ROUGH SETS AND DATA MINING

Analysis for Imprecise Data

ROUGH SETS AND DATA MINING

Analysis for Imprecise Data

T. Y. LIN
San Jose State University
San Jose, California, USA

N. CERCONE
University of Regina
Regina, Saskatchewan, CANADA

KLUWER ACADEMIC PUBLISHERS
Boston/London/Dordrecht

Distributors for North America:
Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061 USA

Distributors for all other countries:
Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

ISBN-13: 978-1-4612-8637-0 e-ISBN-13: 978-1-4613-1461-5
DOI: 10.1007/978-1-4613-1461-5

CONTENTS

PREFACE

We were inspired to compile this book based on a speech delivered by Professor Zdzislaw Pawlak, the creator of rough set theory, at the 1995 ACM Computer Science Conference (CSC '95). This book is based on some of the papers presented at this workshop; but in all cases, the papers have been reviewed, revised and expanded in to chapters. The chapters include newer research results and applications for the mining of databases.

As stated in the workshop program, "Database Mining can be defined as the process of mining for implicit, previously unknown, and potentially useful information from very large databases by efficient knowledge discovery techniques. Consequently, this is proving to be one of the most promising research areas in the fields of artificial intelligence and database systems. There is considerable excitement surrounding these developments and numerous commercial initiatives are under way to make use of rough sets for extracting useful information from databases. This book demonstrates the research and applications that can help point the way for other researchers working in this growing field as well as help database designers and developers better utilize rough sets as a powerful mining tool.

We will take this opportunity to thank our distinguished contributors for developing these chapters for this work. We feel that this book will become an essential handbook for rough set researchers and database designers and developers. Researchers new to this field will also find this book an invaluable reference because of the concise introductions and thorough explanations provided by the authors.

T. Y. Lin and Nick Cercone

ROUGH SETS AND DATA MINING

Analysis for Imprecise Data

PART I

EXPOSITIONS

1

ROUGH SETS

Zdzisław Pawlak

Institute of Computer Science,
Warsaw University of Technolgy, Warsaw 00-665, Poland,
ul. Nowowiejska 15/19, zpw@ii.pw.edu.pl

The concept of the rough set is a new mathematical approach to imprecision, vagueness and uncertainty in data analysis.

The starting point of the rough set philosophy is the assumption that with every object of interest we associate some information (data, knowledge). E.g., if objects are patients suffering form a certain disease, symptoms of the disease form information about patients. Objects are similar or indiscrenible, if they are characterized by the same information. The indiscernibility relation generated thus is the mathematical basis of the rough set theory.

Set of all similar objects is called elementary, and form basic granule (atom) of knowledge. Any union of some elementary sets is referred to as crisp (precise) set – otherwise a set is rough (imprecise, vague).

As a consequence of the above definition each rough set have boundary-line elements, i.e., elements which cannot be with certainty classified as members of the set or its complement. (Obviously crisp sets have no boundary-line elements at all). In other words boundary-line cases cannot be properly classified employing the available knowledge. Thuse rough sets can be viewed as a mathematical model of vague concepts.

In the rough set approach any vaque concept is characterized by pair of precise concepts – called the lower and the upper approximation of the vague concept. The lower approximation consists of all objects which surely belong to the concept and the upper approximation contain all objects which possible belong to the concept. Approximations constitute two basic operations in the rough set approach.

The above presented ideas can be illustrated by the following example. Suppose we are given data table – called also attribute-value table or information system – containing data about 6 patients, as shown below.

Patient	Headache	Muscle-pain	Temperature	Flu
p1	no	yes	high	yes
p2	yes	no	high	yes
p3	yes	yes	very high	yes
p4	no	yes	normal	no
p5	yes	no	high	no
p6	no	yes	very high	yes

Columns of the table are labelled by attributes (symptoms) and rows by objects (patients), whereas entries of the table are attribute values. Thus each row of the table can be seen as information about specific patient. For example patient p2 is characterized in the table by the following attribute-value set

{(Headache, yes), (Muscle-pain, no), (Temperature, high), (Flu, yes)},

which form information about the patient.

In the table patients p2, p3 and p5 are indiscernible wiht respect to the attribute Headache, patients p3 and p6 are indiscernible with respect to attributes Muscle-pain and Flu, and patients p2 and p5 are indiscernible with respect to attributes Headache, Muscle-pain and Temperature. Hance, for example, the attribute Headache generates two elementary sets {p2, p3, p5} and {p1, p4, p6}, whereas the attributes Headache and Muscle-pain form the following elementary sets, {p1, p4, p6}, {p2, p5} and {p3}. Similary one can define elementary set generated by any subset of attributes.

