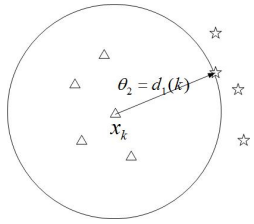


Fig. 2. Maximum radius θ_2

Step 3: Compute radius θ

$$\theta = (\theta_1 + \theta_2)/2 \tag{7}$$

3 New Three-Way Decisions Model Based on CCA

3.1 Definition of Three Regions Based on CCA

According to Algorithm 1, we give the definition of three regions. Because of finite samples, the covers cannot cover the space totally. In other words, there are some blank space which is not covered by the obtained covers. Moreover, the selection of radius can also result in overlap of covers. We call these regions boundary region (BND). In this paper, we only talk the first kind of BND. The definition of three regions is as follows.

Definition of three regions: For convenience in discussion, we assume only two categories C_1 and C_2. The covers of C_1 and C_2 are $(C_1^1,C_1^2,\ldots,C_1^m)$ and $(C_2^1,C_2^2,\ldots,C_2^n)$, respectively, i.e., $C_1=(C_1^1,C_1^2,\ldots,C_1^m)$, $C_2=(C_2^1,C_2^2,\ldots,C_2^n)$. Each category has at least a cover. Assume $C_i=\bigcup C_i^j$ and each C_i represents all covers of the i th category samples. We define POS of C_1, namely, POS(C_1) by the difference of unions $\bigcup C_1^i$-$\bigcup C_2^j$, NEG(C_1) by $\bigcup C_2^j$-$\bigcup C_1^i$ and BND(C_1) by the rest, where i=1,2,. . . ,m, j=1,2,. . . ,n. That is to say, POS(C_1) is equal to NEG(C_2); POS(C_2) is equal to NEG(C_1); BND(C_1) is equal to BND(C_2).

3.2 A New Three-Way Decisions Model

According to the three formed regions, we can make three-way decisions on a test sample x, i.e., the decision rules of x. We assume θ_1^i is the radius of C_1^i, θ_2^i is the radius of C_2^i, c_1^i is the center of C_1^i, and c_2^i is the center of C_2^i. The three-way decisions rules of x are shown as follows.

$x \in$POS(C_1), if dist(x,c_1^i)$\leq \theta_1^i$ and dist(x,c_2^j)$>\theta_2^j$, for all i=1. . . m,j=1. . . n;

$x \in$NEG(C_1), if dist(x,c_1^i)$>\theta_1^i$ and dist(x,c_2^j)$\leq\theta_2^j$, for at least one pair of (i,j) where i=1. . . m, j=1. . . n;

$x \in$BND(C_1), otherwise.

When x falls into a cover of C_1, x belongs to POS(C_1), which is equal to NEG(C_2). When x falls into a cover of C_2, x belongs to POS(C_2), which is equal to NEG(C_1). When x does not fall into any covers or falls into overlap of two covers which are different categories, x belongs to BND(C_1) (also BND(C_2)).

In the new model, the three regions are formed automatically based on distribution of the samples, and we do not need any parameters to form the three regions. We can use the decision rules to classify test samples.

4 Experimental Result

The experimental data used in this paper is from UCI Machine Learning Repository (`http://www.ics.uci.edu/mlearn/MLRepository.html`). Table 1 shows the datasets information as follows.

There are two parts in this section. The first part compares the proposed model with DTRSM using dataset Spambase and Chess on two categories. The second part compares the two models using datasets Iris and Wine on multi-categories.

Table 1. Benchmark datasets information used in the experiment

name	instances	attributes
Iris	150	5
Wine	178	14
Spambase	4601	58
Chess	3196	36

4.1 The Case of Two Categories

Firstly, we define three evaluation criteria.

$$Acc = CCI/TI \tag{8}$$

$$Err = ECI/TI \tag{9}$$

$$Bnd = BI/TI \tag{10}$$

where, CCI represents the number of correctly classified instances; ECI represents the number of mistakenly classified instances; BI represents the number of instances classified to the boundary; TI represents total instances. We compare the proposed model with DTRSM. In this paper, the implementation of DTRSM is based on [17]. We adopt 10-fold cross-validation in the experiments. α, β are the two thresholds. We select eleven pairs of thresholds. Table 2 and Table 3 shows the result of comparison.

Table 2. The comparison of two models on dataset Spambase

Model	CCI	ECI	BI	Acc(%)	Err(%)	Bnd(%)
DTRSM(0.8,0.2)	332	20	109	72.02	4.34	23.64
DTRSM(0.9,0.2)	313	16	128	68.49	3.5	28.01
DTRSM(0.9,0.3)	316	18	123	69.15	3.94	26.91
DTRSM(0.8,0.3)	333	21	104	72.71	4.59	22.7
DTRSM(0.8,0.4)	335	22	101	73.14	4.8	22.06
DTRSM(0.8,0.5)	338	23	97	73.8	5.02	21.18
DTRSM(0.7,0.2)	345	22	91	75.33	4.8	19.87
DTRSM(0.6,0.2)	378	27	52	82.71	5.91	11.38
DTRSM(0.6,0.1)	371	24	63	81	5.24	13.76
DTRSM(0.5,0.2)	409	38	10	89.49	8.32	2.19
DTRSM(0.5,0.1)	402	35	20	87.96	7.66	4.38
Proposed model	395	25	39	86.06	5.45	8.49

From Table 2 and Table 3, we can see that the thresholds have great influence on the result of classification accuracy in DTRSM. The thresholds are given by the experts, and it is difficult to obtain appropriate thresholds. From Table 2, we can see that the Acc of the proposed model is higher than that of DTRSM in most cases. And from Table 3, the Acc of proposed model is higher than that of DTRSM. Above all, the proposed model does not need any parameters and the classification effect is rather good while DTRSM needs given parameters and

the thresholds have great influence on the effect of classification. It is difficult to set appropriate thresholds. The advantage of the proposed model is that it can form three regions automatically. We can classify the samples based on the regions, and the classification effect of the proposed model is rather good.

Table 3. The comparison of two models on dataset Chess

Model	CCI	ECI	BI	Acc(%)	Err(%)	Bnd(%)
DTRSM(0.8,0.2)	148	10	157	46.69	3.15	50.16
DTRSM(0.9,0.2)	117	8	191	37.03	2.53	60.44
DTRSM(0.9,0.3)	136	16	164	43.04	5.06	51.9
DTRSM(0.8,0.3)	168	18	130	53.16	5.69	41.15
DTRSM(0.8,0.4)	183	27	106	57.91	8.54	33.55
DTRSM(0.8,0.5)	197	39	80	62.34	12.34	25.32
DTRSM(0.7,0.2)	175	14	127	55.38	4.43	40.19
DTRSM(0.6,0.2)	195	20	101	61.52	6.3	32.19
DTRSM(0.6,0.1)	168	16	131	53.33	5.08	41.59
DTRSM(0.5,0.2)	211	30	76	66.56	8.08	25.36
DTRSM(0.5,0.1)	184	26	107	58.04	8.2	33.76
Proposed model	237	42	37	75	13.29	11.71

4.2 The Case of Multi-categories

The DTRSM can classify multi-categories datasets. We describe the process in detail as follows.

To discuss conveniently, we assume only three categories C1, C2, C3. Firstly, we regard C2 and C3 as one category, and the number of category becomes two. Then we use DTRSM. Secondly, we regard C1 and C3 as one category, then we use DTRSM. Thirdly, we regard C1 and C2 as one category, then we use DTRSM.

In this part, we compare the proposed model with DTRSM using two datasets Iris and Wine. We use 10-fold cross-validation in the experiments. We select four best pairs of thresholds according to the classification efficiency and compare the proposed model with DTRSM. Table 4 shows the result of Iris dataset and Table 5 shows the result of Wine dataset.

Table 4. The result of comparison using dataset Iris

Model	CCI	ECI	BI	Acc(%)	Err(%)	Bnd(%)
DTRSM(0.6,0.2)	8.1	0.6	6.3	54	4	42
DTRSM(0.5,0.2)	8.6	0.6	5.8	57.3	4	38.67
DTRSM(0.4,0.2)	10.1	0.6	4.3	67.33	4	28.67
DTRSM(0.3,0.2)	13.9	0.6	0.5	92.67	4	3.33
Proposed model	14	0.4	0.6	93.33	2.67	4

From Table 4, we can see that the proposed model has great advantages on Acc, Err and Bnd. From Table 5, we can see that the Err of the proposed model

Table 5. The result of comparison using dataset Wine

Model	CCI	ECI	BI	Acc(%)	Err(%)	Bnd(%)
DTRSM(0.6,0.2)	14.4	1	1.2	86.75	6.02	7.23
DTRSM(0.5,0.2)	14.6	1	1	87.95	6.02	6.03
DTRSM(0.4,0.2)	15.1	1	0.5	90.96	6.02	3.02
DTRSM(0.3,0.2)	15.1	1	0.5	90.96	6.02	3.02
Proposed model	15	0.6	1.4	88.23	3.53	8.24

is lower than that of DTRSM. The proposed model is easier to process multi-categories classification. A few of Acc in DTRSM is better than that of the proposed model. The reason is that the Bnd of the proposed model is higher than that of DTRSM and the Err of the proposed model is lower than that of DTRSM. We will discuss how to deal with samples in BND in next paper. The proposed model does not need any parameters while the DTRSM need given parameters and the parameters have great influence on Acc, Err and Bnd. The proposed model has advantages on multi-categories instances classification.

5 Conclusions

In this paper, we introduced CCA to three-way decisions procedure and proposed a new three-way decisions model based on CCA. According to the samples, we get POS, NEG and BND automatically. The new model does not need any given parameters to form the regions.

The paper compares proposed model with DTRSM in two categories classification and multi-categories classification. DTRSM is a good method to deal with problem of three-way decisions. However, it needs loss functions for calculating the required thresholds. The proposed model makes up for the problem. It does not need any parameters and can form three regions automatically. The proposed model has three advantages: (1) it is easier to process multi-categories classification; (2) it can process discrete type data and continuous type data directly; (3) the most important one is that it provides a new method to form three regions automatically for three-way decisions. The experimental results show that the proposed model is superior in classification efficiency. A few of Acc in DTRSM is better than that of the proposed model. The reason is that the Bnd of the proposed model is higher than that of DTRSM and the Err of the proposed model is lower than that of DTRSM. Therefore, how to deal with samples in BND is our future research.

Acknowledgements. This work is partially supported by National Natural Science Foundation of China under Grant #61073117 and Grant #61175046, and supported by Natural Science Foundation of Anhui Province under Grant #11040606M145, and supported by Provincial Natural Science Research Program of Higher Education Institutions of Anhui Province under Grant #KJ2013A016, and supportde by 211 project of Anhui University.

References

1. Yao, Y.Y.: The superiority of three-way decisions in probabilistic rough set models. Information Sciences 181(6), 1080–1096 (2011)
2. Yao, Y.Y.: Two semantic issues in a probabilistic rough set model. Fundamenta Informaticae 108(3), 249–265 (2011)
3. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180(3), 341–353 (2010)
4. Yu, H., Wang, Y.: Three-way decisions method for overlapping clustering. In: Yao, J., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 277–286. Springer, Heidelberg (2012)
5. Lingras, P., Chen, M., Miao, D.Q.: Rough cluster quality index based on decision theory. IEEE Transaction on Knowledge and Data Engineering 21(7), 1014–1026 (2009)
6. Zhou, X.Z., Li, H.X.: A multi-view decision model based on decision-theoretic rough set. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 650–657. Springer, Heidelberg (2009)
7. Li, H.X., Zhou, X.Z.: Risk decision making based on decision-theoretic rough set:a multi-view decision model. International Journal of Computational Intelligence Systems 4(1), 1–11 (2011)
8. Li, W., Miao, D.Q., Wang, W.L., Zhang, N.: Hierarchical rough decision theoretic framework for text classification. In: Proceedings of 9th IEEE International Conference on Cognitive Informatics, pp. 484–489. IEEE Press (2010)
9. Yu, H., Chu, S.S., Yang, D.C.: Autonomous Knowledge-oriented clustering using decision-theoretic rough set theory. Fundamenta Informaticae 115(2), 141–156 (2012)
10. Yu, H., Chu, S.S., Yang, D.C.: A semiautonomous clustering algorithm based on decision-theoretic rough set theory. In: Proceedings of 9th IEEE International Conference on Cognitive Informatics, pp. 477–483. IEEE Press (2010)
11. Yao, J.T., Herbert, J.P.: Web-based support systems with rough set analysis. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 360–370. Springer, Heidelberg (2007)
12. Herbert, J.P., Yao, J.T.: Game-Theoretic Risk Analysis in Decision-Theoretic Rough Sets. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 132–139. Springer, Heidelberg (2008)
13. Jia, X.Y., Li, W.W.: An Adaptive Learning Parameters Algorithm in Three-way-decision-theoretic Rough Set Model. Acta Electronica Sinica 39(11), 2520–2525 (2011)
14. Zhang, X.H., Pei, D.W., Dai, J.H.: Fuzzy mathematics and theory of Rough Set. Tshing Hua University Press, Beijing (2013) (in Chinese)
15. Zhao, S., Zhang, Y.P., Zhang, L.: Probability Model of Covering Algorithm. In: International Conference on Intelligent Computing, Part 1, pp. 440–444 (2006) (in Chinese)
16. Zhang, L., Zhang, B.: A geometrical representation of McCulloch-Pitts neural model and its applications. IEEE Trans. on Neural Networks 10(4), 925–929 (1999)
17. Zhou, B., Yao, Y.Y., Luo, J.: A three-way decision approach to email spam filtering. In: Farzindar, A., Kešelj, V. (eds.) Canadian AI 2010. LNCS, vol. 6085, pp. 28–39. Springer, Heidelberg (2010)

A Hierarchical Statistical Framework for the Extraction of Semantically Related Words in Textual Documents

Weijia Su[1], Djemel Ziou[2], and Nizar Bouguila[1]

[1] Concordia Institute for Information Systems Engineering
Concordia University, Montreal, QC, Canada
s_weijia@encs.concordia.ca, nizar.bouguila@concordia.ca
[2] Department of Computer Science
University of Sherbrooke, Sherbrooke, QC, Canada
Djemel.Ziou@USherbrooke.ca

Abstract. Nowadays there exist a lot of documents in electronic format on the Internet, such as daily news and blog articles. Most of them are related, organized and archived into categories according to their themes. In this paper, we propose a statistical technique to analyze collections of documents, characterized by a hierarchical structure, to extract information hidden into them. Our approach is based on an extension of the log-bilinear model. Experimental results on real data illustrate the merits of the proposed statistical hierarchical model and its efficiency.

Keywords: Log-bilinear model, hierarchical modeling, semantically related words.

1 Introduction

More and more textual data are digitized and stored online. These data bring us both valuable information and management challenges. Thus, many researches have focused on language modeling using statistical methods to extract useful knowledge from these data. By describing texts in mathematical ways, hidden structures and properties within texts and correlations between them can be discovered, which can help practitioners to organize and manage them more easily. A good organization has many applications. For instance, several studies [1–3] have shown that cyber criminals generally exchange their experiences and knowledge via media such as forums and blogs. These exchanged data, if well extracted and modeled, can provide significant clues to agencies operating in the security field. According to the method used to represent words in documents, modeling approaches can be grouped into two categories: probabilistic topic models and vector space models. Probabilistic topic models such as probabilistic latent semantic indexing (PLSI) [4] and latent Dirichlet allocation (LDA) [5], model a text document as a finite mixture of specific distributions over topics. Each topic is represented as a distribution of words in a given defined vocabulary set.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 354–363, 2013.

They have been applied in various applications to extract semantic properties (e.g. topics, authors' influence, citations relations, etc.) within a document [6–8]. Vector space model (VSM) [9] represents documents as vectors where each vector can be viewed as a point in a multi-dimensional space. The basic idea behind VSM is that in space, the closer the two points are, the more semantic similarity they are sharing and vice versa. VSM approach has shown excellent performance in many real world tasks related to the measurement of semantic similarities between documents, sentences, and words [10–13].
All the methods mentioned above have focused on modeling documents individually, while in real world most documents are related, and organized into hierarchical categories according to their themes. Thus, it is crucial to develop models that take into account these aspects [14–16]. In this paper, we propose a hierarchical statistical model to analyze documents. The proposed model is part of a large cyber security forensics system that we are designing to discover and capture potential security threats by retrieving and analyzing data gathered from the Web. In our method, each node in the structure is modeled using probabilities. A log-bilinear model is adopted to describe words in vector space in such a way that their correlations can be discovered and derived, from their representations, at each level of the hierarchical structure. The rest of this paper is organized as follows. In Section 2, we present the hierarchical statistical model in details, and we present the complete algorithm to estimate its parameters. The experimental results of applying our approach on real data are presented in Section 3. Finally, Section 4 gives the conclusion.

2 Hierarchical Statistical Document Model

The improvement of the state of the art concerning document modeling has been based on three main groups of approaches [17]. The first group has been concerned with the improvement of current learning techniques. The second one has been based on the development of better features. The third one focused on the integration of prior information about the relationship between document classes. The technique that we shall propose in this section belongs to the third group, since our main goal here is take advantage of the hierarchical relationship usually present between classes. Indeed, the automatic extraction of a given document topic and semantic information about a given word meaning generally involves a hierarchy of a large number of classes. The hierarchy encodes crucial information that should be exploited when learning a given model. Thus, we propose here the extension of the log-bilinear model to incorporate the fact that document classes are generally hierarchical. In this section, we start by reviewing the basic log-bilinear model and then we generalize it to encode hierarchies.

2.1 Log-Bilinear Document Model

A log-bilinear model which learns the semantic word vectors from term-document data was introduced in [18]. In this model, a document is represented as a distribution of conditionally independent words given a parameter θ:

$$p(d) = \int p(d, \theta) = \int p(\theta) \prod_{i=1}^{N} p(w_i|\theta) d\theta \tag{1}$$

where d is a document, N is the total number of words in d and w_i represents each word in d. A Gaussian prior is considered for θ.
The model uses bag-of-words representation to describe a document in which words sequences appear in an exchangeable way. The fixed vocabulary set is denoted as V and has a size of $|V|$. Each word is represented by a $|V|$- dimensional vector where only one element is equal to 1 and all the others are equal to 0 (i.e. one-hot vector). The word conditional distribution $p(w|\theta)$ in the document is defined by a log-linear model with parameters R and b. The word representation matrix is $R \in \Re^{\beta \times |V|}$ and contains the β-dimensional vector representation $\phi_w = Rw$ of each word in the vocabulary set. Therefore, the representation, ϕ_w, of each word is the corresponding column in R. Also, θ is a β-dimensional vector which works as a weighting component for the word vector representation. Moreover, the word frequency differences are captured via a parameter b_w. Given all these parameters, the log-bilinear energy assigned to each word is:

$$E(w; \theta, \phi_w, b_w) = -\theta^T \phi_w - b_w \tag{2}$$

Therefore, the word distribution using softmax is given by:

$$p(w|\theta; R, b) = \frac{\exp(-E(w; \theta, \phi_w, b_w))}{\sum_{w' \in V} \exp(-E(w'; \theta, \phi_{w'}, b_{w'}))} = \frac{\exp(\theta^T \phi_w + b_w)}{\sum_{w' \in V} \exp(\theta^T \phi_{w'} + b_{w'})} \tag{3}$$

Note that this model can only find semantic information at the document level.

2.2 Hierarchical Statistical Document Model

In real-world applications, online texts are often classified into categories with respect to their themes. Thus, these texts usually have a hierarchical structure. Moreover, words are hierarchical by nature, since they may be related to different other words at different categories. In this subsection, we extend the log-bilinear document model to take hierarchical structures into account. The main goal is to discover semantic information such as word relations at each level of the hierarchical structure. Modeling a collection of documents into different levels can be achieved by building a probabilistic model for each node in the hierarchical structure. Suppose that we have a node m, which has a total number of N_k children denoted as m_k. Each child node is considered to be a documents collection composed of N_{tk} documents which supposed to be conditionally independent given a variable θ_{jk}. Thus, the probability of node m can be written as:

$$p(m) = \prod_{k=1}^{N_k} \prod_{j=1}^{N_{tk}} \int p(\theta_{jk}) p(d_{jk}|\theta_{jk}) d\theta_{jk} \tag{4}$$

where d_{jk} denotes the jth document in the child node m_k, θ_{jk} is a mixing variable corresponding to document d_{jk}, and $p(\theta_{jk})$ is a gaussian prior. Each document consists of conditionally independent distributed words:

$$p(d_{jk}|\theta_{jk}) = \prod_{i=1}^{N_{wtk}} p(w_{ijk}|\theta_{jk}) \tag{5}$$

where N_{wtk} is the total number of words in document d_{jk}, which actually belongs to m_k, and w_{ijk} denotes the words inside the documents. By combining equations 4 and 5, we obtain the distribution of the node m:

$$p(m) = \prod_{k=1}^{N_k} \prod_{j=1}^{N_{tk}} \int p(\theta_{jk}) \prod_{i=1}^{N_{wtk}} p(w_{ijk}|\theta_{jk}) d\theta_{jk} \tag{6}$$

In the equation above, the p.d.f for each word, $p(w_{ijk}|\theta_{jk})$, is defined by Equation 3 in the previous section. It is worth mentioning that the model can also be applied to classify nodes which are at the same level of the hierarchical collection. This can be achieved by treating each node as an individual document containing words from all the documents it consists of. Therefore, the model can be trained to use parameter θ to distinguish each node from its siblings.

2.3 Model Learning

The model can be learned by maximizing the probability of observed data at each node. The parameters are learned by iteratively maximizing $p(m)$ with respect to θ, word representation R, and word frequency bias b:

$$\hat{\theta}, \hat{R}, \hat{b} = \max_{\theta,R,b} \prod_{k=1}^{N_k} \prod_{j=1}^{N_{tk}} \int p(\theta_{jk}) \prod_{i=1}^{N_{wtk}} p(w_{ijk}|\theta_{jk}) d\theta_{jk} \tag{7}$$

Therefore, the log-likelihoods for θ_{jk}, and for R and b are:

$$L(\theta_{jk}) = \sum_{j=1}^{N_{tk}} (\sum_{i=1}^{N_{wtk}} \log(p(w_{ijk}|\theta_{jk})) - \lambda \theta_{jk}^2) \tag{8}$$

$$L(R, b) = \sum_{k=1}^{N_k} \sum_{j=1}^{N_{tk}} \log(p(\theta_{jk})) \sum_{i=1}^{N_{wtk}} \log(p(w_{ijk}|\theta_{jk})) \tag{9}$$

where λ is the scale parameter of the Gaussian. We take partial derivative with respect to θ_{jk} in Equation 8, to get the gradient:

$$\nabla_{\theta_{jk}} = \frac{\partial L(\theta_{jk})}{\partial \theta_{jk}} = \sum_{i=1}^{N_{wtk}} (\phi_{w_{ijk}} - \sum_{w' \in V} p(w'|\theta_{jk}) \phi_{w'}) - 2\lambda \theta_{jk} \tag{10}$$

Then, we take partial derivative with respect to R and b in Equation 9. For each column R_v of the representation matrix, the gradient ∇_{R_v} is:

$$\nabla_{R_v} = \frac{\partial L(R,b)}{\partial R_v} = \sum_{k=1}^{N_k} \sum_{j=1}^{N_{tk}} \sum_{i=1}^{N_{wtk}} (N_{w_v}\theta_{jk} - N_{wtk} p(w_v|\theta_{jk})\theta_{jk}) \tag{11}$$

And the gradient for b is:

$$\nabla_{b_v} = \frac{\partial L(R,b)}{\partial b_v} = \sum_{k=1}^{N_k} \sum_{j=1}^{N_{tk}} \sum_{i=1}^{N_{wtk}} (N_{w_v} - N_{wtk} p(w_v|\theta_{jk})) \tag{12}$$

Therefore, at each step of the iteration, θ, R and b are updated as:

$$\theta_{jk}^{t+1} = \theta_{jk}^{t} + \alpha \nabla_{\theta_{jk}} \tag{13}$$

$$R_v^{t+1} = R_v^t + \alpha \nabla_{R_v} \qquad b_v^{t+1} = b_v^t + \alpha \nabla_{b_v} \tag{14}$$

Thus, the parameters are optimized by moving in the direction of the gradient. The step size of the movement is indicated by α. The procedure of estimating the model's parameters is based on iteratively optimizing the values of θ, R, and b using Newton's method. It first optimizes θ for each collection child with R and b fixed. Afterwards, we optimize word representation R and bias b with θ fixed. We repeat these two steps until convergence. The complete learning procedure is shown in Algorithm 1. In the proposed model, the related words are found by

Algorithm 1. Model Learning Algorithm

1: Initialize the values of parameters θ , R, and b with randomly generated numbers, set the step size ($\alpha = 1e-4$) and iteration convergence criteria (maximum iteration number $MaxIter = 1000$ and evaluation termination value $TermVal = 1e-7$).
2: Repeat
3: Estimate θ_{jk} at each node using Eq. 13.
4: Optimize R and b using Equations in 14.
5: Until one of the convergence criteria is reached (The iteration exceeds $MaxIter$ or the change of the parameters values is less than $TermVal$)

calculating the cosine similarities between words from the word representation vectors ϕ, which are derived from the representation matrix R. Therefore, for words w_1 and w_2, with representation vectors ϕ_1 and ϕ_2, the similarity is:

$$Similarity(w_1, w_2) = \frac{Rw_1 \cdot Rw_2}{||Rw_1||||Rw_2||} = \frac{\phi_1 \cdot \phi_2}{||\phi_1||||\phi_2||} \tag{15}$$

3 Experiments

In this section, we investigate our proposed hierarchical statistical document model using two challenging tasks. The first one is to find semantically related words for a query word at each level of a collection of documents characterized by a hierarchical structure. The second task is to show the model's performance on words classification. Our experiments have been performed on a 2.70GHz Intel i7 machine (4GB RAM, 64-bit operating system) using Matlab version R2010b.

3.1 Finding Semantically Related Words

The data set in this first experiment is composed of web pages gathered from Wikipedia. The data is obtained via "Wikipedia Export", which allows to retrieve web pages from the database in specific categories. Then, the plain text of each web page is extracted. Afterwards, data are pre-processed by consulting each word property with WordNet to filter stop words (e.g. "the", "and", "or", etc.) and non english words. Only nouns, verbs, adjectives and adverbs are kept. Furthermore, the nouns and verbs are converted to their roots, for example "ate" is changed to "eat", and "cats" is transformed to "cat". This can help us to eliminate the redundancy of a root word presented in multiple formats. The similarity scores between words are derived using the cosine measure.
Here, we report our experimental results on words learned under the "crime" category in Wikipedia. The structure of this collection of documents is displayed in figure 1. As we can see from this figure, the root node is "Crime", which con-

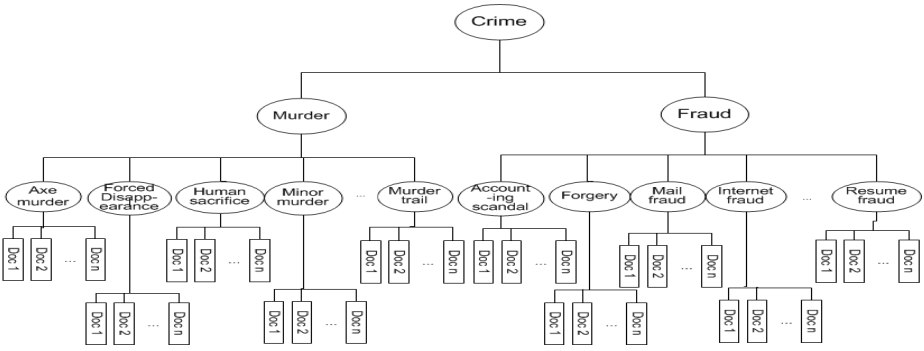


Fig. 1. The hierarchical structure of "Crime" category

tains 5372 documents and has "Fraud" and "Murder" categories as children. The node "Fraud" contains 4341 documents and the node "Murder" contains 1391 documents. Both of them have 14 nodes as children. We report the results found by our model at these three nodes. The most frequently used 2500 words in our nodes are selected to build the vocabulary set. Tables 1, 2 and 3 show

part of the most frequent words as well as their semantically related words when using cosine similarity scores computed at these three nodes. The results in these tables show clearly that our model has a good performance in finding different related words.

Table 1. Semantically related words at node "Crime"

Word	shoot	score	attack	score	murder	score	bury	score
	kill	0.881	wound	0.738	kill	0.830	cremate	0.820
Similar	ambush	0.810	bomb	0.724	mutilate	0.757	die	0.760
Words	gun	0.762	overpower	0.706	confess	0.739	burn	0.712
	fire	0.761			assassinate	0.732	exhume	0.708
	wound	0.750			stab	0.732	survive	0.706
Word	invest	score	disappear	score	marry	score	lie	score
	trade	0.818	vanish	0.840	move	0.848	hear	0.727
Similar	promise	0.759	miss	0.704	bear	0.789	tell	0.718
Words	buy	0.742			die	0.781		
					emigrate	0.759		
					divorce	0.725		

Table 2. Semantically related words at node "Murder"

Word	shoot	score	attack	score	murder	score	disappear	score
	kill	0.906	injure	0.870	kill	0.906	force	0.813
Similar	die	0.878	wound	0.843	try	0.863	kidnap	0.806
Words	fire	0.820	stop	0.765	commit	0.750	detain	0.774
	attempt	0.807	coordinate	0.708	die	0.845	miss	0.770
	murder	0.737			shoot	0.737	confirm	0.711
Word	assassinate	score	fire	score	injure	score	investigate	score
	condemn	0.814	wound	0.838	wound	0.904	conclude	0.767
Similar	oppose	0.807	shoot	0.821	attack	0.870	solve	0.763
Words	execute	0.712	injure	0.773	fire	0.773	examine	0.709
	fail	0.710	occur	0.748	occur	0.752	indicate	0.705
	escape	0.704	surrender	0.723	explode	0.708	file	0.704

3.2 Word Classification

In this subsection, we investigate the performance of our model on word classification problem. The data set used in this experiment is collected from "Thesaurus" web site. In this web site, words are classified into 6 categories: 1) words expressing abstract relations, 2) words related to space, 3) words related to matter, 4) words related to intellectual faculties, formation and communication of ideas, 5) words related to voluntary powers, to individual and inter-social volition, and 6) words related to sentimental and moral powers. Each category has many sub classes. The data that we use in our experiment here are from the

Table 3. Semantically related words at node "Fraud"

Word	invest	score	disappear	score	marry	score	lie	score
	resell	0.782	convince	0.738	divorce	0.981	reveal	0.812
Similar	own	0.773	vanish	0.734	bear	0.933	discover	0.809
Words	collapse	0.762	pose	0.731	widow	0.847	admit	0.745
	trade	0.739	notice	0.723	inherit	0.756	tell	0.724
	promise	0.718	try	0.718	emigrate	0.726	confess	0.701
Word	bury	score	identify	score	examine	score	divorce	score
	marry	0.774	indicate	0.815	prove	0.745	widow	0.842
Similar	burn	0.728	provide	0.758	conclude	0.734	marry	0.826
Words	die	0.715	employ	0.752	verify	0.731	graduate	0.772
	inherit	0.713	report	0.732			remarry	0.742
	survive	0.702	demonstrate	0.709				

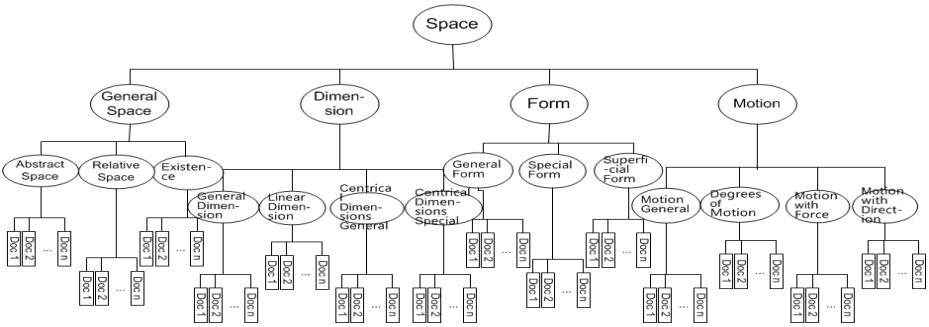


Fig. 2. Hierarchical structure of the "Words related to space" category

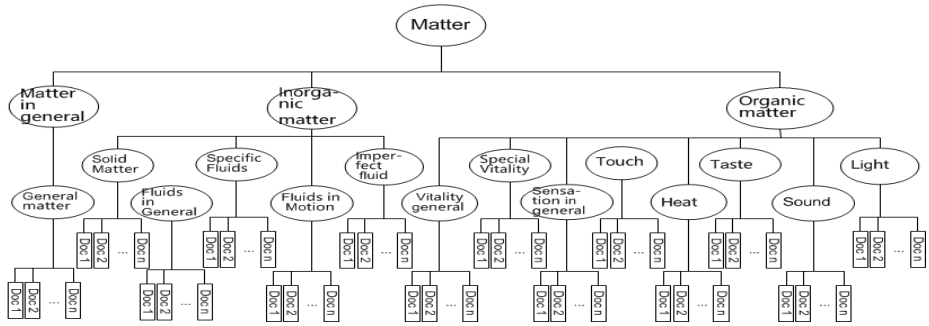


Fig. 3. Hierarchical structure of the "Words related to matter" category

second and third categories which contain 137 and 136 documents, respectively. The hierarchical structures of the data in both categories are shown in figures 2 and 3. 10-fold cross validation is performed on these two categories by randomly

splitting into 10 groups the 9749 and the 7617 words in the vocabularies of the second and third categories, respectively. For each word, we try to find the correct corresponding document to which it belongs using the parameter θ. The probability threshold is set to 0.7. The classification results in terms of accuracy, of our model and the original flat model, are shown in table 4. From this table, we can see that the accuracy scores of the original flat model are 77.23 and 78.04 while for our model, they are 81.76 and 82.17. The improvement is due to the property of our hierarchical model, since we describe the data as a tree structure where the number of classes is reduced at each estimation. Moreover, we performed a significance student t-test, with a confidence level of 95% , on the obtained scores at each cross validation. The results are shown in table 5. According to these results, we can say that the difference in accuracy between our model and the flat one is statistically significant.

Table 4. Results obtained for the word classification task

	Accuracy (%)	
Data	category "words related to space"	category "words related to matter"
Flat Model	77.23	78.04
Our Model	81.76	82.17

Table 5. Statistical significance tests on the accuracy scores

	flat model		our model		t-value	critical t-value ($\alpha = 0.05$)
data	mean	σ	mean	σ		
"words related to space"	77.23	3.67	81.76	2.10	3.39	2.1009
"words related to matter"	78.04	3.28	82.17	2.57	3.13	2.1009

4 Conclusion

We have presented a statistical document model to analyze collections of documents having hierarchical structures. Our model can be viewed as an extension of the flat log-bilinear approach. It has been validated by conducting experiments involving real data gathered from different Web sites. Two main tasks have been considered namely semantically related words extraction and word classification. The obtained results are promising and demonstrate that our model performs well on both tasks. Future potential research works could be devoted to the extension of the model to online settings (e.g. adding, fusing, or deleting nodes) to take into account the dynamic nature of the Web (i.e. new documents are added and others are deleted regularly on the Web). Another promising future work could be dedicated to the consideration of other languages (e.g. French, Arabic, Spanish, Chinese, etc.) for validation purposes.

Acknowledgments. The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Denning, P.J., Denning, D.E.: Discussing cyber attack. Communications of the ACM 53(9), 29–31 (2010)
2. Franklin, J., Paxson, V., Perrig, A., Savage, S.: An inquiry into the nature and causes of the wealth of internet miscreants. In: Proc. of the 14th ACM Conference on Computer and Communications Security (CCS), pp. 375–388. ACM (2007)
3. Sanjay, G.: Cyberwarfare: connecting the dots in cyber intelligence. Communications of the ACM 54(8), 132–140 (2011)
4. Hofmann, T.: Probabilistic latent semantic indexing. In: Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 50–57 (1999)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
6. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America 101, 5228–5235 (2004)
7. Dietz, L., Bickel, S., Scheffer, T.: Unsupervised prediction of citation influences. In: Proc. of the 24th International Conference on Machine Learning (ICML), pp. 233–240. ACM (2007)
8. Mimno, D., McCallum, A.: Mining a digital library for influential authors. In: Proc. of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL), pp. 105–106. ACM (2007)
9. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research, 141–188 (2010)
10. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
11. Lin, D., Pantel, P.: Dirt - discovery of inference rules from text. In: Proc. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 323–328 (2001)
12. Turney, P.D.: Similarity of semantic relations. Computational Linguistics 32(3), 379–416 (2006)
13. Nakov, P.I., Hearst, M.A.: Ucb: System description for semeval task 4. In: Proc. of the Fourth International Workshop on Semantic Evaluations (2007)
14. Dumais, S., Chen, H.: Hierarchical classification of web content. In: Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 256–263. ACM (2000)
15. Zhang, D., Lee, W.S.: Web taxonomy integration using support vector machines. In: Proc. of the 13th International Conference on WWW, pp. 472–481 (2004)
16. Ruiz, M.E., Srinivasan, P.: Hierarchical text categorization using neural networks. Information Retrieval 5(1), 87–118 (2002)
17. Hofmann, T., Cai, L., Ciaramita, M.: Learning with taxonomies: Classifying documents and words. In: Proc. of Synatx, Semantics and Statistics NIPS Workshop (2003)
18. Maas, A., Ng, A.: A probabilistic model for semantic word vectors. In: Proc. of the Deep Learning and Unsupervised Feature Learning Workshop NIPS 2010 (2010)

Anomaly Intrusion Detection Using Incremental Learning of an Infinite Mixture Model with Feature Selection

Wentao Fan[1], Nizar Bouguila[1], and Hassen Sallay[2]

[1] Concordia University, Montreal, QC, Canada
wenta_fa@encs.concordia.ca, nizar.bouguila@concordia.ca
[2] Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia
hmsallay@imamu.edu.sa

Abstract. We propose an incremental nonparametric Bayesian approach for clustering. Our approach is based on a Dirichlet process mixture of generalized Dirichlet (GD) distributions. Unlike classic clustering approaches, our model does not require the number of clusters to be pre-defined. Moreover, an unsupervised feature selection scheme is integrated into the proposed nonparametric framework to improve clustering performance. By learning the proposed model using an incremental variational framework, the number of clusters as well as the features weights can be automatically and simultaneously computed. The effectiveness and merits of the proposed approach are investigated on a challenging application namely anomaly intrusion detection.

Keywords: Mixture models, clustering, Dirichlet process, generalized Dirichlet, feature selection, variational inference, intrusion detection.

1 Introduction

Huge volumes of data are routinely generated by organizations, scientific activities, internet traffic and so on. An important problem is to model these data to improve the process of making automatic decisions [12]. A widely used approach for data modeling and knowledge discovery is clustering. Clustering can be defined as the task of partitioning a given data set $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ containing N vectors into M homogenous clusters $\mathcal{C}_1, \ldots, \mathcal{C}_M$ such that $\mathcal{C}_j \cap \mathcal{C}_l = \emptyset$, and $\cup_{j=1}^{M} \mathcal{C}_j = \mathcal{X}$. Finite mixture models have been widely applied for clustering during the last two decades [11]. Within finite mixture modeling, selecting the number of components that best describes the underlying data without over- or under-fitting is one of the most challenging problems. This obstacle can be removed by extending finite mixtures to the infinite case through Dirichlet processes [13]. Infinite mixtures allow a natural approach for data clustering. Unlike finite mixtures, the number of clusters does not need to be specified by the practitioner in advance and can be automatically inferred from the dataset. Several approaches have been proposed to learn mixture models. In particular, variational inference has received a lot of attention recently [5,4,1,6]. Variational

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 364–373, 2013.

inference is a deterministic approximation learning technique that only requires a modest amount of computational power in contrast to other well-developed approaches such as Markov chain Monte Carlo (MCMC) techniques, and has a tractable learning process as well. Generally real-world problems involve dynamic data sets where the volume of data continuously grows. Thus, it is crucial to adopt an incremental way to learn the statistical model used for clustering.

In this paper, we adopt an incremental version of variational inference proposed by [7] to learn infinite generalized Dirichlet (GD) mixtures with unsupervised feature selection. The employment of the GD as the basic distribution in our mixture model is motivated by its favorable performance when dealing with non-Gaussian data [2,3]. The advantages of our framework are summarized as following: First, the difficulty of choosing the appropriate number of components is avoided by assuming that there is an infinite number of components. Second, thanks to its incremental nature, it is very efficient when dealing with sequentially arriving data, which is an important factor for real-time applications. Third, within the proposed framework, the model parameters and features saliencies can be estimated simultaneously and automatically. The effectiveness of our approach is illustrated through a challenging task namely anomaly intorsion detection. The rest of this paper is organized as follows. Section 2 reviews briefly the infinite GD mixture model with unsupervised feature selection. In Section 3, we develop an incremental variational inference framework for model learning. Section 4 is devoted to the experimental results. Finally, conclusion follows in Section 5.