Because patient p2 has flu, whereas patient p5 does not, and they are indiscernible with respect to the attributes Headache, Muscle-pain and Temperature, thus flu cannot be characterized in terms of attributes Headache, Muscle-pain and Temperature. Hance p2 and p5 are the boundary-line cases, which connot be properly classified in view of the available knowledge. The remaining patients p1, p3 and p6 display symptoms which enable us to classify them with certainty as having flu, patients p1 and p5 cannot be excluded as having flu and patient p4 for sure has not flu, in view fo the displayed symptoms. Thus

the lower approximation for the set of patients having flu is the set {p1, p3, p6} and the upper approximation of this set is the set {{p1, p2, p3, p5, p6}. Similary p4 has not flu and p2, p5 can not be excludes as having flu, thus the lower approximation of this concept is the set {p4} whereas – the upper approximation is the set {p2, p4, p5}.

We may also ask whether all attributes in this table are necessary to define flu. One can easily see, for example that, if a patient has very high temperature, he has for sure flu, but if he has normal tempetrature he has not flu whatsoever.

In general basic problems which can be solved using the rough set approach are the following:

1) description of set of objects in terms of attribute values

2) dependencies (full or partial) between attributes

3) reduction of attributes

4) significance of attributes

5) decision rules generation

and others.

The rough set methodology has been applied in many real-life applications and it seems to be important to machne learning, decision analysis, knowledge discovery, expert systems, decision support systems, pattern recognition and others.

Some current research on rough controllers has pinted out a new very promising area of applications of the rough set theory.

The rough set concept coincided with many other mathematical models of vagueness and uncertainty – in particular fuzzy sets and evidence theory – but it can be viewed in its own rights.

REFERENCES

[1] Grzymała-Busse J.W., (1991), *Managing Uncertainty in Expert Systems.* Kluwer Academic Publishers, Dordrecht, Boston, London.

[2] Lin, T.Y., (ed.), (1994), *The Third International Workshop on Rough Sets and Soft Computing Proceedings (RSSC'94)*, San Jose State University, San Jose, California, USA, November 10–12.

[3] Pawlak Z., (1982), "Rough sets". *International Journal of Computer and Information Sciences*, 11, 341–356.

[4] Pawlak Z., (1991), *Rough Sets - Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Dordrecht, Boston, London.

[5] Pawlak Z., Grzymała-Busse J. W., Słowiński R., and Ziarko, W., (1995),"Rough sets", *Communication of the ACM*, 38, 88–95.

[6] Słowiński, R., (ed.), (1992), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory*, Kluwer Academic Publishers, Dordrecht.

[7] Ziarko, W., (ed.), (1993), *Rough Sets, Fuzzy Sets and Knowledge Discovery. Proceedings of the International Workshop on Rough Sets and Knowledge Discovery (RSKD'93)*, Banff, Alberta, Canada, October 12–15, Springer-Verlag, Berlin.

Biographical Sketch

Zdzisław I. Pawlak is Professor of Computer Science and Member of the Polish Academy of Sciences. He is head of the Group for Algorithmic Method of Reasoning in the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences and Director of the Institute of Computer Science, Warsaw University of Technology. Current research interests include intelligent systems and cognitive sciences, in particular, decision support systems, reasoning about knowledge, machine learning, inductive reasoning, vagueness, uncertainty and conflict analysis. **Author's Present Address:** Institute of Computer Science,

Warsaw University of Technolgy, Warsaw 00-665, Poland, ul. Nowowiejska 15/19, zpw@ii.pw.edu.pl

2

DATA MINING: TRENDS IN RESEARCH AND DEVELOPMENT

Jitender S. Deogun, Vijay V. Raghavan*, Amartya Sarkar*, and Hayri Sever**

The Department of Computer Science and Engineering,
University of Nebraska-Lincoln
Lincoln, NE 68588, USA

** The Center for Advanced Computer Studies*
University of Southwestern Louisiana
Lafayette, LA 70504, USA

*** The Department of Computer Science*
Hacettepe University,
Beytepe, Ankara 06532, TR

ABSTRACT

Data mining is an interdisciplinary research area spanning several disciplines such as database systems, machine learning, intelligent information systems, statistics, and expert systems. Data mining has evolved into an important and active area of research because of theoretical challenges and practical applications associated with the problem of discovering (or extracting) interesting and previously unknown knowledge from very large real-world databases. Many aspects of data mining have been investigated in several related fields. But the problem is unique enough that there is a great need to extend these studies to include the nature of the contents of the real-world databases. In this chapter, we discuss the theory and foundational issues in data mining, describe data mining methods and algorithms, and review data mining applications. Since a major focus of this book is on rough sets and its applications to database mining, one full section is devoted to summarizing the state of rough sets as related to data mining of real-world databases. More importantly, we provide evidence showing that the theory of rough sets constitutes a sound basis for data mining applications.