2 Infinite GD Mixture Model with Feature Selection

In this section, we review briefly the infinite generalized Dirichlet (GD) mixture model with feature selection, which is constructed using a stick-breaking Dirichlet process framework. If a D-dimensional random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_D)$ is sampled from a mixture of GD distributions with infinite number of components:

$$p(\boldsymbol{Y}|\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^{\infty} \pi_j \mathrm{GD}(\boldsymbol{Y}|\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j) \tag{1}$$

where $\boldsymbol{\pi}$ represents the mixing coefficients with the constraints that are positive and sum to one. Here we adopt the Dirichlet process framework with a stick-breaking representation [15], where the mixing coefficients $\{\pi_j\}$ are constructed by recursively breaking a unit length stick into an infinite number of pieces as $\pi_j = \lambda_j \prod_{k=1}^{j-1}(1 - \lambda_k)$. The stick breaking variable λ_j is distributed according to $\lambda_j \sim \mathrm{Beta}(1, \zeta)$, where ζ is a positive real number and is the concentration parameter of the Dirichlet process. In Eq. (1), $\boldsymbol{\alpha}_j = (\alpha_{j1}, \ldots, \alpha_{jD})$ and $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jD})$ are the positive parameters of the GD distribution $\mathrm{GD}(\boldsymbol{Y}|\boldsymbol{\alpha}_j, \boldsymbol{\beta})$ associated with component j, where $\mathrm{GD}(\boldsymbol{X}|\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j)$ is given by

$$\mathrm{GD}(\boldsymbol{Y}|\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j) = \prod_{l=1}^{D} \frac{\Gamma(\alpha_{jl} + \beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} Y_l^{\alpha_{jl}-1} \left(1 - \sum_{k=1}^{l} Y_k\right)^{\gamma_{jl}} \tag{2}$$

where $\sum_{l=1}^{D} Y_l < 1$ and $0 < y_l < 1$ for $l = 1, \ldots, D$, $\gamma_{jl} = \beta_{jl} - \alpha_{jl+1} - \beta_{jl+1}$ for $l = 1, \ldots, D-1$, and $\gamma_{jD} = \beta_{jD} - 1$. $\Gamma(\cdot)$ is the gamma function defined by $\Gamma(x) = \int_0^{\infty} u^{x-1} e^{-u} du$. It is noteworthy that, in practice the features $\{Y_l\}$ are generally not equally significant for the clustering task since some features may be"noise" and do not contribute to clustering process. Therefore, feature selection may act as a crucial role to improve the learning performance. Before incorporating feature selection into our framework, we leverage a handy mathematical property of the GD distribution which is introduced in [3], to transform the original data points into another D-dimensional space with independent features. Then, we can rewrite the infinite GD mixture model as

$$p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^{\infty} \pi_j \prod_{l=1}^{D} \text{Beta}(X_l|\alpha_{jl}, \beta_{jl}) \tag{3}$$

where $X_l = Y_l$ and $X_l = Y_l/(1 - \sum_{k=1}^{l-1} Y_k)$ for $l > 1$. $\text{Beta}(X_l|\alpha_{jl}, \beta_{jl})$ is a Beta distribution parameterized with $(\alpha_{jl}, \beta_{jl})$. Accordingly, the independence between the features in the new space becomes a fact rather than an assumption as considered in previous approaches [8,4]. In this work, we adopt an unsupervised feature selection scheme suggested in [8]: the lth feature is irrelevant if its distribution is independent of the class labels, that is, if it follows a common density. Thus, we can rewrite the mixture density in Eq. (3) as

$$p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\phi}, \boldsymbol{\sigma}, \boldsymbol{\tau}) = \sum_{j=1}^{\infty} \prod_{l=1}^{D} \left[\text{Beta}(X_l|\alpha_{jl}, \beta_{jl})\right]^{\phi_l} \left[\text{Beta}(X_l|\sigma_l, \tau_l)\right]^{1-\phi_l} \tag{4}$$

where $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_D)$ is a set of binary parameters and known as the feature relevance indicator, such that $\phi_l = 0$ if feature l is irrelevant (i.e. noise) and follows a Beta distribution: $\text{Beta}(X_l|\sigma_l, \tau_l)$. The prior of $\boldsymbol{\phi}$ is defined as:

$$p(\boldsymbol{\phi}|\boldsymbol{\epsilon}) = \prod_{l=1}^{D} \epsilon_{l_1}^{\phi_l} \epsilon_{l_2}^{1-\phi_l} \tag{5}$$

where each ϕ_l is a Bernoulli variable such that $p(\phi_l = 1) = \epsilon_{l_1}$ and $p(\phi_l = 0) = \epsilon_{l_2}$. Here the vector $\boldsymbol{\epsilon}$ denotes the features saliencies (i.e. the probabilities that the features are relevant) where $\boldsymbol{\epsilon}_l = (\epsilon_{l_1}, \epsilon_{l_2})$ and $\epsilon_{l_1} + \epsilon_{l_2} = 1$. Furthermore, we place a Dirichlet prior $\text{Dir}(\cdot)$ over $\boldsymbol{\epsilon}$ with positive parameter $\boldsymbol{\varphi}$ as: $p(\boldsymbol{\epsilon}) = \prod_{l=1}^{D} \text{Dir}(\boldsymbol{\epsilon}_l|\boldsymbol{\varphi})$. In mixture modeling, it is convenient to introduce a variable $\boldsymbol{Z} = (Z_1, \ldots, Z_N)$ for an observed dataset $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N)$, where Z_i is an assignment variable of the mixture component with which the data point $\boldsymbol{X}_i$ is associated. The marginal distribution over $\boldsymbol{Z}$ is given by

$$p(\boldsymbol{Z}|\boldsymbol{\lambda}) = \prod_{i=1}^{N} \prod_{j=1}^{\infty} \left[\lambda_j \prod_{k=1}^{j-1} (1 - \lambda_k)\right]^{\mathbf{1}[Z_i = j]} \tag{6}$$

where $\mathbf{1}[\cdot]$ is an indicator function which has the value of 1 when $Z_i = j$ and 0 otherwise. Next, we need to introduce prior distributions over unknown random

variables. In this work, the Gamma distribution $\mathcal{G}(\cdot)$ is adopted to approximate a conjugate prior over parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}$ and $\boldsymbol{\tau}$, by assuming that these Beta parameters are statistically independent: $p(\boldsymbol{\alpha}) = \mathcal{G}(\boldsymbol{\alpha}|\boldsymbol{u}, \boldsymbol{v})$, $p(\boldsymbol{\beta}) = \mathcal{G}(\boldsymbol{\beta}|\boldsymbol{p}, \boldsymbol{q})$, $p(\boldsymbol{\sigma}) = \mathcal{G}(\boldsymbol{\sigma}|\boldsymbol{g}, \boldsymbol{h})$, $p(\boldsymbol{\tau}) = \mathcal{G}(\boldsymbol{\tau}|\boldsymbol{s}, \boldsymbol{t})$.

3 Incremental Variational Model Learning

In this work, we adopt a variational incremental learning approach introduced in [7] to learn the proposed model. According to this approach, data instances can be sequentially processed in small batches where each one may contain one or more data points. There are two phases involved: a model building phase and a compression phase. In the model building phase, the current model with observed data points is optimized. The goal of the compression phase is to determine which mixture component that groups of data points should be assigned to.

3.1 Model Building Phase

Given an observed dataset $\mathcal{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N)$, let $\Theta = \{\boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\phi}, \boldsymbol{\epsilon}, \boldsymbol{\lambda}\}$ be the set of random variables. In variational learning, the main goal is to determine a proper approximation $q(\Theta)$ for the real posterior distribution $p(\Theta|\mathcal{X})$ by maximizing the free energy $\mathcal{F}(\mathcal{X}, q)$:

$$\mathcal{F}(\mathcal{X}, q) = \int q(\Theta) \ln[p(\mathcal{X}, \Theta)/q(\Theta)] d\Theta \tag{7}$$

In our framework, motivated by [1], we truncate the variational distribution $q(\Theta)$ at a value of M, such that $\lambda_M = 1$, $\pi_j = 0$ when $j > M$, and $\sum_{j=1}^{M} \pi_j = 1$. It is noteworthy that the truncation level M is a variational parameter which can be freely initialized and will be optimized automatically during the learning process [1]. Next, we adopt a factorization assumption to factorize $q(\Theta)$ into disjoint tractable factors as: $q(\Theta) = q(\boldsymbol{Z})q(\boldsymbol{\alpha})q(\boldsymbol{\beta})q(\boldsymbol{\sigma})q(\boldsymbol{\tau})q(\boldsymbol{\phi})q(\boldsymbol{\epsilon})q(\boldsymbol{\lambda})$. Then, we can obtain the following update equations for these factors by maximizing the free energy $\mathcal{F}(\mathcal{X}, q)$ with respect to each of them:

$$q(\boldsymbol{Z}) = \prod_{i=1}^{N} \prod_{j=1}^{M} r_{ij}^{\mathbf{1}[Z_i=j]}, \qquad q(\boldsymbol{\lambda}) = \prod_{j=1}^{M} \text{Beta}(\lambda_j|a_j, b_j) \tag{8}$$

$$q(\boldsymbol{\alpha}) = \prod_{j=1}^{M} \prod_{l=1}^{D} \mathcal{G}(\alpha_{jl}|u_{jl}^*, v_{jl}^*), \qquad q(\boldsymbol{\beta}) = \prod_{j=1}^{M} \prod_{l=1}^{D} \mathcal{G}(\beta_{jl}|c_{jl}^*, d_{jl}^*) \tag{9}$$

$$q(\boldsymbol{\sigma}) = \prod_{l=1}^{D} \mathcal{G}(\sigma_l|g_l^*, h_l^*), \qquad q(\boldsymbol{\tau}) = \prod_{l=1}^{D} \mathcal{G}(\tau_l|s_l^*, t_l^*) \tag{10}$$

$$q(\boldsymbol{\phi}) = \prod_{i=1}^{N} \prod_{l=1}^{D} f_{il}^{\phi_{il}} (1 - f_{il})^{(1-\phi_{il})}, \qquad q(\boldsymbol{\epsilon}) = \prod_{l=1}^{D} \text{Dir}(\boldsymbol{\epsilon}_l|\boldsymbol{\varphi}_l^*) \tag{11}$$

where we have calculated

$$r_{ij} = \frac{\exp(\rho_{ij})}{\sum_{j=1}^{M}\exp(\rho_{ij})}, \qquad \varphi_{l_1}^* = \varphi_{l_1} + \sum_{i=1}^{N}\langle\phi_{il}\rangle, \qquad \varphi_{l_2}^* = \varphi_{l_2} + \sum_{i=1}^{N}\langle 1-\phi_{il}\rangle \tag{12}$$

$$\rho_{ij} = \sum_{l=1}^{D}\langle\phi_{il}\rangle[\widetilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl}-1)\ln X_{il} + (\bar{\beta}_{jl}-1)\ln(1-X_{il})] + \big\langle\ln\lambda_j\big\rangle + \sum_{k=1}^{j-1}\big\langle\ln(1-\lambda_k)\big\rangle$$

$$f_{il} = \frac{\exp(\widetilde{f}_{il})}{\exp(\widetilde{f}_{il})+\exp(\widehat{f}_{il})}, \qquad a_j = 1 + \sum_{i=1}^{N}\langle Z_i = j\rangle, \qquad b_j = \zeta_j + \sum_{i=1}^{N}\sum_{k=j+1}^{M}\langle Z_i = k\rangle$$

$$\widetilde{f}_{il} = \sum_{j=1}^{M}\big\langle Z_i = j\big\rangle\big[\widetilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl}-1)\ln X_{il} + (\bar{\beta}_{jl}-1)\ln(1-X_{il})\big] + \langle\ln\epsilon_{l_1}\rangle$$

$$\widehat{f}_{il} = \widetilde{\mathcal{F}}_l + (\bar{\sigma}_l - 1)\ln X_{il} + (\bar{\tau}_l - 1)\ln(1-X_{il}) + \langle\ln\epsilon_{l_2}\rangle$$

$$u_{jl}^* = u_{jl} + \sum_{i=1}^{N} r_{ij}\langle\phi_{il}\rangle\bar{\alpha}_{jl}\big[\psi(\bar{\alpha}_{jl}+\bar{\beta}_{jl}) - \psi(\bar{\alpha}_{jl}) + \bar{\beta}_{jl}\psi'(\bar{\alpha}_{jl}+\bar{\beta}_{jl})(\langle\ln\beta_{jl}\rangle - \ln\bar{\beta}_{jl})\big]$$

$$c_{jl}^* = c_{jl} + \sum_{i=1}^{N} r_{ij}\langle\phi_{il}\rangle\bar{\beta}_{jl}[\psi(\bar{\alpha}_{jl}+\bar{\beta}_{jl}) - \psi(\bar{\beta}_{jl}) + \bar{\alpha}_{jl}\psi'(\bar{\alpha}_{jl}+\bar{\beta}_{jl})(\big\langle\ln\alpha_{jl}\big\rangle - \ln\bar{\alpha}_{jl})]$$

$$v_{jl}^* = v_{jl} - \sum_{i=1}^{N}\big\langle Z_i = j\big\rangle\big\langle\phi_{il}\big\rangle\ln X_{il}, \qquad d_{jl}^* = d_{jl} - \sum_{i=1}^{N}\big\langle Z_i = j\big\rangle\big\langle\phi_{il}\big\rangle\ln(1-X_{il})$$

In the above equations, $\psi(\cdot)$ is the digamma function, and $\langle\cdot\rangle$ is the expectation evaluation. $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{F}}$ are the lower bounds of $\mathcal{R} = \big\langle\ln\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\big\rangle$ and $\mathcal{F} = \big\langle\ln\frac{\Gamma(\sigma+\tau)}{\Gamma(\sigma)\Gamma(\tau)}\big\rangle$, respectively. Since these expectations are intractable, we use the second-order Taylor series expansion to find their lower bounds. The hyperparameters of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are calculated in a similar way as for the hyperparameters of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The expected values in the above formulas are given by $\langle Z_i = j\rangle = r_{ij}$, $\langle\phi_{il}\rangle = f_{il}$, $\bar{\alpha}_{jl} = \langle\alpha_{jl}\rangle = u_{jl}^*/v_{jl}^*$, $\bar{\beta}_{jl} = c_{jl}^*/d_{jl}^*$,$\big\langle\ln\lambda_j\big\rangle = \Psi(a_j) - \Psi(a_j+b_j)$, $\big\langle\ln(1-\lambda_j)\big\rangle = \Psi(b_j) - \Psi(a_j+b_j)$, $\langle\ln\epsilon_{l_1}\rangle = \psi(\varphi_1^*) - \psi(\varphi_1^*+\varphi_2^*)$, $\big\langle\ln\alpha_{jl}\big\rangle = \Psi(u_{jl}^*) - \ln v_{jl}^*$, and $\big\langle\ln\beta_{jl}\big\rangle = \Psi(c_{jl}^*) - \ln d_{jl}^*$.

After convergence, the observed data points are clustered into M groups according to corresponding responsibilities r_{ij}. Following [7], these newly formed groups of data points are denoted as "clumps", and these clumps are subject to the constraint that all data points $\boldsymbol{X}_i$ in the clump m share the same $q(Z_i) \equiv q(Z_m)$ which is a key factor in the following compression phase.

3.2 Compression Phase

In the compression phase, we attempt to determine clumps that possibly belong to the same mixture component while taking into account future arriving data. Suppose that we have already observed N data points, and our goal is to make an inference at some target time T where $T \geq N$. This is fulfilled by scaling the current observed data to the target size T, which is equivalent to using the variational posterior distribution of the observed data N as a predictive model

of the future data [7]. Therefore, we can obtain the modified free energy for the compression phase as the following

$$\mathcal{F} = \sum_{j=1}^{M}\sum_{l=1}^{D}\left[\left\langle \ln \frac{p(\alpha_{jl})}{q(\alpha_{jl})}\right\rangle + \left\langle \ln \frac{p(\beta_{jl})}{q(\beta_{jl})}\right\rangle\right] + \sum_{l=1}^{D}\left[\left\langle \ln \frac{p(\sigma_l)}{q(\sigma_l)}\right\rangle + \left\langle \ln \frac{p(\tau_l)}{q(\tau_l)}\right\rangle + \left\langle \ln \frac{p(\boldsymbol{\epsilon}_l)}{q(\boldsymbol{\epsilon}_l)}\right\rangle\right]$$
$$+ \sum_{j=1}^{M}\left\langle \ln \frac{p(\lambda_j)}{q(\lambda_j)}\right\rangle + \frac{T}{N}\sum_{m}|n_m|\Big[\ln \sum_{j=1}^{M}\exp(\rho_{mj}) + \ln \sum_{l=1}^{D}\exp(f_{ml})\Big] \tag{13}$$

where $\frac{T}{N}$ is the data magnification factor and $|n_m|$ represents the number of data points in clump m. The corresponding update equations for maximizing this free energy function are

$$r_{mj} = \frac{\exp(\rho_{mj})}{\sum_{j=1}^{M}\exp(\rho_{mj})}, \qquad f_{ml} = \frac{\exp(\widetilde{f}_{ml})}{\exp(\widetilde{f}_{ml}) + \exp(\widehat{f}_{ml})}, \qquad \vartheta = \frac{T}{N}\sum_{m}|n_m| \tag{14}$$

$$\varphi^*_{l_1} = \varphi_{l_1} + \frac{T}{N}\sum_{m}|n_m|\langle\phi_{ml}\rangle, \qquad \varphi^*_{l_2} = \varphi_{l_2} + \frac{T}{N}\sum_{m}|n_m|\langle 1-\phi_{ml}\rangle$$

$$\rho_{mj} = \sum_{l=1}^{D} f_{ml}[\widetilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl}-1)\ln X_{ml} + (\bar{\beta}_{jl}-1)\ln(1-X_{ml})] + \langle\ln\lambda_j\rangle + \sum_{k=1}^{j-1}\langle\ln(1-\lambda_k)\rangle$$

$$a_j = 1 + \vartheta\langle Z_m = j\rangle, \qquad b_j = \zeta_j + \vartheta\sum_{k=j+1}^{M}\langle Z_m = k\rangle$$

$$\widetilde{f}_{ml} = \sum_{j=1}^{M}\langle Z_m = j\rangle\left[\widetilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl}-1)\ln X_{ml} + (\bar{\beta}_{jl}-1)\ln(1-X_{ml})\right] + \langle\ln\epsilon_{l_1}\rangle$$

$$\widehat{f}_{ml} = \widetilde{\mathcal{F}}_l + (\bar{\sigma}_l - 1)\ln X_{ml} + (\bar{\tau}_l - 1)\ln(1-X_{ml}) + \langle\ln\epsilon_{l_2}\rangle$$

$$u^*_{jl} = u_{jl} + \vartheta r_{mj}\langle\phi_{ml}\rangle\bar{\alpha}_{jl}\left[\psi(\bar{\alpha}_{jl}+\bar{\beta}_{jl}) - \psi(\bar{\alpha}_{jl}) + \bar{\beta}_{jl}\psi'(\bar{\alpha}_{jl}+\bar{\beta}_{jl})(\langle\ln\beta_{jl}\rangle - \ln\bar{\beta}_{jl})\right]$$

$$c^*_{jl} = c_{jl} + \vartheta r_{mj}\langle\phi_{ml}\rangle\bar{\beta}_{jl}[\psi(\bar{\alpha}_{jl}+\bar{\beta}_{jl}) - \psi(\bar{\beta}_{jl}) + \bar{\alpha}_{jl}\psi'(\bar{\alpha}_{jl}+\bar{\beta}_{jl})(\langle\ln\alpha_{jl}\rangle - \ln\bar{\alpha}_{jl})]$$

$$v^*_{jl} = v_{jl} - \vartheta r_{mj}\langle\phi_{ml}\rangle\ln X_{ml}, \qquad d^*_{jl} = d_{jl} - \vartheta r_{mj}\langle\phi_{ml}\rangle\ln(1-X_{ml})$$

where $\langle X_{ml}\rangle$ represents the average over all data points contained in clump m. In the compression phase, the first step is to hard assign each clump or data point to the component with the highest responsibility r_{mj} obtained from the model building phase as

$$I_m = \arg\max_j r_{mj} \tag{15}$$

where $\{I_m\}$ represent which component the clump (or data point) m belongs to in the compression phase. Next, we cycle through each component and split it into two subcomponents along its principal component. This splitting process can be refined by updating Eqs. (14). After convergence criterion is reached for refining the split, the clumps are then assigned to one of the two candidate components. Among all the potential splits, we choose the one that results in the largest change in the free energy (Eq. (13)). We iterate this splitting process until a stopping criterion is satisfied. Based on [7], a stoping criterion for the splitting process can be expressed as a limit on the amount of memory required to store the components. In our case, the memory cost for the mixture model is $\mathcal{MC} = 5DN_c$, where $5D$ is the number of parameters contained in a D-variate GD component with feature selection, while N_m denotes the number of components. Thus, We

Algorithm 1

Choose the initial truncation level M.
Initialize hyper-parameters: u_{jl}, v_{jl}, c_{jl}, d_{jl}, g_l, h_l, s_l, t_l, ζ_j, φ_{l_1} and φ_{l_2}.
Initialize the values of r_{ij} by K-Means algorithm.
while More data to be observed **do**
  Perform the model building phase through Eqs. (8)$\sim$(11).
  Initialize the compression phase using Eq. (15).
  while $\mathcal{MC} \geq \mathcal{C}$ **do**
    for $j = 1$ **to** M **do**
      if $evaluated(j) =$ **false then**
        Split component j and refine this split using Eqs (14).
        $\Delta\mathcal{F}(j) =$ change in Eq. (13).
        $evaluated(j) =$ **true**.
      end if
    end for
    Split component j with the largest value of $\Delta\mathcal{F}(j)$.
    $M = M + 1$.
  end while
  Discard the currently observed data points.
  Save the resultant components for next learning round.
end while

can define an upper limit on the component memory cost $\mathcal{C}$, and the compression phase stops when $\mathcal{MC} \geq \mathcal{C}$. As a result, the computational time and the space requirement is bounded in each learning round. After the compression phase, the currently observed data points are discarded while the resultant components are treated in the same way as data points in the next round of leaning. The proposed incremental variational inference algorithm for infinite GD mixture model with feature selection is summarized in Algorithm 1.

4 Anomaly Intrusion Detection

The construction of intrusion detection models has been the topic of extensive research in the past. The main goal is to protect networks against criminals. Indeed, the target of Intrusion Detection Systems (IDSs) is to discover inappropriate, incorrect, or anomalous activities within computers or networks and this can be considered as classification problem in the context of machine learning (see, [9], for instance and references therein). In general, IDSs can be broadly divided into two main categories: misuse detection and anomaly detection systems [14]. In contrast to the signature-based misuse detection, the anomaly detection has the superiority of being able to detect new or unknown attacks. In this experiment, we evaluate the effectiveness of the proposed incremental infinite GD mixture model with feature selection (referred as *InGD-Fs*) by applying it to tackle the problem of anomaly intrusion detection. In our case, the truncation level M is initialized as 20. Our specific choice for the ini-

tial values of the hyperparameters is: $(u_{jl}, v_{jl}, c_{jl}, d_{jl}, g_l, h_l, s_l, t_l, \zeta_j, \varphi_{l_1}, \varphi_{l_2}) = (1, 0.01, 1, 0.01, 1, 0.01, 1, 0.01, 0.1, 0.1, 0.1)$.

4.1 Databases and Experimental Design

We investigate our approach on two challenging publicly available databases known as the KDD Cup 1999 Data[1] and the Kyoto traffic Data[2]. In our case, a 10 percent subset of the KDD database is adopted. Specifically, the training set consists of 494,020 data instances of which 97,277 are normal and 396,743 are attacks, while the testing set contains 292,393 data instances of which 60,593 are normal and 231,800 are attacks. Each instance in this data set is composed of 41 features. This database has five categories in total including one 'Normal' and four attack classes namely: DOS, R2L, U2R and Probing. The Kyoto database consists of real traffic data obtained from several types of honeypots by the Kyoto University. In our experiment, the Kyoto database contains 784,000 21-dimensional instances where 395,368 are normal sessions and 388,632 are attacks. In this application, the training data are used to learn the current model, where the testing data instances are supposed to be obtained sequentially in an online fashion. It is noteworthy that the features in the two original databases are on quite different scales, we therefore require to normalize the databases so that one feature would not dominate the others. By finding the maximum and minimum values of a given feature X_l in a data instance $\boldsymbol{X}$, we can transform the feature into the range $[0, 1]$ by $X_l = \frac{X_l - \min(X_l)}{\max(X_l) - \min(X_l)}$, where X_l is set to a smallest value if the maximum is equal to the minimum.

4.2 Experimental Results

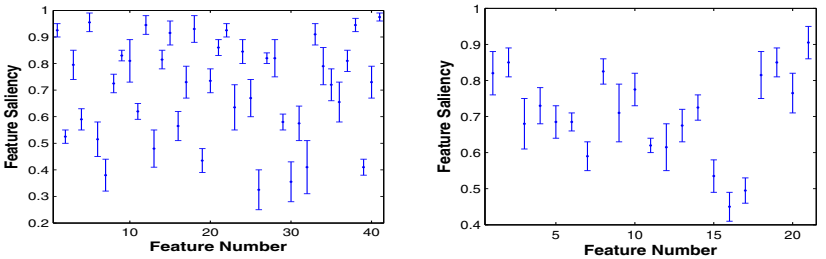
We run the the proposed *InGD-Fs* 20 times to investigate its performance. For comparison, we have also applied three other mixture-modeling approaches: the infinite GD mixture model without feature selection (*InGD*), the finite GD mixture model without feature selection(*FiGD*) and the infinite Gaussian mixture model with feature selection (*InGM-Fs*). In order to provide a fair comparison, all of these tested approaches are learned through the incremental variational inference. The results of applying different approaches on the KDD99 database and Kyoto database are shown in Table 1, in terms of the average classification accuracy rate (Accuracy) and the false positive (FP) rate. According to this table, it is obvious that our approach (*InGD-Fs*) has the best performance among all the tested approaches by providing the highest accuracy rate and the lowest FP rate for both databases. There are several important conclusions which can be drawn from this table: First, the fact that *InGD-Fs* outperforms *InGD* proves that feature selection is a significant factor for improving clustering performance; Second, *InGD* has better results than *FiGD*, which demonstrates

[1] `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`
[2] `http://www.takakura.com/Kyoto_data/`

the advantage of using infinite mixture models over finite ones. Third, *InGM-Fs* provides the worst performance among all tested approaches which verifies that the GD mixture model has better modeling capability than the Gaussian for compactly supported data. In addition, the saliencies of the 41 features in the KDD 99 database and 21 features in the Kyoto database calculated by the *InGD-Fs* over 20 runs are illustrated in Fig. 1. As shown in this figure, it is clear that the different features do not contribute equally in the classification, since they have different relevance degrees.

Table 1. Average classification accuracy rate (Accuracy) and false positive (FP) rate computed using different approaches

	KDD data		Kyoto data	
	Accuracy (%)	FP (%)	Accuracy (%)	FP (%)
InGD-Fs	86.73	6.27	81.34	11.78
InGD	84.18	7.14	78.61	13.52
FiGD	82.52	9.63	76.59	16.37
InGM-Fs	79.45	13.91	75.01	18.23



KDD data Kyoto data

Fig. 1. Features saliencies obtained using the proposed *InGD-Fs* approach

5 Conclusion

In this paper, we have proposed an incremental clustering algorithm that allows the simultaneous computation of the number of clusters and features weights during execution. Our approach is based on an incremental variational learning of the infinite GD mixture model with unsupervised feature selection. The effectiveness of the proposed approach has been evaluated on a challenging real application namely anomaly intrusion detection. Future works could be devoted to the inclusion of a localized feature selection scheme, such as the one proposed in [10], to improve further the generalization capabilities of our framework.

Acknowledgments. The first two authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC). The third author would like to thank King Abdulaziz City for Science and Technology (KACST) for their funding support under grant 08-INF36-8.

References

1. Blei, D., Jordan, M.: Variational inference for Dirichlet process mixtures. Bayesian Analysis 1, 121–144 (2005)
2. Bouguila, N., Ziou, D.: A hybrid SEM algorithm for high-dimensional unsupervised learning using a finite generalized Dirichlet mixture. IEEE Transactions on Image Processing 15(9), 2657–2668 (2006)
3. Boutemedjet, S., Bouguila, N., Ziou, D.: A hybrid feature extraction selection approach for high-dimensional non-Gaussian data clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(8), 1429–1443 (2009)
4. Constantinopoulos, C., Titsias, M., Likas, A.: Bayesian feature and model selection for Gaussian mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(6), 1013–1018 (2006)
5. Corduneanu, A., Bishop, C.M.: Variational Bayesian model selection for mixture distributions. In: Proc. of the 8th International Conference on Artificial Intelligence and Statistics (AISTAT), pp. 27–34 (2001)
6. Fan, W., Bouguila, N., Ziou, D.: Variational learning for finite Dirichlet mixture models and applications. IEEE Transactions on Neural Netw. Learning Syst. 23(5), 762–774 (2012)
7. Gomes, R., Welling, M., Perona, P.: Incremental learning of nonparametric Bayesian mixture models. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
8. Law, M.H.C., Figueiredo, M.A.T., Jain, A.K.: Simultaneous feature selection and clustering using mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1154–1166 (2004)
9. Lee, W., Stolfo, S.J., Mok, K.W.: Adaptive intrusion detection: A data mining approach. Artificial Intelligence Review 14(6), 533–567 (2000)
10. Li, Y., Dong, M., Hua, J.: Simultaneous localized feature selection and model detection for Gaussian mixtures. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 953–960 (2009)
11. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
12. Mitchell, T.M.: Machine learning and data mining. Communications of the ACM 42(11), 30–36 (1999)
13. Neal, R.M.: Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics 9(2), 249–265 (2000)
14. Northcutt, S., Novak, J.: Network Intrusion Detection: An Analyst's Handbook. New Riders Publishing (2002)
15. Sethuraman, J.: A constructive definition of Dirichlet priors. Statistica Sinica 4, 639–650 (1994)

Hybridizing Meta-heuristics Approaches for Solving University Course Timetabling Problems

Khalid Shaker[1,*], Salwani Abdullah[2], Arwa Alqudsi[2], and Hamid Jalab[3]

[1] Collage of Computer, University of Anba, Ramadi, Iraq
khalidalhity@gmail.com
[2] Center of Artificial Intelligence Technology,
Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia
{salwani,arwa}@ftsm.ukm.my
[3] Multimedia Unit, Faculty of Computer Science and Information Technology,
Universiti Malaya
50603 Kuala Lumpur, Malaysia
hamidjalab@um.edu.my

Abstract. In this paper we have presented a combination of two meta-heuristics, namely great deluge and tabu search, for solving the university course timetabling problem. This problem occurs during the assignment of a set of courses to specific timeslots and rooms within a working week and subject to a variety of hard and soft constraints. Essentially a set of hard constraints must be satisfied in order to obtain a feasible solution and satisfying as many as of the soft constraints as possible. The algorithm is tested over two databases: eleven enrolment-based benchmark datasets (representing one large, five medium and five small problems) and curriculum-based datasets used and developed from the International Timetabling Competition, ITC2007 (UD2 problems). A new strategy has been introduced to control the application of a set of neighbourhood structures using the tabu search and great deluge. The results demonstrate that our approach is able to produce solutions that have lower penalties on all the small and medium problems in eleven enrolment-based datasets and can produce solutions with comparable results on the curriculum-based datasets (with lower penalties on several data instances) when compared against other techniques from the literature.

Keywords: Great Deluge, Tabu Search, Course Timetabling.

1 Introduction

In the timetabling literature, significant attention has been paid to the problem of constructing university course timetables. Various techniques have been applied to this complex and difficult problem. However, optimum solutions are only possible for problems of a limited size [6]. In this paper, a combination of two meta-heuristic based techniques i.e. great deluge and tabu search are applied to the university course timetabling problem. The approach is tested over eleven benchmark datasets that were introduced by [17] and twenty eight ITC2007 datasets as described in the 2nd

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 374–384, 2013.

International Timetabling Competition (ITC2007), [10]. The results demonstrate that our approach is capable of producing high quality solutions when compared to other techniques in the literature.

2 Problems Descriptions

2.1 Enrolment-Based Course Timetabling Problem

In university course timetabling, a set of courses are scheduled into a given number of rooms and timeslots across a period of time. This usually takes place within a week and the resultant timetable replicated for as many weeks as the courses run. Also, students and teachers are assigned to courses so that the teaching delivery activities can take place. The course timetabling problem is subject to a variety of hard and soft constraints. Hard constraints need to be satisfied in order to produce a feasible solution. In this paper, the 1st experiment is testing our approach on the problem instances introduced by [17] who present the following hard constraints: *No student can be assigned to more than one course at the same time* (**Hard1**). *The room should satisfy the features required by the course* (**Hard2**). *The number of students attending the course should be less than or equal to the capacity of the room* (**Hard3**). *Not more than one course is allowed to be assigned to a timeslot in each room* (**Hard4**).

[17] also present the following soft constraints that are equally penalized: *A student has a course scheduled in the last timeslot of the day* (**Soft1**). *A student has more than 2 consecutive courses* (**Soft2**). *A student has a single course on a day* (**Soft3**).

The problem has:

- A set of n courses, $E = \{e_0, e_1,\ldots,e_{n-1}\}$, 45 timeslots, $T = \{t_0, t_1,\ldots,t_{44}\}$, A set of m rooms, $R = \{r_0, r_1,\ldots,r_{m-1}\}$, A set of q room features, $F = \{f_0,\ldots, f_{q-1}\}$ and A set of v students $S = \{s_0, s_1,\ldots,s_{v-1}\}$.

The objective of this problem is to satisfy the hard constraints and to minimise the violation of the soft constraints. The formula represents the objective function for this problem given as below:

$$min \sum_{i=1}^{v} \boldsymbol{Soft_1 + Soft_2 + Soft_3}$$

We have evaluated our results on the instances taken from [17], (available at http://iridia.ulb.ac.be/~msampels/tt.data/). They are divided into three categories: small, medium and large. We deal with 11 instances: 5 small, 5 medium and 1 large.

2.2 Curriculum-Based Course Timetabling Problem

In the Curriculum-based timetabling problem real datasets will be used from the University of Udine, Italy. The problem consists of the weekly scheduling of lectures for several university courses within a given number of rooms and time periods, where conflicts between courses are set according to the curricula of the university. All the details, updates and news about the problem can be obtained via the website (http://tabu.diegm.uniud.it/ctt/index.php).The following hard and soft constraints, are

presented: **Hard constraints;** (1) ***Lectures.*** All lectures of a course must be scheduled, and they must be assigned to distinct periods. (2) ***Conflicts.*** Lectures of courses in the same curriculum or taught by the same teacher must all be scheduled in different periods. (3) ***Room Occupancy***. Two lectures cannot take place in the same room in the same period. (4) ***Availability***. The teacher of the course must be available to teach that course at a given period; otherwise no lecture of the course can be scheduled at that period.

Soft constraints: (1) ***Room Capacity.*** The number of students attending the course should be less than or equal to the capacity of the room. (2) ***Minimum Working Days***. The lectures of each course must be spread into the given minimum number of days. (3) ***Isolated Lectures.*** Lectures belonging to a curriculum should be in consecutive periods. (4) ***Room Stability.*** All lectures of a course *should* be given in the same room.

The main contribution of this work consists of a combination of the Great Deluge algorithm with a Tabu Search approach in solving the university course timetabling problem through fully satisfying the hard constraints and minimising as much as possible the violation of the soft constraints.

Several university course timetabling papers have appeared in the literature in the last few years which tackle various benchmark course timetabling problems. [17] employed a local search and ant based algorithms, tested on the eleven problems produced by Paechter's[1] course timetabling test instance generator (note that these instances are used to evaluate the method described in this paper). [3] employed a genetic and local search approach on the eleven benchmark course data sets. [14] employed a nonlinear great deluge on the same instances. In addition, [19] apply a nonlinear great deluge hyper-heuristic on the same eleven datasets and approved that the algorithm is able to obtain good results. On the other hand, a great deluge with kempe chain neighbourhood structure was employed by [2] to solve university course timetabling.

[22] applied a constraint-based solver approach to the curriculum-based course timetabling problems in the 2nd International Timetabling Competition (Track 1 and Track 3) as introduced by [10] and achieved first place in this competition. [15] applied a hybrid heuristic algorithm called adaptive tabu search to the same instances. [7] introduced a new solver based on a hybrid meta-heuristic to tackle scheduling problems. They applied it first on Udine data sets (based on Track 2 of ICT2007), achieving good solutions within a practical timeframe. [5] proposed a hybrid local search algorithm to solve curriculum-based course timetabling problems (ITC2007-Track 3). On the other hand, [23] proposed a branch-and-cut procedure. An integer programming is used in order to model the problem which to choose decision variables.

3 The Algorithm

The algorithm presented here is divided into two parts: construction and improvement algorithms. Within the latter stage, four neighbourhood structures have been employed.

[1] http://www.dcs.napier.ac.uk/~benp/

3.1 Neighbourhood Structures

The different neighbourhood structures and their explanation are outlined as follows:

N_1: Choose a single course/lecture at random and move to a feasible timeslot that can generate the lowest penalty cost.
N_2: Select two courses/lectures at random from the same room (the room is randomly selected) and swap timeslots.
N_3: Move the highest penalty course from a random 10% selection of the courses to a new feasible timeslot which can generate the lowest penalty cost.
N_4: Move the highest penalty course to a random feasible timeslot (both courses are in the same room).

3.2 Constructive Heuristic

The first part of our algorithm generates a feasible initial solution satisfying all the hard constraints. A saturation degree heuristic and largest degree heuristic are used to generate initial solutions for enrolment-based and curriculum-based course timetabling problems, respectively.

3.2.1 Enrolment-Based Course Timetabling Problem

Before applying the improvement algorithm a least saturation degree heuristic is used to generate initial solutions starting with an empty timetable [16]. The events with fewer rooms available and more likely difficult to be scheduled will be attempted to be scheduled first, without taking into consideration the violation of any soft constraints. This process is carried out in the first phase. If a feasible solution is found the algorithm terminates, otherwise phase 2 is executed. In the second phase, neighbourhood moves (N_1 and/or N_2) are applied with the goal of achieving feasibility. N_1 is applied for a certain number of iterations (set to 500, from experimentation). If a feasible solution is met, then the algorithm stops. Otherwise the algorithm continues by applying a N_2 neighbourhood structure for a certain number of iterations. Across all instances tested, solutions were made feasible before the improvement algorithm was applied.

3.2.2 Curriculum-Based Course Timetabling Problem

A feasible timetable is achieved by employing a largest degree heuristic, again starting with an empty timetable (Gaspero & Schaerf, 2003). The degree of an event is a count of the number of other events which conflict, in the sense that students are enrolled in both events. This heuristic orders events in terms of those with the highest degree first [14]. The events with highest degree of conflict will be attempted first without taking into consideration the violation of any soft constraints, until the hard constraints are met. All events are scheduled by randomly selecting the timeslot and the room that satisfies the hard constraints. Some events cannot be scheduled to a specific room; in this case they will be inserted in any randomly selected room. If all

the hard constraints are met the feasible solution is found and the algorithm terminates. Otherwise, phase 2 is executed. In phase 2, the process is carried out in a similar manner to the process in phase 2 in subsection 3.2.1. However, in this experiment the solutions were made feasible before the improvement algorithm is applied (such as in subsection 3.2.1).

3.3 Improvement Algorithm

An improvement algorithm is only applied on feasible solutions obtained from the constructive heuristic for both problems. During the improvement stage a set of the neighbourhood structures as outlined in subsection 3.1 are applied. The hard constraints are never violated during the timetabling process.

At the beginning of the search, four candidate solutions represented as *Sol** are generated by applying a set of neighbourhood structures (N_1, N_2, N_3, and N_4) within a Great Deluge algorithm andanother two candidate solutions are generated by applying two neighbourhood structures (i.e. N_1 and N_2) within a Tabu Search algorithm. The pseudo code for the algorithm implemented in this paper is given in fig. 1.

There are 3 steps involved in the improvement algorithm. In Step 1, the great deluge (see [11]) algorithm is employed followed by a Tabu Search (see [13]) in Step 2. Step 3 deal with accepting best solution obtained from Step1 and Step2.

```
Set the initial solution, Sol by employing a constructive heuristic;
   Calculate initial cost function f(Sol);
   Set best solution Sol_best ← Sol;
do while (not termination criteria)
     Step 1: Great Deluge
     Step 2: Tabu Search
     Step 3: Accepting Solution
end do
```

Fig. 1. The pseudo code for the improvement algorithm

Step 1: Great Deluge

At the start, the current solution, SolGD and best solution, *SolbestGD* is set to be *Sol* (obtained from the constructive algorithm). The quality measure of the solution *SolGD* and *SolbestGD* is given by *f*(*SolGD*) and f(*SolbestGD*), respectively. Let *K* be the total number of neighbourhood structures to be used in the search (*K* is set to be 4, applied in the preliminary experiments outlined in subsection 4.1.1 and it is set to be 2, applied in subsection 4.1.2). Note that, the number of neighbourhood structures employed in subsection 4.2 is due to the results obtained from our first experiment on enrolment based course timetabling problem in subsection 4.1.

In a *do-while* loop, a set of neighbourhoods *i* where $i \in \{1,\ldots,K\}$ is applied to *SolGD* to obtain $TempSolGD_i$. The best solution among $TempSolGD_i$ is identified, called, *SolGD**. The cost *f*(*SolGD**) is compared to that with *f*(*SolbestGD*). If it is better, then the current and best solutions are updated. Otherwise *f*(*SolGD**) will be compared against the boundary level. If the quality of *SolGD** is less than or equal to

the level, the current solution, *SolGD* will be updated as *SolGD**. Otherwise, the level will be increased with a random generated number (we set between 1 and 3) in order to allow some flexibility in accepting a worse solution. The process is repeated until the termination criterion is met.

Indeed, on the curriculum-based course timetabling problem (ITC2007 datasets), the time limit as in the competition is set as a termination criterion, thus we only employed two neighbourhood structures (i.e. N_1 and N_2) as generally the implementation of these neighbourhood structures is less time consuming.