1 INTRODUCTION

It is estimated that the amount of information in the world doubles every 20 months [1]; that is, many scientific, government and corporate information systems are being overwhelmed by a flood of data that are generated and stored routinely, which grow into large databases amounting to giga (and even tera) bytes of data [2]. These databases contain potential gold mine of valuable information, but it is beyond human ability to analyze such massive amounts of data and elicit meaningful patterns. Given certain data analysis goal, it has been a common practice to either design a database application on on-line data or use a statistical (or an analytical) package on off-line data along with a domain expert to interpret the results. Even if one does not count the problems related with the use of standard statistical packages (such as its limited power for knowledge discovery, the needs for trained statisticians and domain experts to apply statistical methods and to refine/interpret results, etc.), one is required to state the goal (i.e., what kind of information one wishes to extract from data) and gather relevant data to arrive at that goal. Consequently, there is still strong possibility that some significant and meaningful patterns in the database, waiting to be discovered, are missed.

As often argued in the literature it is desirable to pursue a more general goal, which is to extract implicit, previously unknown, hidden, and potentially useful information from raw data in an automatic fashion, rather than developing individual applications for each user need. Unfortunately, the database technology of today offers little functionality to explore data in such a fashion. At the same time KD techniques for intelligent data analysis are not yet mature for large data sets [3]. Furthermore, the fact that data has been organized and collected around the needs of organizational activities may pose a real difficulty in locating relevant data for knowledge discovery techniques from diverse sources. The *data mining*[1] problem is defined to emphasize the challenges of searching for knowledge in large databases and to motivate researchers and application developers for meeting that challenge. It comes from the idea that *large databases* can be viewed as data mines containing valuable information that can be discovered by *efficient* knowledge discovery techniques.

This chapter is organized as follows. In the Section 2, we discuss the fact that data mining is an interdisciplinary research area. In Section 3, current research on theoretical issues in data mining including data and knowledge representation, probabilistic modeling and uncertainty management, and metrics for

[1] In the literature, data mining problem is also known as *database mining* or *the knowledge discovery in databases (KDD)*. Some researchers view KDD as a broader discipline, with data mining as one component dealing with knowledge discovery methods [4].

evaluation of data mining results is summarized. In Section 4, we classify data mining queries into four categories: data dependency, classification, clustering and characterization. A variety of data mining methods available to handle each of these query classes are presented. In Section 5, the focus is on the state of rough set methodology in the context of data mining and discuss research directions in rough set theory to make the rough set model suitable for data mining applications. In Section 6, we review data mining systems and tools. In Section 7, recommendations for future research directions in rough set based approaches to data mining are presented.

2 A PERSPECTIVE ON DATA MINING AND RELATED RESEARCH AREAS

Data mining is a promising interdisciplinary area of research shared by several fields such as database systems, machine learning, intelligent information systems, statistics, data warehousing and knowledge acquisition in expert systems [4]. It may be noted that data mining is a distinct descipline and its objectives are different from the goals and emphases of the individual fields. Data mining may, however, heavily use theories and developments of these fields [5, 3, 6, 7, 8]. In the following we present basic differences (and/or similarities) between data mining and various allied research areas.

In developing database systems to manage uncertain (or imprecise) information as well as certain (or precise) information, several extensions to relational model have been suggested [9, 10, 11]. The direction of such extensions include data representation as well as basic relational operations. In Lee's approach [9], the uncertainty associated with an attribute (treated as random variable) is represented using a probability distribution on the power set (basic probability assignment) of its domain instead of an atomic value, while a set of values is allowed for the representation of imprecise data. For each tuple in a relation, a system attribute consisting of a pair of belief and plausibility values is attached to show confidence level in that tuple. With this representation, the traditional null value is handled naturally by subdividing it into three cases such as unknown, inapplicable, and unknown or inapplicable. Lee has extended the Dempster-Shafer theory to handle the comparison of two independent basic probability assignments so that condition criteria involving independence of relational operations can be covered. Since the concern is to capture only the uncertainty in the data, Barbara et al. have associated discrete probabilistic functions with the values of attributes [11]. An attribute in a relation may be

deterministic or probabilistic in nature, while keys must be deterministic, which is a restriction imposed by the authors leading to simple relational operators. Probability values associated with the range of an attribute in a tuple should add to one and are either entered into the database system as confidence or belief values or computed from underlying sample. Barbara et al. have incorporated missing probability, denoted by a wildcard symbol, so that the uninteresting range of values are eliminated and it facilitates the insertion of data into a relation without knowing all information about probabilistic measures of an attribute's domain. In this model, called probabilistic relational algebra, basic probability theory, under the assumption of conditional independence, is used to extend relational operations with the drawback that missing probabilities involved in a join operation causes "information loss" Studies in either approximate queries or in uncertainty modeling may not be directly linked to the problem of data mining, but certainly provide a sound basis for the knowledge discovery process. For example, identifying probabilistic relationships in data can be useful in discovering functional or production-rule relationships in the data.

The last few years have seen an increasing use of techniques in data mining that draw upon or are based on statistics; namely, in feature selection [12], data dependency involving two variables for constructing data dependency networks [13, 14], classification of objects based on descriptions [7], discretization of continuous values [13, 15], data summarization [14], predicting missing values [16], etc. The motivation behind this trend can be explained by the fact that statistical techniques for data analysis are well developed and in some cases, we do not have any other means to apply. In many data analysis problems statistical methods are, however, not suitable either because of strong statistical assumptions, such as adherence to a particular probability distribution model, or due to fundamental limitations of the statistical approach. The primary limitation is the inability to recognize and generalize relationships, such as the set inclusion, that capture structural aspects of a data set, as a result of being entirely confined to arithmetic manipulations of probability measures [17, 18]. The chi-square test is used, for example, by some decision-tree based systems during tree pruning to determine whether a node should be branched [19]. It is also used to select a good set of features with which to perform the learning process [20]. Despite its popularity, it should be noted that the chi-square test only tells us whether an attribute, as a whole, is helpful in determining the class membership of an object. It does not, however, provide us with much information about whether an object characterized by certain values should be assigned to a particular class.

In the earlier work on machine learning, a number of theoretical and foundational issues of interest to data mining (e.g., learning from examples, formation of concepts from instances, discovering regular patterns, noisy and incomplete data, and uncertainty management, etc.) have been investigated. Data mining problem simply combines all aspects of knowledge discovery in the context of ultra large data. More specifically, data mining is the process of deriving rules, where a database takes on the role of training data set. In other words, a data mining application distinguishes itself from a machine learning problem, in the sense that available techniques must be extended to be applicable to uncontrolled, real world data. That is, one does not have the luxury of specifying the data requirements from the perspective of knowledge discovery goals before collecting the data.

It may furthermore be worth pointing out that the connection of the data mining problem to a database is loosely defined because of the terminological gap between artificial intelligence (AI) and database communities on perceiving what a database is; that is, the researchers in database systems think of a database as a collection of interrelated data within a database management system, while the researchers in AI consider it as a simple file structure or an off-line data collection, e.g., a single relation in a relational database. Therefore, the nature of the problem depends on the context that one intends to target. If the knowledge model is integrated/related to a data base within a DBMS, then it should also address issues related to the management of data such as data security, viewing levels of data, transaction management, and the use of general database functions/facilities [1, 3].

3 THEORETICAL AND FOUNDATIONAL ISSUES

The data (or instance space) is represented by a relation , which is the predominant structure adopted in either machine learning or database systems. Each tuple in a relation corresponds to an entity (also known as object, instance or background fact). Entities are made up of attributes (also called fields or features). The given data set is divided into a training and a test set. The training set is then used to generate some knowledge and the test set is used to determine validity of and/or to refine that knowledge. In this section, we emphasize theoretical and foundational issues related to the very nature of real-world data from the perspective of knowledge discovery in databases.

3.1 Ultra Large Data

One of the important issues in data mining is related to the volume of data, because many knowledge discovery techniques, involving exhaustive search over instance space, are highly sensitive to the size of data in terms of time complexity and inducing compact patterns. For example, candidate elimination algorithm [21], a tuple oriented learning technique from examples, aims to search the version space, whose size is doubly-exponential in the number of attributes, of training examples to induce a generalized concept that is satisfied by all of the positive examples and none of the negative examples. Hence the data driven techniques either rely on heuristics to guide their search through the large space of possible relations between combinations of attribute values and classes or reduce their search space horizontally or vertically.