Step 2: Tabu Search

A similar process as in Step 1 is applied, whereby a tabu search approach is employed on a different set of neighbourhood structures. In this experiment two neighbourhood structures are used to obtain $TempSolTS_i$ where $i \in \{1,2\}$. The best solution among $TempSolTS_i$ is identified, called *SolTS**. The *f*(*SolTS**) is compared to the *f*(*SolbestTS*). If it is better, then the current and best solutions are updated. Our tabu search algorithm uses only a short term memory. We add any moves that generate *SolTS** to the tabu list (if currently not in the tabu list) denoted as *TL*. These moves are not allowed to be part of any search process for a certain number of iterations (the tabu tenure). The tabu tenure is decreased after each of the iteration until it reaches zero. All tabu moves will change to non-tabu status when the tabu tenure is zero. In these experiments, we set the tabu tenure to be 10. The determination of these values was based on experimentation. The process is repeated until the termination criterion is met. In this step, the termination criterion is based on the number of iterations or when the optimal value (*OptimalValue*) is reached.

Note that the *OptimalValue* is set to 0/50/500 for small/medium/large datasets for the enrolment-based course timetabling problem, and for the curriculum-based course timetabling problem the *OptimalValue* is set to 0 for all datasets (i.e. lower than the best known results so far). In this experiment, we used two neighbourhood structures (i.e N_1 and N_2), for enrolment-based course timetabling problems one neighbourhood structure (i.e N_1) is employed. This is due to the fact that we are using the same time limit as in the ITC2007 competition. Step 3 involves accepting a solution to be used in the search process in the next iteration where the best solution from Step 1 (i.e. *SolbestGD*) and Step 2 (i.e. *SolbestTS*) is chosen (called *Sol**) and compared with the best solution so far (called Sol_{best}). If the quality of the *Sol** is better than the quality of the Sol_{best}, then the current solution (*Sol*) and best solution (Sol_{best}.) will be updated with *Sol** as shown in Fig. 2.

```
Choose the best between SolbestGD and SolbestTS, called Sol*
if (f(Sol*) < f(Sol_best))
        Sol ← Sol*;
    Sol_best ← Sol*
end if
```

Fig. 2. The pseudo code for Step 3 (in Fig. 1)

Note that the process is repeated and stops when the termination criterion is met. Note that the termination criteria for the enrolment-based and curriculum-based course timetabling problems are number of iterations and time limit, respectively.

4 Experimental Results

The algorithm is coded using Matlab under Windows XP and performed on the Intel Core 2 CPU 1.86 GHz computer, tested on eleven enrolment-based benchmark datasets and on Track 3 (UD2 datasets) from curriculum-based course timetabling problems.

4.1 First Experiment: Enrolment-Based Course Timetabling Problem

In this experiment, we have evaluated the search potential of our algorithm with a relaxed stop condition. For this purpose, we ran our algorithm for 200000 iterations (which took approximately twelve hours) with a different set of moves as presented in our preliminary experiment. The best results out of 11 runs obtained are presented. Table 1 shows the comparison of the approach in this paper with other available approaches in the literature on all instances, i.e. [M1] genetic algorithm and local search by [3], [M2] hybrid harmony search algorithm by [4], [M3] nonlinear great deluge hyper heuristic by [19], [M4] Ant Colony system with Simulated Annealing by [21], [M5] Ant Colony system with Tabu Search by [21], [M6] extended great deluge by [16], [M7] nonlinear great deluge by [14], [M8] harmony search by [20], [M9] dual simulated annealing by [1] and [M10] electromagnetic-like mechanism with great deluge by [18]. Note that the best results are presented in bold. The best results out of 11 runs obtained are presented. It can be seen our approach has better results on *medium1* and *medium2* datasets.

Table 1. Best results and comparison with other algorithms under relaxed stop condition

Dataset	Our method	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
small1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	3	**0**	**0**	**0**
small2	**0**	**0**	**0**	**0**	**0**	**0**	**0**	4	**0**	**0**	**0**
small3	**0**	**0**	**0**	**0**	**0**	**0**	**0**	6	**0**	**0**	**0**
small4	**0**	**0**	**0**	**0**	**0**	**0**	**0**	6	**0**	**0**	**0**
small5	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
medium1	**78**	175	99	88	117	150	80	140	168	93	96
medium2	92	197	**73**	88	121	179	105	130	160	98	96
medium3	135	216	130	**112**	158	183	139	189	176	149	**135**
medium4	**75**	149	105	84	124	140	88	112	144	103	79
medium5	68	190	**53**	103	134	152	88	141	71	98	87
Large	556	912	**385**	915	645	750	730	876	417	680	683

Fig. 3 (a), (b) and (c) show the box plots of the cost when solving *small*, *medium* and *large* instances, respectively. The results for the *large* dataset are less dispersed compared to *medium* and *small* (worse dispersed case in these experiments). We believe that the neighbourhood structures (N_1 and N_2) applied to the *large* dataset are able to force the search algorithm to diversify its exploration of the solution space by

moving from one neighbourhood structure to another even though there may be fewer and more sparsely distributed solution points in the solutions space since too many courses are conflicting with each other.

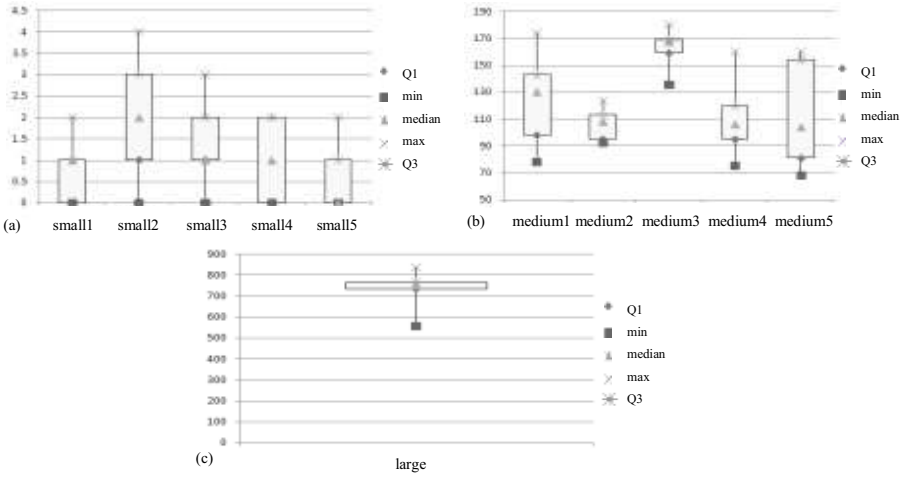


Fig. 3. (a), (b) and (c). Box plots of the penalty costs for small, medium and large datasets respectively

The comparisons between *small* and *medium* datasets in Fig. 3 (b) shows less dispersion of solution points compared to Fig. 3 (a). Again, applying the same neighbourhood structures (N_2 and N_3) for both instances most likely does not result in similar behaviour of the search algorithm. This is supported by Fig. 3 (a) where the dispersion of solution points for *small* datasets is not consistent from one to another. For example *small2* in Fig. 3 (a) shows worse dispersion compared to *small4*. From these experiments, we believe that the size of the search space may not be dependent on the problem size due to the fact that the dispersion of solution points are significantly different from one to another, even though the problems are from the same group of datasets with the same parameter values.

4.2 Second Experiment: Curriculum-Based Course Timetabling Problem

Here only two neighbourhood structures (N_1 and N_2) are applied on the twenty eight datasets (comp01-comp02 and DDS1-DDS7). In the beginning of the Great Deluge procedure a new strategy is used to control the application of the two neighbourhood structures on SolGD to obtain TempSolGD. When N_i is selected at random the other neighbourhood is marked as 'used', keeping N_i as "unused" in an attempt to keep using N_i in subsequent iterations while it continues to provide an improved solution. N_i is applied continually until a certain number of non-improvements (worse solutions) are observed, currently set to 5. N_i is then marked as "used" and other

neighbourhoods then marked as "unused", to allow a differing neighbourhood selection in the next iteration. The process is repeated until the termination criterion is met (in this case the time limit imposed by the ITC2007 competition rules).

Table 2 shows the comparison between the best results obtained by our algorithm with the best known results obtained from other approaches available in the literature for each instance, i.e. A hybrid approach by [22], a dynamic tabu search algorithm by [9], Adaptive Tabu Search by [15], repair-based heuristic by [8], and local search approach based on threshold acceptance by [12]. We have also compared our results with the best uploaded results in Curriculum-Based Course Timetabling, web site[2]. Note that the best results are presented in bold. The best results obtained by our approach are competitive to the previously best known results and has obtained a better result on the DDS4 dataset.

Table 2. Best results and comparison with other algorithms

Dataset	Our Method	Best Known	Best uploaded to CBCTT	[22]	[9]	[15]	[8]	[12]
comp01	**5**	**5**	**5**	**5**	**5**	**5**	9	**5**
comp02	39	43	**24**	43	75	34	103	108
comp03	73	72	**66**	72	93	70	101	115
comp04	36	**35**	**35**	**35**	45	38	55	67
comp05	309	298	**292**	298	326	298	370	408
comp06	43	41	**28**	41	62	47	112	94
comp07	17	14	**6**	14	38	19	97	56
comp08	40	39	**37**	39	50	43	72	75
comp09	104	102	**96**	103	119	99	132	153
comp10	12	9	**4**	9	27	16	74	66
comp11	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
comp12	334	331	**310**	331	358	320	393	430
comp13	67	66	**59**	66	77	65	97	101
comp14	54	53	**51**	53	59	52	87	88
comp15	88	87	**66**	84	87	69	119	128
comp16	52	47	**22**	34	47	38	84	81
comp17	88	86	**60**	83	86	80	152	124
comp18	84	71	**65**	83	71	67	110	116
comp19	71	74	**57**	62	74	59	111	107
comp20	34	54	**4**	27	54	35	144	88
comp21	98	117	**86**	103	117	105	169	174
DDS1	132	-	**83**	-	1024	-	-	-
DDS2	**0**	-	**0**	-	**0**	-	-	-
DDS3	**0**	-	**0**	-	**0**	-	-	-
DDS4	**24**	-	30	-	233	-	-	-
DDS5	**0**	-	**0**	-	**0**	-	-	-
DDS6	7	-	**0**	-	11	-	-	-
DDS7	**0**	-	**0**	-	**0**	-	-	-

Fig. 4 shows the box plot of the penalty cost on some of the instances in the UD2 problem considered in this experiment. The results from the figures show less dispersions of solution points.

[2] http://tabu.diegm.uniud.it/ctt/

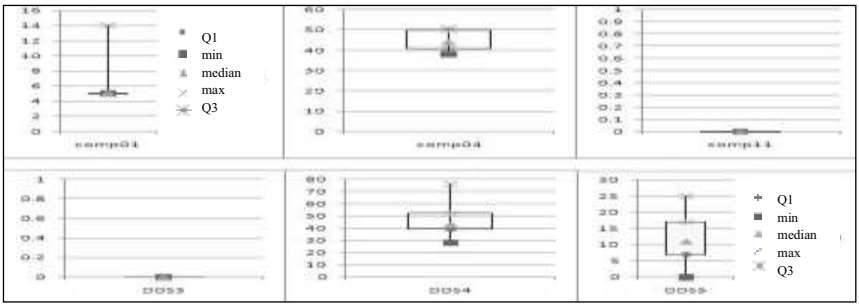


Fig. 4. Box plots of the penalty costs for comp01, comp04, comp11, DDS3, DDS4 and DDS5.

5 Conclusions and Future Work

This paper has focused on investigating the combination of great deluge and tabu search algorithms with a set of neighbourhood structures. A new strategy had been employed to control the application of a set of candidate neighbourhood structures. Preliminary comparisons indicate that this algorithm is competitive with other approaches in the literature, obtaining best results to those published to date on the original eleven enrolment-based course timetabling benchmark datasets, and several best results from the curriculum-based course timetabling benchmark data sets from the last ITC2007 competition. From analyzing and comparing the results obtained from UD2 datasets, it can be seen that a big number of soft constraints contributes to an increasing complexity of the problem. This can lead to increased difficulty in obtaining good solutions, although our approach is able to produce high quality solutions. Furthermore, from the results we can conclude that our approach is able to obtain some of best known results, whereby showing that this is a robust algorithm for a given different nature of the problems. The computational results and comparisons shown in this paper demonstrate the efficiency of our approach. In future, our approach would be applied to other similar problems, including Track 1 and Track 2 problems as described in the 2nd International Timetabling Competition (ITC2007).

References

1. Abdullah, S., Shaker, K., McCollum, B., McMullan, P.: Dual Sequence Simulated Annealing with Round-Robin Approach for University Course Timetabling. In: Cowling, P., Merz, P. (eds.) EvoCOP 2010. LNCS, vol. 6022, pp. 1–10. Springer, Heidelberg (2010a)
2. Abdullah, S., Shaker, K., McCollum, B., McMullan, P.: Incorporating Great Deluge with Kempe Chain Neighbourhood Structure for the Enrolment-Based Course Timetabling Problem. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS, vol. 6401, pp. 70–77. Springer, Heidelberg (2010b)
3. Abdullah, S., Turabieh, H.: Generating university course timetable using genetic algorithms and local search. In: The Third International Conference on Convergence and Hybrid Information Technology, ICCIT, vol. I, pp. 254–260 (2008)

4. Al-Betar, M., Khader, A.: A harmony search algorithm for university course timetabling. Annals of Operations Research 194(1), 3–31 (2012), doi:10.1007/s10479-010-0769-z
5. Bellio, R., Di Gaspero, L., Schaerf, A.: Design and statistical analysis of a hybrid local search algorithm for course timetabling. Journal of Scheduling, 1–13 (2011)
6. Burke, E., Kendall, G.: Search methodologies: introductory tutorials in optimization and decision support techniques. Springer (2005)
7. Burke, E., Marecek, J., Parkes, A., Rudová, H.: Decomposition, reformulation, and diving in university course timetabling. Computers & Operations Research 37(3), 582–597 (2010)
8. Clark, M., Henz, M., Love, B.: QuikFix A Repair-based Timetable Solver. In: Proceedings of the 7th PATAT Conference, Burke, Gendreau (2008)
9. De Cesco, F., Di Gaspero, L., Schaerf, A.: Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, and results. In: Proceedings of the 7th PATAT Conference (2008)
10. Di Gaspero, L., McCollum, B., Schaerf, A.: The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). In: The 14th RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion (2007)
11. Dueck, G.: New optimization heuristics. Journal of Computational Physics 104(1), 86–92 (1993)
12. Geiger, M.: An application of the Threshold Accepting metaheuristic for curriculum based course timetabling. In: Proceedings of the 7th PATAT Conference (2008)
13. Glover: Tabu Search. Kluwer Academic, Boston (1997)
14. Landa-Silva, D., Obit, J.: Great deluge with non-linear decay rate for solving course timetabling problems. In: Proceedings of the 2008 IEEE Conference on Intelligent Systems (IS 2008), pp. 8.11–8.18. IEEE Press (2008)
15. Lü, Z., Hao, J.: Adaptive tabu search for course timetabling. European Journal of Operational Research 200(1), 235–244 (2010)
16. McMullan, P.: An extended implementation of the great deluge algorithm for course timetabling. In: Computational Science–ICCS 2007, pp. 538–545 (2007)
17. Socha, K., Knowles, J., Sampels, M.: A max-min ant system for the university course timetabling problem. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) Ant Algorithms 2002. LNCS, vol. 2463, p. 1. Springer, Heidelberg (2002)
18. Turabieh, H., Abdullah, S., McCollum, B.: Electromagnetism-like Mechanism with Force Decay Rate Great Deluge for the Course Timetabling Problem. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS (LNAI), vol. 5589, pp. 497–504. Springer, Heidelberg (2009)
19. Obit, J., Landa-Silva, D., Ouelhadj, D., Sevaux, M.: Non-linear great deluge with learning mechanism for solving the course timetabling problem. In: Proc. 8th Metaheuristics Int. Conf., p. 10 (2009)
20. Al-Betar, M.A., Khader, A.T.A., Liao, I.Y.: A Harmony Search with Multi-pitch Adjusting Rate for the University Course Timetabling. In: Geem, Z.W. (ed.) Recent Advances In Harmony Search Algorithm. SCI, vol. 270, pp. 147–161. Springer, Heidelberg (2010)
21. Ayob, M., Jaradat, G.: Hybrid ant colony systems for course timetabling problems. In: Proc. 2nd Conf. Data Mining Optimization, October 27-28, pp. 120–126 (2009)
22. Müller, T.: ITC2007 solver description: a hybrid approach. Annals of Operations Research 172(1), 429–446 (2009)
23. Burke, E.K., Mareček, J., Parkes, A.J., Rudová, H.: A branch-and-cut procedure for the Udine course timetabling problem. Annals of Operations Research, 1–17 (2011)

Weight Learning for Document Tolerance Rough Set Model*

Wojciech Świeboda[1], Michał Meina[2], and Hung Son Nguyen[1]

[1] Institute of Mathematics, The University of Warsaw,
Banacha 2, 02-097, Warsaw, Poland
[2] Faculty of Mathematics and Computer Science,
Nicolaus Copernicus University, Toruń, Poland

Abstract. Creating a document model for efficient keyword search is a long studied problem in Information Retrieval. In this paper we explore the application of Tolerance Rough Set Model for Documents (TRSM) for this problem. We further provide an extension of TRSM with a weight learning procedure (TRSM-WL) and compare performance of these two algorithms in keyword search. We further provide a generalization of TRSM-WL that imposes additional constraints on the underlying model structure and compare it to a supervised variant of Explicit Semantic Analysis.

1 Introduction

Current Information Retrieval (IR) systems share a standard interaction scenario: a user formulates a query and then the system provides results based on query-to-document relevance. Retrieval efficiency is dependent upon the number of terms that overlap between the query and a document. The main issue in IR is known as *vocabulary problem*, which is a common mismatch in choosing the same terms by a user and by an indexer. One of the first evaluations of an IR system [1] concludes that formulating a proper keyword for search query demands from the user predicting existence of possible words in a document. Such prediction can be challenging and eventually can lead to scalability issues for IR systems. In addition, users may not agree with the choice of keywords while searching for the same objects. Furnas et. al [2] reports that in spontaneous word choice for objects in five domains, two people favored same term with less than 20 percent frequency. Therefore, two main techniques were developed to overcome this issue: (1) Query Expansion and (2) Document Expansion.

Query Expansion (QE) is on-line process of adding additional words to the query that best describe the user's intent in searching. Document Expansion

* The authors are supported by grant 2012/05/B/ST6/03215 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).

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 385–396, 2013.

(DE), on the other hand, is a process of document vocabulary augmentation at indexing time, predicting possible queries for each document. QE is a long studied problem in Information Retrieval (see exhaustive survey [3]), but the approach of extending documents at indexing time do not get such attention. Lack of transaction constraints (as in QE) should be considered as a promise in more effective model design. On the other hand DE involves enlarging the size of the index which can lead to problems with maintaining it. In this paper we will also investigate trade-off between retrieval effectiveness and index size problems.

Many approaches have been studied for solving the *vocabulary problem* both in QE and DE. Straightforward methods are based examination of word co-occurrence statistics in the corpus [4] outputting possible extension words. Then those words are enclosed into index or used in on-line query processing. Cluster-based Information Retrieval assumes that belonging of two documents to the same cluster carry some information about their correspondence therefore this information can be used in search process. Variety of cluster-based search methods and clustering algorithms was introduced and tested [5,6]. Different category of techniques exploits semantic information from external sources. Expanding term by theirs synonyms after word sense disambiguation using WordNet Ontology (along with other relation) [7] reports good performance boost. Although we see major differences between all of the techniques the common feature of term correspondence is exploited in the final extension. In this paper we present a document extension method which may encapsulate different tolerance relations between terms. The method is a variant of a Tolerance Rough Set Model for Documents [8,9] (TRSM). We supplement TRSM by a weight learning method in an unsupervised setting and apply the model to the problem of extending search results. We also introduce a method for a supervised multilabel classification problem and briefly compare it to an algorithm described in [10], which is based on Explicit Semantic Analysis [11].

The outline is as follows: We begin this paper by reviewing the Vector Space Model (a basic model widely used in Information Retrieval), and fundamentals of Rough Set Theory based both on indiscernibility relation and based on tolerance relation. Afterwards, we review TRSM model and introduce a weight learning scheme which we validate in the context of document retrieval. We further discuss an extension of the model and show its connection to a multilabel classifier based on Explicit Semantic Analysis [11].

2 Basic Notions

Rough Set Theory, developed by Pawlak[12] is a model of approximation of sets. An Information System $\mathbf{I}$ is a pair $\mathbf{I} = (U, A)$. U is called the *Universe* of objects and is the domain whose subsets we wish to represent or approximate using attributes, i.e. elements of A. Each attribute $a_i \in A$ is a function $a : U \to V_a$, where V_a is called the *value set* of attribute a.

For a subset of attributes $B \subseteq A$ we define a *B-indiscernibility* relation $IND(B) \subseteq U \times U$ as follows:

$$(x, y) \in IND(B) \iff \forall_{a \in A} a(x) = a(y) \tag{1}$$

$IND(B)$ is an equivalence relation and defines a partitioning of U into equivalence classes which we denote by $[x]_B$ ($x \in U$). B-Lower and B-upper approximations of a concept $X \subseteq U$ are defined as follows:

$$\mathcal{L}(X) = \{x \in U : [x]_B \subseteq X\} \tag{2}$$

$$\mathcal{U}(X) = \{x \in U : [x]_B \cap X \neq \emptyset\} \tag{3}$$

2.1 Tolerance Approximation Spaces

Indiscernibility relation in standard Rough Set Model is an equivalence relation. This requirement is known to be too strict in various applications. Skowron et al. [13] introduced Tolerance Approximation Spaces (and Generalized Approximation Spaces), relaxing conditions on the underlying relation. In this framework, indiscernibility of objects is defined by a tolerance relation.