Horizontal reduction is related to merging identical tuples following either the substitution of an attribute value by its higher level value in a pre-defined generalization hierarchy of categorical values of the attribute [22] or the quantization (or discretization) of continuous (or numeric) values [13, 15, 23]. Vertical reduction is realized by either applying some *feature selection* methods or using attribute dependency graph [24]. We consider vertical reduction as a part of methods for handling redundant data, in Section 3.5. We elaborate on some notable studies on horizontal reduction in the following.

The simplest discretization procedure is to divide the range of a continuous variable into equal-width intervals as many as a user-defined number of intervals. A variation of that method is the use of Shannon's entropy theory such that the entropy scheme determines the interval boundaries by making the total gain of information from the observed occurrences in each interval equal. This procedure is called 'even information intervals quantization' method [25]. The obvious drawback of such a procedure is that there may be a large amount of information loss, because the cut points would not necessarily be on boundaries of pre-defined classes. In other words, their criteria of discretization fail to take into consideration the relationship between pre-assigned classes and interval boundaries. Both Ching et al. [23] and Fayyad & Irani [15] suggest class dependent discretization algorithms. Note that the whole idea here is to reduce the number of attribute values without destroying the interdependence relationship between the class and attribute values.

Class-dependent discretization of Ching et al. [23] consists of three main processes: interval initialization, interval improvement, and interval reduction. In the first process, after an initial default number of intervals are selected, a de-

scription of intervals, called the boundary set consisting of ordered end points of intervals, are determined such that the sample is distributed over intervals as evenly as possible. The maximum entropy criterion is used to minimize the information loss. The boundary improvement process, which uses an interdependence criterion given by a normalized class-attribute mutual information, considers all possible local adjustments on the boundary set to ensure a good estimation of global optimal interdependence. The last process combines statistically insignificant intervals.

Fayyad & Irani in [15] formally prove that the information entropy minimization criterion of ID3, used for binary splitting of continuous valued attributes, always selects a value between two examples of different classes in the sequence of sorted examples with respect to increasing order of that attribute values, i.e., the selected value is actually a boundary point. Note that there is a side benefit of this result, from the point of view of efficiency, since the algorithm needs only to examine a small number of boundary points polynomially related to the number of classes rather than all distinct values of continuous variable. The binary splitting method is generalized using divide-and-conquer principle; that is, the algorithm is applied recursively to select the boundary values once the training set is sorted. A criterion is applied to decide when to refrain from applying further binary splitting to a given interval. Given a potential binary partition π_T on a current training set S, let HT be a hypothesis induced by π_T if it were accepted; and let NT be the null hypothesis. Then $\{HT, NT\}$ are two states of a binary decision problem that decides whether or not to recognize the partition π_T,. Such a problem can be expressed in terms of Bayesian decision strategy, involving, for example, probability-of-error criterion. The decision criterion has been estimated using the minimum description length principle (MDLP) [26].

3.2 Noisy Data

Non-systematic errors, which can occur during data-entry or collection of data, are usually referred to as *noise*. Unfortunately there is little support by commercial DBMSs to eliminate/reduce errors that occur during data entry, though the potential exists for providing such capability in relational data models, to force consistency among attribute values with respect to predefined functional dependencies. Hence, erroneous data can be a significant problem in real-world databases. This implies that a knowledge discovery method should be less sensitive to noise in the data set. This problem has been extensively investigated

for variations of inductive decision trees, depending on where and how much the noise occurs [27].

If a training sample is corrupted with noise, the system should be able to identify and ignore it. Presence of noise in the class information of training set affects the accuracy of generated rules; hence an attempt should be made to eliminate noise that affects the class information of the objects in the training set. Quinlan [27] has performed experiments to investigate the effect of noise on classifying examples from the test set. The experimental results indicate that for some systems adding substantial noise to the data results in low level of misclassification of unseen examples (test set). It has also been observed that rules learned from corrupted training set perform better in classifying noisy test data than rules that are learned from noise free training set. Chan and Wong [7] have used statistical techniques to analyze the effect of noise. Their solution involves estimating the class conditional density in presence of noise, comparing it with the true class density and then determining a classifier whose level of confidence is set accordingly.