An *Approximation Space* is defined as a tuple $\mathcal{R} = (U, I, \nu, P)$, where:

- U is a non-empty universe of objects,
- An *uncertainty function* $I : U \to \mathcal{P}(U)$ is any function such that the following conditions are satisfied:
  - $x \in I(x)$ for $x \in U$,
  - $y \in I(x)$ iff $x \in I(y)$.
- A *vague inclusion function* $\nu : \mathcal{P}(U) \times \mathcal{P}(U) \to [0, 1]$, such that $\nu(X, \cdot)$ is a monotone set function for each $X \in P(U)$.
- A *structurality function* $P : I(U) \to \{0, 1\}$, where $I(U) = \{I(x) : x \in U\}$.

A *vague membership function* $\mu(I, \nu) : U \times \mathcal{P}(U)$ is defined as $\mu(I, \nu)(x, X) = \nu(I(x), X)$ and lower and upper approximations of $X \subseteq U$ are defined as:

$$L_{\mathcal{A}}(X) = \{x \in U : P(I(x)) = 1 \wedge \mu(I, \nu)(x, X) = 1\} \tag{4}$$

$$U_{\mathcal{A}}(X) = \{x \in U : P(I(x)) = 1 \wedge \mu(I, \nu)(x, X) > 1\} \tag{5}$$

We will further refer to I as to the *tolerance relation* and to $I(u)$ as to the *tolerance class*.

2.2 Tolerance Rough Set Model for Documents

An approximation space that has been applied to represent documents is Tolerance Rough Set Model introduced in [8,9].

Let $D = \{d_1, \ldots, d_N\}$ denote the set of documents and $T = \{t_1, \ldots, t_M\}$ denote the set of index terms. Each document d_i may be thus represented as a bag-of-words or (in a vector space model) as a vector $\langle w_{i,1}, \ldots, w_{i,M} \rangle$.

TRSM model is a tolerance approximation space $\mathcal{R} = (T, I_\theta, \nu, P)$, defined as follows:

- Universe is the set of terms T,
- Uncertainty function $I_\theta(t_i) = \{t_j : f_D(t_i, t_j) \geq \theta\} \cup \{t_i\}$, where $f_D(t_i, t_j)$ is the number of documents in D containing both term t_i and term t_j,
- Vague inclusion function $\nu(X, Y) = \frac{|X \cap Y|}{|X|}$,
- Structurality function: $P(I_\theta(t_i)) = 1$ for $t_i \in T$.

It is worth stressing that the trivial choice of structurality function guarantees that $d_i \subseteq U_\mathcal{R}(d_i)$.

The tolerance class $I_\theta(t_j)$ of a term t_j in this model is the set of terms frequently co-occurring with t_j. For the sake of illustration let us call such terms similar to t_j. Lower and upper approximations of a set is defined for an arbitrary subset of T and thus for any document $d_i \in D$. While lower approximations are not used in applications of this model, the upper approximation of a document d_i is the set of all such terms that are similar to any term $t_j \in d$.

If we consider the upper approximation of a document $d_i \in D$ for varying values of parameter θ in TRSM, we notice that as θ gets larger, it imposes further restriction on tolerance classes and thus shrinks the extension.

2.3 Extended Document Representation

In applications of TRSM one typically uses upper approximations of documents as a means of enriching document representations. While Bag-of-Words document representation can be extended directly using the model defined above, authors of TRSM [8,9] have also introduced a scheme for assigning term-document weights w^*_{ij} in Vector Space Model for terms t_j from the upper approximation of a document d_i. The extended weighting scheme is derived from standard TF-IDF weights w_{ij} as follows:

$$w^*_{ij} = \begin{cases} (1 + \log f_{d_i}(t_j)) \log \frac{N}{f_D(t_j)} & \text{if } t_j \in d_i \\ 0 & \text{if } t_j \notin \mathbf{U}_\mathcal{R}(d_i) \\ \min_{t_k \in d_i} w_{ik} \frac{\log \frac{N}{f_D(t_j)}}{1 + \log \frac{N}{f_D(t_j)}} & \text{otherwise} \end{cases} \tag{6}$$

where $f_D(t)$ denotes the number of documents containing term t.

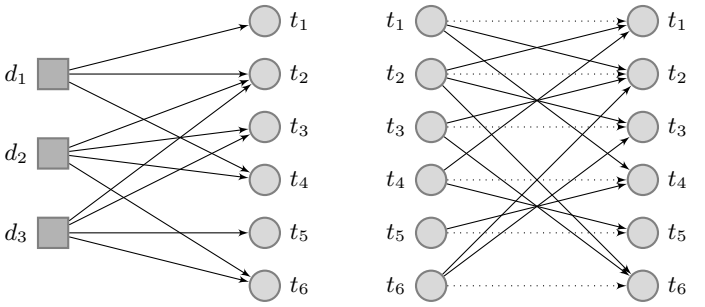
It is worth stressing that for a document d_j and a pair of terms $t_k \in d_j$, $t_j \notin d_j$, $w_{ij} < w_{kj}$, i.e. weights of extension terms of a document never exceed weights of terms that are in the original document.

The typical application of TRSM is document clustering [14,15,16,17,18], though see also [19] for discussion of other problems.

3 TRSM with Weight Learning

The purpose of our research is to explore an alternative framework for weight assignment in TRSM model. We define the underlying term-document structure in the (extended) Vector Space Model using TRSM – in other words, we assume that $w^*_{ij} = 0$ for $t_j \notin U_\mathcal{R}(d_i)$. We will speak of the model structure, the set of

permitted (i.e., nonzero) weights and the set of tolerance classes interchangeably. We further propose an alternative method of determining w_{ij} for $t_j \in U_{\mathcal{R}}(d_i)$.



(a) Bag-of-words document representation of a simple corpus with three documents and six terms.

(b) Structural part of TRSM model for parameter $\theta = 2$. Arrows describe tolerance classes of terms. .

Fig. 1. Bag-of-words document representation

The model that we propose aims to approximate original TF-IDF weights by a conical combination of TF-IDF weights of related terms. The set of terms related to t_i is the tolerance class of t_i in TRSM model (excluding t_i itself). In other words:

$$w^*_{ij} = \sum_{k=1}^{N} \delta(i,k,j)\beta_{kj} w_{ik} \tag{7}$$

where

$$\delta(i,k,j) = \begin{cases} 1 \text{ for } t_k \in d_i \wedge t_j \in I_\theta(t_k) \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

We will further demand that $\beta_{kj} > 0$ to stress the fact that tolerance classes in TRSM aim to capture similarities rather than dissimilarities between terms. We can thus rewrite equation 7 as follows:

$$w^*_{ij} = \sum_{k=1}^{N} \delta(i,k,j) e^{\alpha_{kj}} w_{ik} \tag{9}$$

In what follows, we propose a framework for determining weights α_{kj}. The underlying idea is to train a set of linear neurons (with trivial transfer functions) whose inputs are determined by TRSM. One can think of the problem as of "document recovery" and wish to approximate hidden w_{ij} by w^*_{ij}, i.e. try to assign weights w^*_{ij} so as to minimize error $E = \sum_{i=1}^{N} \sum_{j=1}^{M} L(w_{ij}, w^*_{ij})$ for a convenient loss function L. For simplicity, we pick the square loss function. Since

the choice of a weights α_{kj} has no bearing on $L(w_{ij}, w^*_{ij})$ for $t_j \notin U_{\mathcal{R}}(d_i)$, we can further restrict the summation to $i = 1, \ldots, N$ and j such that $t_j \in U_{\mathcal{R}}(d_i)$.

A natural additive update (proportional to negative gradient) is:

$$\Delta\alpha_{kj} \propto \sum_{i=1}^{N} (w_{ij} - w^*_{ij}) \delta(i,k,j) e^{\alpha_{kj}} w_{ik} \tag{10}$$

A commonly used mode of training is on-line learning, where the algorithm iterates the corpus document-by-document. In this approach, the corresponding updates during processing of document i are: $\Delta\alpha_{kj} \propto (w_{ij} - w^*_{ij})\delta(i,k,j)e^{\alpha_{kj}} w_{ik}$

Please note that as soon as the model structure is determined, perceptron weights are updated independently of each other. However, typically the model itself (whose size is determined by the set of nonzero weights) can be stored in computer memory, whereas the document corpus needs to be accessed from a hard drive. Processing the corpus sequentially document-by-document and updating all relevant weights is beneficial for technical purposes due to a smaller (and sequential) number of reads from a hard disk. Thus, training all perceptrons simultaneously is beneficial (strictly) for technical reasons.

Let us further call this model TRSM-WL (TRSM with weight learning). In principle it is an unsupervised learning method. The learned model can be applied to new documents and thus is inferential in nature.

Algorithm 1. Weight update procedure in TRSM-WL.

Input: $W = (w_{ik})_{i,k}$ (the document-term matrix), $Tol : T \to T$ (tolerance class mapped to each term).

Output: $\alpha = (\alpha_{kj})_{k,j}$.

for k, j such that $t_j \in I_\theta(t_k)$ **do**
  `/* The initializing distribution of` α_{kj} `(implicitly - of` $\beta_{kj} = e^{\alpha_{kj}}$`) is a modeling choice. */`
  $\alpha_{k,j} = RandNorm(\mu = 0, \sigma^2 = 1)$
for i in $1, \ldots, |D|$ **do**
  `/*` $\tilde{d}$`: the set of terms in the extension of document` d`. */`
  $\tilde{d} = d \cup \bigcup_{t_k \in d_i} I_\theta(t_k)$;
  `/* Determine weights` w^*_{ij} `for` $t_j \in \tilde{d}_i$`. */`
  for j in $1, \ldots, |\tilde{d}_i|$ **do**
    $w^*_{ij} = \sum_{k:t_k \in d_i} e^{\alpha_{kj}} w_{ik}$
  for j in $1, \ldots, |\tilde{d}_i|$ **do**
    for k in $1, \ldots, |d_i|$ **do**
      `/* Apply updates to weights` α`.` $\eta(i)$ `is a damping factor which determines the proportionality ratio of consequent updates. */`
      $\alpha_{kj} = \alpha_{kj} + \eta(i)(w_{ij} - w^*_{ij}) e^{\alpha_{kj}} w_{ik}$

Algorithm 1. shows pseudo-code for the weight update procedure. For simplicity, we assume that we iterate only once over each document (document d_i is processed in step i). In practice (and in experiments that follow) we iterated over the document corpus several times. The damping factor $\eta(i)$ used in experiments was picked inversely proportional i.

4 Experimental Results

We conducted experiments on the ApteMod version of `Reuters-21578` corpus. This corpus consists of 10,788 documents from Reuters financial service. Each document is annotated by one or more categories. The distribution of categories is skewed with 36.7% of the documents in the most common category and only 0.0185% (2 documents) in each of the five least common categories.

We applied stemming and stop word removal in order to prepare a bag-of word representation of documents. Test queries were chosen among all words in corpus using Mutual Information (MI) coefficient:

$$I(C,T) = \sum_{c\in\{0,1\}} \sum_{t\in\{0,1\}} p(C=c,T=t)\log_2\left(\frac{p\,(C=c,T=t)}{p\,(C=c)\,p\,(T=t)}\right), \qquad (11)$$

where $p(c=0)$ represents the probability that randomly selected document is a member of particular category and $p(c=1)$ represents probability that it isn't. Similarly, $p(t=1)$ represents the probability that a randomly selected document contains a given term, and p(T = 0) represents the probability that it doesn't. Next for each category 5 terms with highest MI was taken and used as query.

4.1 Model Convergence

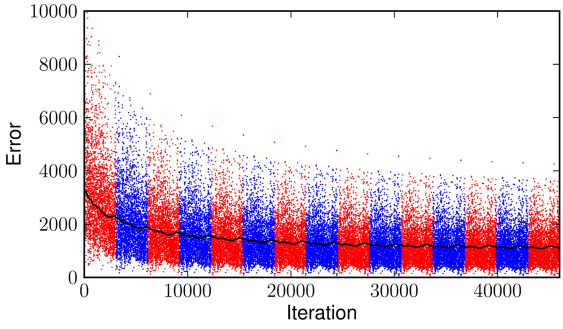


Fig. 2. Model convergence

Fig. 2 shows the squared loss for each document in the training set when the algorithm is executed. The y axis on the plot shows the contribution of each

document i to error E, i.e. $E_i = \sum_{j=1}^{M}(w_{ij} - w_{ij}^*)^2$. Each block corresponds to a single iteration over the entire document repository, and thus the error corresponding to each document appears several times, once in each alternating block. The graph shows that the first two iterations provide the biggest contribution to overall model fit.

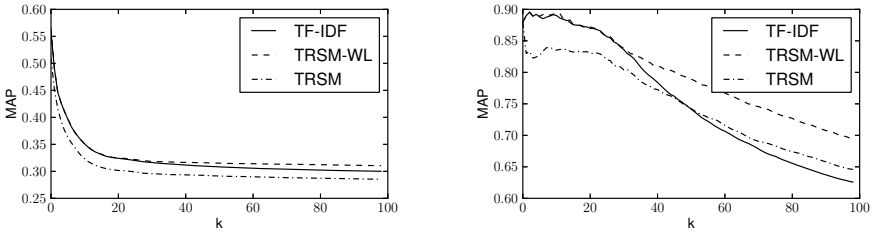
4.2 Searching Using Extended Document Representation

In order to measure effectiveness of search we prepared Information Retrieval systems based on TF-IDF, TRSM and our TRSM-WL weighting schemes (the methodology is described in section 2). We considered two test cases by dividing the query set into two subsets Q_1 and Q_2 taking into account size of expected result list, eg. number of documents in category. First one describes small categories ($n < 100$) and second large ones ($n \geq 100$).

Query-Set	TF-IDF	TRSM / TRSM-WL
All	0.457 ± 0.317	0.764 ± 0.335
Q_1	0.329 ± 0.252	0.733 ± 0.364
Q_2	0.325 ± 0.200	0.993 ± 0.006

Fig. 3. Recall of document search based on TF-IDF and TRSM Extended Document Representations.

As we see on Fig. 3 TRSM and TRSM-WL resulted in significant better recall comparing to standard TF-IDF. Both models with extended representation outputs the same recall score since TRSM-WL reassigns weight for the same co-located terms. It's worth stressing out that extended representation for low ϕ parameters tends to be very large in most cases. Moreover size of the extended representation correlates positively with corpora size with makes the model unusable for larger documents collection. Therefore our weighting schemata for TRSM discards some terms (by assigning low weights) that are insignificant for retrieval performance.



(a) Query set Q_1. (b) Query set Q_2.

Fig. 4. Mean Average Precision at k cut-off rank. TRSM and TRSM-WL with parameter $\phi = 10$

In order to examine top-k queries we present Mean Average Precision on Fig. 4. TF-IDF is treated as a baseline document r model in search task. It is noticeable that in two test cases classic TRSM is worse that baseline method. Our weighting schemata for TRSM makes our model batter or on the same level that the baseline. In second test run (Q_2) after significant drop after $k = 30$ represents the "vocabulary problem". Simply there are no terms in documents to match with the query and our model tries to match additional ones on the same level of precision baseline method.

5 Concluding Remarks and Future Work

While this section is not covered in experiments (experimental section focuses on document extension for Information Retrieval), we nevertheless stress that the outlined method is much more universal. In this section we provide a generalization of TRSM-WL and compare it with a variant of Explicit Semantic Analysis used for multilabel classification.

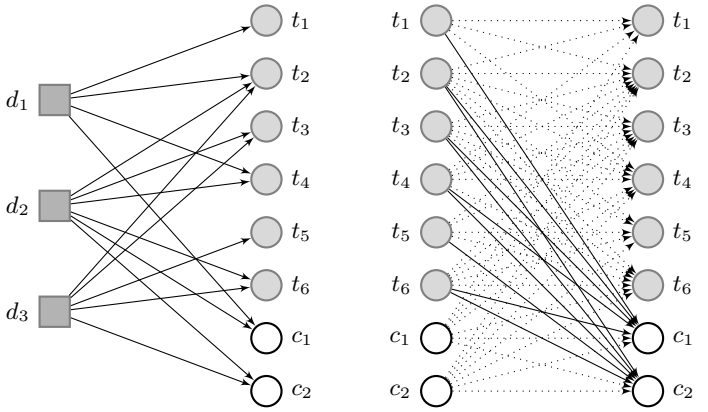
Explicit Semantic Analysis (ESA) introduced in [11] is a model of semantic relatedness which represents documents as a weighted vector of ontology-based concepts. The ontology applied in the original paper was Wikipedia. A weight update procedure which preserves the underlying model structure was introduced in [10]. For convenience, let us call this algorithm ESA-WL, by analogy to TRSM-WL introduced earlier.

In this model the structure is determined by the content of ontology: a term t_j is considered relevant to concept $c_i \in C$ iff t_j appears in the accompanying description of c_i. The input for the weight updating algorithm is a set of documents that share the same dictionary as the set of concept descriptions. Furthermore, these documents are labeled using concepts from C.

In experiments described in [10], the ontology used was MeSH dictionary [20], while inputs used for weight updating algorithm were documents from PubMed Central Open Access Subset[21] along with their document-concept assignments. For convenience (we introduce a slightly different notation) we may assume that documents are represented as vectors in Vector Space Model using an extended set of terms $T' = T \dot{\cup} C$. Weights corresponding to terms in T are TF-IDF weights, whereas weights corresponding to concepts are binary.

In this section we introduce a generalization of the model we have defined earlier in our paper, which encompasses ESA-WL with a slightly modified weight update procedure.

In this model only term-concept relations and weights are relevant, whereas TRSM (along with weight updating) introduced so far also models term-term and concept-concept relations and weights. Structurality function P in tolerance approximation spaces was designed to be the mechanism for filtering tolerance classes which are actual building blocks of lower and upper approximations. Therefore, for the purpose of this particular model, we propose a slightly different definition of a tolerance approximation space. Instead of structurality function P as in the original definition we propose to introduce:



(a) Bag-of-words document representation of our corpus. Documents are now additionally tagged with concepts c_1 and c_2.

(b) Structural part of ESA model (solid lines) coincides with term-concept edges in the approximation space of TRSM model with parameter $\theta = 1$.

Fig. 5. For a fixed parameter θ, bag-of-words document representation on Figure (a) determines the structural part of ESA model on Figure (b).

$$P^* : U \to \mathcal{P}(U) \text{ such that } \forall_{u \in U} P^*(u) \subseteq I_\theta(u) \tag{12}$$

Rather than a binary indicator whether a tolerance class can be used as a building block for approximations, now structurality function enables us to filter model structure in arbitrary manner. In this approach, the tolerance relation itself is used for modeling domain knowledge, whereas structurality function imposes constraints on the resulting model. Such formulation easily leads to additional extensions and potential applications.

Traditionally in TRSM we write T (for terms) rather than U to denote the Universe. Since our focus now is on ESA-WL, the Universe now consists of the extended set of terms $T' = T \dot{\cup} C$. The index θ in I_θ is often omitted when the approximation space is not explicitly parametrized, i.e. when one analyzes a single model rather than a model family. Thus, our definition of structurality function P^* can be rewritten as follows:

$$P^* : T' \to \mathcal{P}(T') \text{ is such that } \forall_{t' \in T} P^*(t') \subseteq I(t') \tag{13}$$

We now aim to define structurality function P^* which is appropriate to the model at hand. In order to define the same structure as ESA we simply pick $\theta = 1$ and define:

$$P^*(t') = \begin{cases} I_1(t') \cap C & \text{if } t' \in T \\ \emptyset & \text{if } t' \in C \end{cases} \tag{14}$$

The definition of $\delta(i, k, j)$ also requires a minor modification:

$$\delta(i,k,j) = \begin{cases} 1 \text{ for } t_k \in d_i \wedge t_j \in P^*(t_k) \\ 0 \text{ otherwise} \end{cases} \tag{15}$$

Our weight update procedure provides an alternative to the algorithm described in [10]. While TRSM-WL is in essence an unsupervised learning method, by dividing the set of terms $T' = T \dot{\cup} C$ into conditions T and decisions C, and by imposing constraints (by our choice of P^*), we have thus transformed it into a supervised learning method (a multilabel classifier). In this example, the structural part of TRSM-WL and ESA-WL[10] models is essentially the same, with slightly different weight update procedures.

Below we present future plans concerning the model discussed in this paper:

- In this paper we have assumed that the tolerance relation is defined simply by co-occurrences. Furthermore, we assumed that it is known beforehand (only weight updates were determined in an online manner). The model structure can also be incrementally refined (approximated) online while following user queries.
- Other definitions of tolerance relations (uncertainty functions) may be applied to define the underlying structure, e.g. using term similarity induced by WordNet ontology[22] or similar.
- One could add a regularizing factor to guarantee uniqueness of the global optimum. While uniqueness is desirable, for simplicity we have omitted regularization in the current formulation.
- Model analysis: extended and un-extended document representations.
  Let W be the document-matrix of the text corpus and W^* be the matrix of extended TF-IDF weights (i.e., inferred by the algorithm). We plan evaluate the effect of α in a model defined as

$$\widetilde{W} = \alpha W + (1 - \alpha) W^*$$

  for $\alpha \in [0, 1]$.
- Similarly, we plan to evaluate the effect of parameter θ on the size of the model $\sum_{j:t_j \in T} |I_\theta(t_j)|$, on the density of W^* and on information content (inferential value) of the model.

References

1. Blair, D.C., Maron, M.E.: An evaluation of retrieval effectiveness for a full-text document-retrieval system. Commun. ACM 28(3), 289–299 (1985)
2. Furnas, G.W., Landauer, T.K., Gomez, L.M., Dumais, S.T.: The vocabulary problem in human-system communication. Commun. ACM 30(11), 964–971 (1987)
3. Carpineto, C., Romano, G.: A survey of automatic query expansion in information retrieval. ACM Comput. Surv. 44(1), 1:1–1:50 (2012)
4. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
5. Voorhees, E.M.: The cluster hypothesis revisited. In: Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1985, pp. 188–196. ACM, New York (1985)
6. Leuski, A.: Evaluating document clustering for interactive information retrieval. In: Proceedings of the Tenth International Conference on Information and Knowledge Management, CIKM 2001, pp. 33–40. ACM, New York (2001)

7. Agirre, E., Arregi, X., Otegi, A.: Document expansion based on wordnet for robust IR. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING 2010, pp. 9–17. Association for Computational Linguistics, Stroudsburg (2010)
8. Kawasaki, S., Nguyen, N.B., Ho, T.B.: Hierarchical document clustering based on tolerance rough set model. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 458–463. Springer, Heidelberg (2000)
9. Ho, T.B., Nguyen, N.B.: Nonhierarchical document clustering based on a tolerance rough set model. International Journal of Intelligent Systems 17, 199–212 (2002)
10. Janusz, A., Świeboda, W., Krasuski, A., Nguyen, H.S.: Interactive document indexing method based on explicit semantic analysis. In: JRS 2012, pp. 156–165 (2012)
11. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 1606–1611 (2007)
12. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers (1991)
13. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27(2/3), 245–253 (1996)
14. Ho, T.B., Kawasaki, S., Nguyen, N.B.: Intelligent exploration of the web, pp. 181–196. Physica-Verlag GmbH, Heidelberg (2003)
15. Nguyen, H.S., Ho, T.B.: Rough document clustering and the internet. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, pp. 987–1003. John Wiley & Sons, Inc., New York (2008)
16. Nguyen, S.H., Świeboda, W., Jaśkiewicz, G.: Extended document representation for search result clustering. In: Bembenik, R., Skonieczny, Ł., Rybiński, H., Niezgódka, M. (eds.) Intelligent Tools for Building a Scientific Information Platform. Studies in Computational Intelligence, vol. 390, pp. 77–95. Springer, Heidelberg (2012)
17. Nguyen, S.H., Świeboda, W., Jaśkiewicz, G., Nguyen, H.S.: Enhancing search results clustering with semantic indexing. In: SoICT 2012, pp. 71–80 (2012)
18. Nguyen, S.H., Świeboda, W., Jaśkiewicz, G.: Semantic evaluation of search result clustering methods. In: SYNAT 2012, pp. 393–414 (2012)
19. Virginia, G., Nguyen, H.S.: Lexicon-based document representation. Fundam. Inform. 124(1-2), 27–46 (2013)
20. United States National Library of Medicine: Introduction to MeSH - 2011 (2011)
21. Roberts, R.J.: PubMed Central: The GenBank of the published literature. Proceedings of the National Academy of Sciences of the United States of America 98(2), 381–382 (2001)
22. Miller, G.A.: Wordnet: A lexical database for english. Communications of the ACM 38(1), 39–41 (1995)

A Divide-and-Conquer Method Based Ensemble Regression Model for Water Quality Prediction

Xuan Zou[1,2,*], Guoyin Wang[1,2], Guanglei Gou[2,3], and Hong Li[1,2]

[1] Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing, China
zouxuan@cigit.ac.cn
[2] Institute of Electronic Information & Technology, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China
[3] School of Information Science and Technology, Southwest Jiao tong University, Chengdu, China

Abstract. This paper proposes a novel ensemble regression model to predict time series data of water quality. The proposed model consists of multiple regressors and a classifier. The model transforms the original time series data into subsequences by sliding window and divides it into several parts according to the fitness of regressor so that each regressor has advantages in a specific part. The classifier decides which part the new data should belong to so that the model could divide the whole prediction problem into small parts and conquer it after computing on only one part. The ensemble regression model, with a combination of Support Vector Machine, RBF Neural Network and Grey Model, is tested using 450-week observations of COD_{Mn} data provided by Ministry of Environmental Protection of the People's Republic of China during 2004 and 2012. The results show that the model could approximately convert the problem of prediction into a problem of classification and provide better accuracy over each single model it has combined.

Keywords: Water quality prediction, ensemble regression, divide-and-conquer method, time series data mining.

1 Introduction

Water quality prediction plays an important role in water resource management. Accurate predictions could provide supports to early warning of water pollution and save time for decision-making. So far, two kinds of approaches have been proposed for water quality prediction. One kind is the models based on the mechanism of movement, physical, chemical and other factors in the water and has been widely employed in different basins, e.g. QUASAR [1], WASP [2]. But the mechanistic models usually need complete observed data and mechanism knowledge, of which are difficult to get. Another kind is the models based on statistics and artificial intelligence. The rapid development of artificial intelligence provides us with more approaches for regression and better

* Corresponding author.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 397–404, 2013.

accuracy under varies situations. The Grey Model (GM) [3-5], Artificial Neural Network (ANN) [6-8] and Support Vector Machine (SVM) [9-10] have been widely used for prediction and forecasting in water resources and environmental engineering.

Apart from the ones with single model, approaches with multiple models are explored. Due to the chaotic nature of environment, single model may counter difficulties when making precise predictions, while an ensemble model could combine and enhance multiple predictors and make predictions based on different approaches. Recently multi-model approach is a popular research topic in solving water quality prediction problem [11-13].

Usually, as the work above, ensemble models need to train and test for every component, of which is time consuming. And the key factors are the differences between sub-models and assignment the weights of them, for they are trained to solve the same problem. In the same time, the divide-and-conquer method could divide the whole problem into minor problems and save time by training different sub-models for different tasks. The key factor of divide-and-conquer method is the division algorithm for the original problem.

As for the predictions in complicated environment, the number of tasks being divided may vary and is difficult to determine. So a self-adapted division algorithm is needed. The parallel neural network architecture based on NARA model and sieving method (PNN) could make division and automatically determine the number of tasks according to the feature of data [14]. Based on the fitting of each neural network, PNN divides the classification problem into small spaces and behaves well on the "Two-Spiral" problem, confirming that the division algorithm based on fitness of components could be an effective way.

In this paper, we propose an ensemble regression model based on the division algorithm of PNN. In section 2.1 we will introduce the division algorithm of PNN, and in section 2.2 we will explain the details of our model. In section 2, the prediction using water quality data will be employed to test our model and the results will be analyzed. And in the last section we will draw the conclusion.

2 Materials and Methods

2.1 The Parallel Neural Network Architecture Based on NARA Model and Sieving Method

PNN is composed of the control network CN, recognize network $RN_i (i = 1, 2, \ldots, p)$ and logic switch $\mathrm{LS_i}$.

The function of control network is the rough division of problem space. The control network outputs the subspace Q_i that the input vector X should belongs to and close the respect logic switch LS_i so that only the output of recognize network RN_i would be chosen and the outputs of other recognize networks $RN_j (j \neq i)$ would be ignored. The recognize network RN_i could correctly deal the problem in subspace Q_i while for the problems in other subspace, the correctness of RN_i could not be

guaranteed. The function of logic switch LS_i is to make sure the result of RN_i would be the effective output of the system ($LS_i = 1$, the switch is closed) or be ignored ($LS_i = 0$, the switch is open).

2.2 The Divide-and-Conquer Method Based Ensemble Regression Model (DM-ERM)

Based on the divide-and-conquer method, DM-ERM improves the division algorithm of PNN so that it would have the capacity of dealing problem of regression instead of classification, and introduces the variable threshold to enhance the convergence even in the worst situations. The process of DM-ERM consists of three parts: data preprocessing, model training and output. After the details are expressed, we will analyze the feature of DM-ERM.

Data Preprocess. We use the sliding window algorithm to extract subsequences from the original data. After removing the unavailable values, we place a fixed-size window in the start of the original time series data T. The data inside the window would be picked up as a whole to form a subsequence C_1. And then the window slides one data forward, with the data inside it forming a new subsequence C_2. The process is made repeatedly till the window reaches the end of the original data. In the case the length of original data is m and the length of sliding window is w, the quantity of subsequences is $m-w+1$. For the x steps prediction, each subsequence is split into two vectors (X_i, Y_i), with the first $w-x$ data of C_i compose the input vector X_i and the last x data compose the output vector Y_i. The set $S = \{(X_i, Y_i) \mid i = 1, 2, \ldots, m\text{-}w+1\}$ would be the training set of DM-ERM.

Model Training. Based on the division algorithm of PNN, DM-ERM has the similar structure of it. DM-ERM is made up of the regression layer and the control layer. The regression layer consists of one or more regressors RL_i while the control layer consists of one classifier CL.

Each time a regressor is trained and added into regression layer. The first regressor RL_1 would be trained by the initial training set S. For the next regressor RL_i, the training set is determined by the fitting result of RL_{i-1} and the value of variable threshold α. In the training process, DM-ERM needs an extra set C_{RL}.

Initially, the training set S_1 is equal to S, α is the initial value assigned by user (range from 0 to 1), and the set C_{RL} is empty.

For the training set S_i, the fitting result of RL_i is stored in set P_i:

$$P_i = \left\{(X_j, \hat{Y}_j) \mid X_j \in S_i\right\} \tag{1}$$

Where $\hat{Y}_j$ is the output of RL_i corresponding to the input X_j.

And then we measure the fitting effect of each output vector $\hat{Y}_j$ by mean relative error (MRE):

$$MRE_j = \frac{1}{n}\sum_{k=1}^{n}\frac{|\hat{Y}_{jk} - Y_{jk}|}{Y_{jk}} \tag{2}$$

Where $Y_j \in S_i$ and $n = \left|\hat{Y}_j\right| = |Y_j|$. After the MRE is calculated, we get the recognizing set RCG_i:

$$RCG_i = \left\{\left(X_j, Y_j\right) | MRE_j < \alpha\right\} \tag{3}$$

If $RCG_j \neq \varnothing$, then we could update the set C_{RL} and training set S_{i+1}:

$$S_{i+1} = S_i - RCG_i \tag{4}$$

$$C_{RL} = C_{RL} \bigcup \left\{\left(X_i, j\right) | X_i \in RCG_i\right\} \tag{5}$$

Else, the value of a is decreased and $\mathrm{RCG_i}$ is re-calculated till $RCG_j \neq \varnothing$.

After C_{RL} and S_{i+1} have been updated, the new regressor is added as above till $S_{i+1} = \varnothing$.

At last, C_{RL} is used to train the classifier CL.

Output. If the regression layer has n regressors, the final output f would be:

$$f = \sum_{i=1}^{n} w_i f_i \tag{6}$$

Where w_i is the weight of RL_i and f_i is its prediction value.

The weight could be calculated as follows:

$$w_i = \begin{cases} 0, (i \neq j) \\ 1, (i = j) \end{cases} \tag{7}$$

Where j is the classification result of CL on the testing data. Thus, after the classification of CL, only the regression results of RL_j needs to be computed.

Model Analyze. After the variable threshold α is introduced, the convergence is improved. In the division algorithm in PNN, when a network could not recognize any data, it would be trained repeatedly until some data are recognized. When in the worst situation that the data are too hard to recognize, α in DM-ERM would gradually decrease to zero and the model would converge in finite steps, while the training time of network in PNN is uncertain.

The initial value of α would affect the performance of DM-ERM. If the initial value of α is too big, DM-ERM would contain too many regressors whose training data would be in small size, making the final result unstable. If α is too small, the first regressor would overwhelm other regressors, resulting that the performance of DM-ERM would be mainly based on the first regressor and have little improvement compared with it.

In the situation that DM-ERM combines different submodels, the order of training would also affect DM-ERM's performance. The desampling process in DM-ERM would bring the whole training set to the first regressor, with the regressors after it would be trained by less data so that different submodels would deal with problems in

different data size. Usually, the first regressor would influence the result most, for it is the only one who has learned the whole problem.

The process of desampling also grants the ability of converting the regression problem into classification problem to DM-ERM. After the training of DM-ERM, we could know the division of problems, but we are not certain about the numerical relationship between different problems. So, as for DM-ERM, a classifier would behave better than linear weights assigned on the output of submodels.

3 Experiments and Discussions

3.1 Model Establishing of DM-ERM

Data Preprocess

The Three Gorge Reservoir is the biggest inundated area in the world, with the water level changing from 145m to 175m. The increasing of water level brings a bigger body of water, a lower flow velocity, a weaker self-purification capacity and a more complicated hydrological variation. Thus, the accurate prediction of water quality is of vital important. The Ministry of Environmental Protection of the People's Republic of China has provided the water quality time series data in Panzhihua[1] which is situated at the upper reaches of the Three Gorge Reservoir. Ranging from the first week in 2004 to the39th week in 2012, the data set contains 450-week observations on the four main water quality index (pH, DO, COD_{Mn}, NH_3-N). In the dataset, the water quality of Panzhihua has reached level 3 or higher for 26 times, with 22 times are caused by COD_{Mn}. So we choose COD_{Mn} as the aim of prediction.

In the 450 observations, two are unavailable due to the cut-out of the basin and removed. And then we employ the sliding window with a length of six to convert the original streaming time series data into 443 short time series subsequences. Each subsequence consists of six data, of which the first five history data regarded as the input, while the last regarded as the output of one-step prediction. Considering that leaving out one subsequence does not remove all the associated information due to the correlations with the subsequences after it, we choose to divide the subsequences according to their order in time series instead of cross-validation. The first 300 subsequences are chosen as the training set, while the rest as the testing set.

Model Training

DM-ERM could be composed of any regression models. We choose Genetic Algorithm optimized SVM (GA-SVM) [15], RBF Neural Network (RBF-NN) [16] and GM(1,1) as optional components. As is mentioned in section 2.2, when uniting heterogeneous models, the order of training in each model may affect the result of DM-ERM. Considering that GA-SVM has the best result in the three models, RBF-NN needs a relative huge dataset for training and GM(1,1) behaves well in small dataset, we choose to train GA-SVM first, RBF-NN second and GM(1,1) last. We choose

[1] Available at `http://datacenter.mep.gov.cn/`

k-Nearest Neighbor algorithm (kNN) [17] as the classifier. Along with 0.9 as the initial value of a, a DM-ERM with heterogeneous models is built and tested.

Evaluation Criteria of Model Performance

We choose mean relative error (MRE), quadratic loss function (QLF), Pearson product-moment correlation coefficient (R) and mean square error (MSE) as the evaluation measures.

3.2 Results and Analyzes

After training, DM-ERM is made up of four submodels, that is, a GA-SVM, two RBF-NNs and a GM(1,1). The details of submodels in the process of training are shown in **Table 1**. Meanwhile, we choose the best predictions of the four submodels for each observation in testing set to be displayed in **Fig.1**, for they could represent the limit of DM-ERM's accuracy.

For further contrast, the bagging regression[18] of GA-SVM and bagging regression of RBF-NN are made under the same training set and testing set. Each bagging contains 20 homogeneous individuals and the results of individuals' prediction are averaged as the result of bagging. The results of all models mentioned above are compared in **Table 2**. The best value of each criteria is made bold in the table.

Table 1. The details of submodels in DM-ERM

Submodel	Training order	Number of training data	Number of recognized data
GA-SVM	1	300	198
RBF-NN(1)	2	102	45
RBF-NN(2)	3	57	34
GM(1,1)	4	23	23

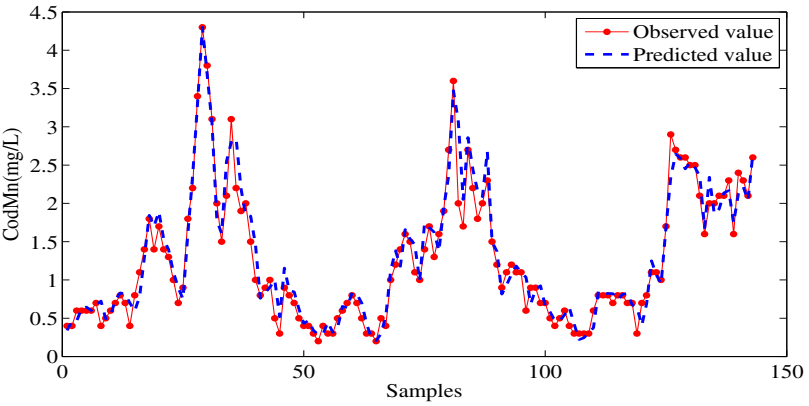


Fig. 1. The accuracy limit of DM-ERM

Table 2. Performance of these algorithms

Criteria	MRE	QLF	R	MSE
DM-ERM	**0.2447**	**0.1019**	**0.8955**	**0.1472**
GA-SVM	0.2659	0.1241	0.8794	0.1622
RBF-NN	0.2938	0.1797	0.88	0.1632
GM(1,1)	0.3156	0.1587	0.8525	0.2761
Bagging GA-SVM	0.2870	0.1494	0.8477	0.2071
Bagging RBF-NN	0.3216	0.1663	0.8856	0.1566

As shown in **Fig.1**, the combination of best predictions of submodels in DM-ERM have little loss compared with the observed data (with its MSE is 0.0441 and R is 0.9712), indicating that DM-ERM has the ability of converting the problem of regression into the problem of classification approximately. According to **Table 2**, the prediction of DM-ERM is superior to the one of every other model in all evaluation criteria.

4 Conclusion

The divide-and-conquer method could solve the problem of ensemble regression in time consuming and weights assigning. The key of divide-and-conquer method is the division algorithm. We improved the division algorithm in PNN to enhance its convergence and ability of solving regression problems, and proposed the divide-and-conquer method based ensemble regression model (DM-ERM) based on it. The experiment shows that: (1) DM-ERM could combine same or different models and transfer the problem of regression into the problem of classification. (2) The prediction made by DM-ERM is superior to the one made by its component.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (No. 61272060 and No. 61073146) and Chongqing Scientific and Technological Program (No. CSTC2011GGC40008 and No. CSTC2013JJB40003).

References

1. Wang, P.F., Martin, J., Morrison, G.: Water Quality and Eutrophication in Tampa Bay, Florida. Estuarine, Coastal and Shelf Science 49, 1–20 (1999)
2. Peng, S., Fu, G.Y., Zhao, X.: Integration of USEPA WASP model in a GIS platform. I. J. Zhejiang Univ.-Sci. A 11(12), 1015–1024 (2010)
3. Deng, J.: Introduction to grey system theory. The Journal of Grey System 1(1), 1–24 (1989)
4. Zhang, W., Liu, F., Sun, M.: The Application of grey model in dawu water quality predication water resource site. Journal of Shandong Agricultural University 33(1), 66–71 (2002) (in Chinese)

5. Ran, Y., He, W., Lei, X., Xia, H.: Application of GM(1,1) model and improved model to predict the water Quality of Weihe River in Tianshui section. Journal of Water Resources and Water Engineering 22(5), 88–91 (2011) (in Chinese)
6. Palani, S., Liong, S.-Y., Tkalich, P.: An ANN application for water quality forecasting. Marine Pollution Bulletin 56, 1586–1597 (2008)
7. May, D.B., Sivakumar, M.: Prediction of urban stormwater quality using artificial neural networks. Environmental Modelling & Software 24, 296–302 (2009)
8. Hong, G., Qi, L., Jun, F.: An efficient self-organizing RBF neural network for water quality prediction. Neural Networks 24, 717–725 (2011)
9. Dai, H.: Forecasting and evaluating water quality of Changjiang River based on composite least square SVM with intelligent genetic algorithms. Application Research of Computers 26(1), 79–81 (2009)
10. Xiang, Y., Jiang, L.: Water Quality Prediction Using LS-SVM and Particle Swarm Optimization. In: Second International Workshop on Knowledge Discovery and Data Mining, WKDD 2009, January 23-25, pp. 900–904 (2009)
11. Partalas, I., Tsoumakas, G., Hatzikos, E.V., Vlahavas, I.: Greedy regression ensemble selection: Theory and an application to water quality prediction. Information Sciences 178, 3867–3879 (2008)
12. Faruk, D.O.: A hybrid neural network and ARIMA model for water quality time series prediction. Engineering Applications of Artificial Intelligence 23, 586–594 (2010)
13. Sun, Z., Wang, B., Ji, H., Huang, Z., Li, H.: Water quality prediction based on probability-combination. China Environmental Science 31(10), 1657–1662 (2011) (in Chinese)
14. Wang, G., Shi, H.: Parallel Neural Network Architectures and Their Applications. In: Proceedings of International Conference on Neural Networks, Perth, Australia, III, pp. 1234–1239 (1995)
15. Huang, C.L., Wang, C.J.: A GA-based feature selection and parameters optimization for support vector machines. Expert Systems with Applications 31(2), 231–240 (2006)
16. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Computation 3(2), 246–257 (1991)
17. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
18. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)

A Self-learning Audio Player That Uses a Rough Set and Neural Net Hybrid Approach

Hongming Zuo and Julia Johnson

Department of Mathematics and Computer Science,
Laurentian University, Sudbury, Ontario, P3E 2C6 Canada

Abstract. A self-learning Audio Player was built to learn a users habits by analyzing operations the user does when listening to music. The self-learning component is intended to provide a better music experience for the user by generating a special playlist based on the prediction of a users favorite songs. The rough set core characteristics are used throughout the learning process to capture the dynamics of changing user interactions with the audio player. The engine is evaluated by simulation data. The simulation process ensures the data contain specific predetermined patterns. Evaluation results show the predictive power and stability of the hybrid engine for learning a users habits and the increased intelligence achieved by combining rough sets and NN when compared with using NN by itself.

Keywords: Artificial Neural Network, Rough Set, self-learning, hybridization.

1 Introduction

An audio player is popular software in every day life. Most people are using it to enjoy music when they are jogging, reading, resting and so on. There are a lot of different types of audio players, for example, Windows Media Player and iTunes. Most of them have just a basic function, like playing songs or shuffling song lists. The objective of this reasearch is to develop a RS and NN hybrid system that learns users preferences in a music player.

1.1 Background

When we will use the short form APP, we are referring specifically to a digital iPhone application though apps are used for all sorts of platforms other than digital iPhone applications, and we will use iOS to mean iPhone operating system.The Apple App store is a digital application distribution platform for iOS, developed and maintained by Apple Inc. People can develop their own applications, and publish them in the APP store. As of February 10, 2012, there are more than 700,000 third-party apps officially available on the APP store. As in May 15, 2013, downloads from the APP store reached 50 billion. Compared

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 405–412, 2013.

with the other digital platforms, the iOS APP store is the most popular digital application store.

iTunes is a product developed by Apple Company to play music. iTunes has a inventive function called Genius. This function can recommend playlists (a list of songs), provide a mixture of songs that go great together chosen from different libraries, and suggest songs that the user may like. The mechanism by which iTunes Genius works is proprietary to Apple. Researchers are studying Genius,trying out Genius on a test collection of music, and analyzing how it may be working. Their conclusion[1] is, first of all that Genius performs well at detecting acoustically similar songs, and secondly that its recommendation is derived from a purely content-based system. By content-based system, they mean that the songs are compared by descriptors developed by musicologists to judge whether two songs sounds similar. Those researchers contrast the content-based approaches with the meta- data approaches by which they mean the information about music such as artist, album and genre.

iTunes records the detailed information about songs the user has played including how many times the song has been played. It seems as though iTunes Genius[1] [2] recommends the playlist by analyzing such detailed information that reflect the habits of users, like which genre of music the user usually listens to, which artist is the users favorite and which album the user tends to play. It is a very interesting area in AI to build software that can learn user habits and try to make a better solution for the user.

1.2 Objectives and Novelty

The main objective of this project is to analyze music with respect to the users operation on it such as how many times the song has been skipped and how many times the song has been picked. The first novelty of our approach is to develop a rough set and neural network hybrid system that learns users preferences in the music player. While others have used previously implemented RS and NN engines as third-part software, we implement the RS and NN engine by developing our own software. That is, we do the hybridization at the code level. The expected advantage is avoidance of the manual step between the Rough Set engine and the NN, that was required on the previous approaches[3][4]. We use rough sets core characteristics to guide NNs learning process in code.

In previous work, RS core characteristics were used to initialize the weight of inputs to the NN [4]. We tried this approach and found that it does not offer much advantage in improving the learning speed of the NN. However, in our research, we are using the important attributes to guide weight training throughout the training process not just at initialization to make sure the insignificant attributes will not have any influence on the learning process. The novelty lies in development of dynamic weight training because an attribute may become significant as people use this system while earlier it was not significant. Conversely, a significant attribute may no longer be significant.

2 Construction of Multilayer Neural Network

As its name suggests, a multilayer neural network consists of multiple layers of neurons. Normally, there is an input layer, hidden layers and an output layer. The neurons in the hidden layer process inputs and give the output to every neuron in the output layer.

There are many different learning algorithms for a multilayer neural network. The most popular one is back-propagation [5][6], in which one of the important steps is weight training.This step is to train the weight of each neuron and do the weight correction. When we are calculating the weight correction, there is a concept named learning rate. Learning rate a variable that defines the degree of weight correction.[7]. The learning rates function is to adjust the weight to a certain precision. If the learning rate is too large, the training of weights will not be successful. More successful weight training was obtained with a smaller learning rate.

The configuration of our NN is that there are 6 neurons in the hidden layer, 1 neuron in the output layer, and the learning rate is 0.1. The input neurons are the attributes from user's operations, for example, how many times the song has been skipped and how many times the song has been picked. The result of output neuron is the levels of song. The song level is a measurement for deciding how much the user likes this song. For example, level 5 means this song is the users favorite.

According to Zhang [11], one hidden layer neural network is sufficient to deal with any complex system. The more neurons in the hidden layer, the more precise the NN will be. But, at the same time, more neurons will cost more computer resources. Referring to [3], they found out that 5 neurons in the hidden layer gave better results in their problem. Because our NN needs to process more input attributes than in [3], we increased the number of neurons in the hidden layer. We make the NN good enough to solve the problem by increasing the hidden layer neurons, but also does not cost too much computer resource.

3 Rough Sets

Rough Set Theory (RST) [8] provides an idea to deal with imprecision in data. In order to understand RST, we need to know about the following concepts [9].

Information system framework: Adopting common notation in the field, we assume U to consist of objects in the universe, and A to consist of features of those objects. Then, $I = (U, A)$ is an information system. For every $\alpha \in A$, there are $a : U \Rightarrow V_a$. The V_a is the set of values that attribute a may assume.

Equivalence relation: For every $P \subset A$, there is an equivalence relation $IND(P)$. It is called the P-indiscernibility relation, defined as follows:

$$IND(P) = \{(x, y) \in U^2 | \forall a \in P, a(x) = a(y)\} \tag{1}$$

Reduct and Core: For any attribute $a \in A$, a is dispensable in the set A if $IND(A) = IND(A - a)$. If we eliminate all dispensable attributes from A, we get P as the reduct set of A, denoted by $RED(A)$ (using notation in [4]).

$$RED(A) = \{R : R \subset A, IND(R) = IND(A)\} \tag{2}$$

The intersection of all the reducts of A is called the core. [4]

$$CORE(A) = \cap RED(A) \tag{3}$$

All attributes in the core are indispensable for approximating X. X denotes an object set to be approximated where $X \subseteq U$. They cannot be removed from A without changing the original classification.

4 Hybridization

The way to hybridize NN and RS is that we use the ability of Rough Set to core the conditional attributes and find out the significant ones. Then, we use the core to help NN training. The whole process is illustrated as Figure. 1.

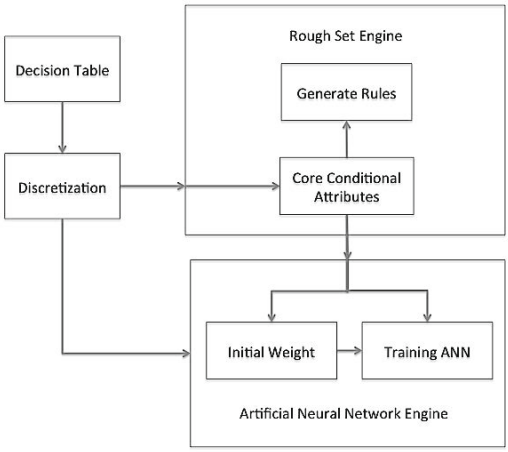


Fig. 1. The process by which the hybrid engine runs

Step 1: We discretize the decision table by using equal frequency binning discretization method.[10]

Step 2: We put the data into RS engine. Firstly, we core the conditional attributes. Secondly, we input the coring to NN engine to help initialize the weights of NN engine. By using the Rough Set Engine, we core the decision table and give a set from RS engine to NN engine. The set will be like $ca[6] = \{0, 1, 1, 1, 0, 1\}$.The set of zeros and ones mean there a 6 conditional attributes. Besides the first and the fifth elements in the list both of which is 0, the attributes are all significant attributes indicated by 1 in the set. We initialized the weight as 0 when the attribute is insignificant.

Step 3: NN engine uses the data from discretization and the input from RS engine to train itself. The result from RS engine will help NN to initialize weights and also guide the training process. In the training process, we will assign the learning rate of insignificant attributes to be 0. It means that we will not do any weight corrections for insignificant attributes. So, from the beginning of the training process to the end, the insignificant attributes will not influence the whole system.

5 Evaluation Result and Discussions

By correctness of NN, we mean the NN will make a good prediction after training. A good prediction is defined as small error between desired prediction and actual prediction. Two methods will be described for assessing quality of the NN hybrid Engine. The first method was intended to show the precision of prediction by using trained NN with the help of RST. In fact, the precision of prediction did not improve with addition with RST. The second method was to show that the training process is more stable when using the hybrid engine than when using only the NN engine. Stablility means that the number of iterations used to train is about the same for every training run.

Table 1. Evaluation of hybrid NN for different simulation sizes

Serial number	Times of Simulation Process	The absolute error value of prediction				
		No error	1	2	3	4
1	10	26(52%)	19(38%)	5(10%)	0	0
2	20	36(72%)	12(24%)	2(4%)	0	0
3	30	27(54%)	23(46%)	0	0	0
4	40	44(88%)	5(10%)	1(2%)	0	0
5	50	39(78%)	11(22%)	0	0	0
6	60	42(84%)	8(16%)	0	0	0
7	70	46(92%)	4(8%)	0	0	0
8	80	45(90%)	5(10%)	0	0	0
9	90	42(84%)	8(16%)	0	0	0
10	100	46(92%)	4(8%)	0	0	0

Method 1: Check the accuracy of hybird NNs prediction by using different numbers of shuffle playlists used in the simulation

Step 1: The system generates data from 100 playlists by simulation.

Step 2: By using the simulated data, we train the NN and record after training, the weights and threshold values into the database. Every neurons has it own weights for each input and one theshold value. For example, if the neuron has 11 inputs, there will be 11 weights and 1 threshold value.

Step 3: The system removes the older simulated data, and repeats the simulation process with difference number of iteration in the simulation process.

Step 4: The trained NN is used to predict the level of songs by using the new data from step 3.

The absolute error between predicted value and desired value are analyzed in Table 1. This table is based on 50 songs. The numbers shown in third to seventh columns is the number of songs out of 50 and the percentage. For example, in first row and third column, the number is 26, so that is 52% of the songs. $The\ absolute\ error\ value\ of\ prediction = |Predicted\ value - Desired\ value|$.

From Table 1, we can see that the NN engine predicts with fewer errors, with increasing number of iterations of the simulation process. For example, in row 7, there are 46 songs out of 50 with no error when we have simulation times of 70. When the sample data are small, the accuracy of NNs prediction is variable as shown in rows 1, 2 and 3. When the sample size is small, the pattern is not clear. In comtrast, considering rows 5 to 10 with larger sample size, the pattern in the data is more clear.

Referring to Table 1 again, columns 4, 5, 6 and 7 indicate how many errors occurred as absolute error value increased. Recall that song level is a measure of how much the user likes this song. For example, if the level of a song is 5, but the prediction is 1, then the error will be 4. Errors of magnitude 4 are shown in column 7. In the table, we can see that no errors of such magnitude occurred.

We are now discussing the second method to evaluate hybrid engine. By using Rough Sets we core the data, which means we come to know which condition attributes are core attributes. None of the core attributes are redundant meaning that any additional attribute will provide no additional classification power. The core data are applied to NN to make it converge more quickly.

Method 2: Comparison of NN hybrid RS engine and NN engine by itself with fixed size simulation data

Step 1: The system does the simulation process with fixed size 100 times playlists. By "100 times", we mean that the simulation process generates 100 playlists.

Step 2: By using these simulation data, we train the two engines (the hybrid engine and the NN by itself) and record how many iterations each of them needs for convergence in the form of one row in Table 2.

Step 3: Go back to step 2 15 times. Now Table 2 has been generated.

Table 2 indicates that NN hybrid RS stabilizes with less iterations than NN by itself. This can be explained as follows: In NN engine, weights are initialized randomly. That is the reason why NN engine is unstable. When the randomly initialized weights are so far from what we really need, NN engine will take a long time to converge. However, with NN hybrid RS engine, the RS engine will control the initialization and weight training process of NN by using core characteristics.

In NN hybrid RS engine, the weight of any unnecessary condition attributes will be initialized to 0, if it is the first time to run the engine. The learning rates of the weight of these condition attributes are also set to 0. It means that no changes to such weights are made when the engine does weight training. This means that unnecessary condition attributes will have no impact on the

Table 2. Comparison of NN hybrid RS and NN with fixed size simulation data

Serial number	Times of Simulation Process	ANN hybrid RS		ANN	
		Number of Iteration	Errors	Number of Iteration	Errors
1	100	27	0.0000003	59	0.000009
2	100	25	0.0000001	38	0.000010
3	100	18	0.0000000	2486	0.000010
4	100	26	0.0000001	44	0.000009
5	100	22	0.0000005	28	0.000008
6	100	26	0.0000001	93	0.000010
7	100	23	0.0000000	45	0.000010
8	100	35	0.0000008	75	0.000008
9	100	40	0.0000002	46	0.000006
10	100	24	0.0000007	2	0.000005
11	100	23	0.0000009	126	0.000010
12	100	31	0.0000006	3852	0.000010
13	100	25	0.0000001	37	0.000003
14	100	31	0.0000000	236	0.000010
15	100	25	0.0000002	1275	0.000010

weight training process. The chance that the system needs to converge with worse initial weights is reduced with fewer attributes. It will improve the stability of the system. The stability can influent the performance of the APP. The hybrid engine is going to be active when the APP is working. The stability of hybrid engine will ensure the stability of the APP.

6 Conclusions

In this project, we built a hybrid engine to learn user preferences for music. The engine is based on a multilayer neural net to make the learning process more effective. Rough Set (RS) is a method to analysis data characterized by imprecision, inconsistency and incompleteness. RS was used during the learning process to help train the weights of the neural net. We evaluated the engine by using simulation data. The simulation data were generated based on specific rules, and the hybrid engine was able to learn the patterns input to data by those rules. The hybrid engine was designed to be portable, so that it can be used not only in this project, but also to solve other real life problems. The engine provides an effective way to run artificial neural network and rough set engines separately or in combination.

This project reveals a new way for predicting users favored playlists. It is not like iTunes of Apple Company. We do not have a big database and big server to analyze all customers data and draw the conclusions. In contrast, we focus on analysis of each specific user. Each APP has its own database. The APP needs to learn the user habits based only on this particular users operations. Consequently, every user has his own specific ANN for predicting their favorite

songs. Each hybrid engine will be custom tailored for each user. The more a user uses the APP, the more the APP will fit him or her. Our APP has been accepted and published in APP Store, and people are currently using it. There is a function in our APP to collect feedback from users. The future work is to let users decide whether this engine is good based on their experience with the APP, and we can improve the APP by considering their feedback.

In artificial neural network hybrid rough set domain, we have described implementation details of the hybridization. Elsewhere, the authors are using some third-party software to combine RS and ANN. However, we implement the RS and ANN engine in its entirety in objective C, so we can do the hybridization at the coding level. We use rough sets core characteristics to guide NNs learning process in the code. The result shows that this hybrid approach improves the stability of training the NN when compare with using NN by itself. However, the accuracy of prediction depends mainly on the NN and not on the Rough Set component of hybrid engine.

To summarise the findings, the hybrid engine predicts with fewer errors, with increasing number of iterations of the simulation process. When the number of iterations of the simulation process is greater, the pattern in the data will be more clear. Generally, the hybrid engine in which insignificant attributes are not considered stabilizes with fewer iterations than NN by itself in which weights of all attributes are initialized randomly.

References

1. Barrington, L., Oda, R., Lanckriet, G.: Smarter than genius? human evaluation of music recommender systems. In: Symposium on Music (2009)
2. Waston, A.: The world according to iTunes: mapping urban networks of music production. Global Networks 12(4), 446–466 (2012)
3. Chuang, C., Huang, S.: A hybrid neural network approach for credit scoring. Expert System 28(2), 185–196 (2011)
4. Shen, Y., Li, T., Hermans, E., Ruan, D., Wets, G., Vanhoof, K., Brijs, T.: A hybrid system of neural networks and rough sets for road safety performance indicators. Soft Computing 14(12), 1255–1263 (2009)
5. Negnevitsky, M.: Artificial neural networks. In: Artificial Intelligence: A Guide To Intelligent Systems, 2nd edn., pp. 165–217. Addison-Wesley (2005)
6. Bryson, A.E., Ho, Y.C.: Applied Optimal Control. Blaisdell, New York (1969)
7. Haykin, S.: Neural networks: A comprehensive foundation, 2nd edn. Prentice Hall (1999)
8. Zdzislaw, P.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
9. Own, H.S., Abraham, A.: A new weighted rough set framework based classification for Egyptian NeoNatal Jaundice. Applied Soft Computing 12(3), 999–1005 (2012)
10. Frank, E., Witten, I.: Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmannn, San Francisco (2005)
11. Zhang, G., Eddy Patuwo, B., Hu, M.Y.: Forecasting with artificial neural networks. International Journal of Forecasting 14(1), 35–62 (1998)

Author Index

Abdullah, Salwani 374
Alqudsi, Arwa 374
Alsolami, Fawaz 154
Alusaifeer, Tariq 251
Azam, Nouman 145

Błaszczyński, Jerzy 133
Bouguila, Nizar 354, 364

Chikalov, Igor 154
Clark, Patrick G. 41
Csajbók, Zoltán Ernő 99

Deng, Dayong 229
Deng, Xiaofei 313

Fan, Wentao 364

Gou, Guanglei 397
Grześ, Tomasz 263
Grzymała-Busse, Jerzy W. 41

Henry, Christopher J. 251
Hirano, Shoji 188
Hu, Qinghua 178
Huang, Bing 325
Huang, Houkuan 229
Huang, Pingli 221
Huang, Zhiqiu 338

Inuiguchi, Masahiro 133, 166

Jalab, Hamid 374
Janicki, Ryszard 87
Jia, Xiuyi 279, 338
Johnson, Julia 405

Kato, Yuichi 213
Kopczyński, Maciej 263
Kusunoki, Yoshifumi 133

Lenarčič, Adam 87
Li, Hong 397
Li, Huaxiong 325
Li, Jianlin 313
Li, Leijun 178
Li, Mei-Zheng 109
Li, Ping 271
Li, Tianrui 240, 291
Li, Weiwei 338
Liang, Decui 291
Liu, Dun 291, 325
Liu, Youli 279
Liu, Yu-Ao 221

Meina, Michał 385
Mihálydeák, Tamás 99
Mizuno, Shoutarou 213
Moshkov, Mikhail 154

Nakata, Michinori 7
Nguyen, Hung Son 385

Ohki, Motoyuki 166

Pan, Lei 279
Pan, Yi 240
Pei, Minghua 229
Peters, James 251

Ramanna, Sheela 251
Rząsa, Wojciech 41

Saeki, Tetsuro 213
Sakai, Hiroshi 7
Sallay, Hassen 364
Shaker, Khalid 374
Shang, Lin 271
Ślęzak, Dominik 200
Słowiński, Roman 133
Stawicki, Sebastian 200
Stepaniuk, Jarosław 263
Su, Weijia 354
Świeboda, Wojciech 385
Szczuka, Marcin 1

Tsumoto, Shusaku 188

Wang, Chongjun 279
Wang, Guoyin 53, 109, 397

Wang, Jin 109, 221
Wang, Jingqian 75
Wong, S.K.M. 66
Wu, Jianyang 271
Wu, Mao 7
Wu, Xiangqian 178

Xie, Junyuan 279
Xing, Hang 346
Xu, Changlin 53

Yamaguchi, Naoto 7
Yao, JingTao 28, 145
Yao, Yiyu 16, 121, 313
Yu, Daren 178
Yu, Hong 53, 302
Yun, Bo 221

Zhang, Junbo 240
Zhang, Yan 28
Zhang, Yanping 346
Zhao, Shu 346
Zhou, Bing 121
Zhou, Qingfeng 302
Zhou, Xianzhong 325
Zhu, William 75
Zhu, Yun 240
Ziou, Djemel 354
Zou, Huijin 346
Zou, Xuan 397
Zuo, Hongming 405