3.3 Null Values

In DBMSs, a null value (also known as missing value) may appear as the value of any attribute that is not a part of the primary key and is treated as a symbol distinct from any other symbol, including other occurrences of null values. The null value does not only mean an *unknown* value, but also can mean *inapplicable.* In relational databases this problem occurs frequently because the relational model dictates that all tuples in a relation must have the same number of attributes, even if values of some attributes are inapplicable for some tuples. For example, in the list of personal computers, the attribute that contains the model type of the sound cards would be null for some model of computers.

Lee provides an approach to extend relational database model for uncertain and imprecise information [9], where the traditional null value is handled by subdividing it into three cases such as unknown, inapplicable, and unknown or inapplicable. Other than this work, which does not offer any solution for existing data, we have not come across any work that deals with null values, though there are some recent studies on unknown values [28, 29, 30]. When the database contains missing attribute values, either the values can be discarded or an attempt can be made to replace them with the most likely values [19]. These are the ideas adopted by Quinlan [19] for inductive decision trees. In [31] it is suggested to construct rules that predict the value of the missing attribute,

based on the value of other attributes in the example, and the class information. These rules can then be used to "fill in" the missing attribute values and the resulting data set could be used to construct the descriptions.

Grzymala-Busse [29], citing the drawbacks of the approaches given above, has transformed a given decision table with unknown values to a new and possibly inconsistent decision table, in which every attribute value is known, by replacing the unknown value of an attribute with all possible values of that attribute. In other words, he reduced the missing value problem to that of learning from inconsistent examples. He, then, used rough set theory to induce certain and possible rules. Using similar line of interpretation of missing values, Barbara et al. in [11] have interpreted missing values as uninteresting values of an attribute with which they have associated missing probability measures. Probabilistic relational operations would yield certain or possible probabilities (lower or upper bounds on the probability of a random variable) depending on whether missing probabilities are facilitated, or not. In [30], the problem of missing value is solved using the EM algorithm. The EM algorithm assumes that the missing values are missing at random, but the importance of this method lies in its underlying message— even when the data is complete, it is often useful to treat the data as a missing value problem for computational purposes [16].

3.4 Incomplete Data

Suppose each object in the universe of discourse is described or characterized by the values of a set of attributes. If the description of the individual objects are sufficient and precise enough with respect to a given concept, one can unambiguously describe the class, a subset of objects, representing the concept.

However, the available knowledge in many practical situations is often incomplete and imprecise. The fact that data has been organized and collected around the needs of organizational activities causes incomplete data from the view point of the knowledge discovery task. Under such circumstances, the knowledge discovery model should have the capability of providing approximate decisions with some confidence level.

Many methods were proposed to deal with the approximation of a concept. For example, the well-known fuzzy set theory characterizes a concept *approximately* by a membership function with a range between 0 and 1. Another approach is based on the rough set theory which provides the lower and upper approximations of a concept depending on how relationship between two

different partitions of a finite universe of discourse is defined. If this relationship is probabilistic in nature, Wong and Ziarko [32] demonstrated that the generalized notion of rough sets can indeed be conveniently described by the concept of fuzzy sets when proper fuzzy set operations are employed. In a related study [33], Wong and Yao introduced a Bayesian decision theoretic framework which provides a plausible unification of the fuzzy set and rough set approaches for approximating a concept. Particularly they show that if a given concept is approximated by positive and negative regions of that concept, the same result given by the $\alpha-$cut in the fuzzy set theory is obtained. We explain how the rough set approach reasons about incomplete data in Section 5, which is devoted to the state of rough sets in the context of data mining. In the rest of this subsection, we review work on inductive decision trees aimed at making them suitable for incomplete data.

ID3-like algorithms [19, 34, 35], during the process of inducing decision trees as well as of refining induced decision trees, implicitly assume that enough information is available in the data to decide exactly how each object should be classified. In other words, there is a single correct label for any given combination of attribute values, describing objects, in the training set. Hence, for some time, inconclusive objects in a training set, i.e., objects having the same description and yet different class labels, have been interpreted as noise either in their descriptions or in their labels. Uthurusamy et al. in [36] have argued that this assumption is not valid in the first place on the ground that inconclusive data sets are different from noisy data set, especially when descriptions of objects are incomplete to arrive at certain conclusions. The INFERRULE algorithm of Uthurusamy et al. improves ID3-like methods essentially around this issue.

In particular, they have proposed a controlled feature selection measure , say R, to generate inductive decision trees such that INFERRULE stops specializing (or partitioning a node) further whenever R exceeds a threshold value and returns a probabilistic guess of possible classes. INFERRULE selects the best attribute-value pair, rather than the best attribute, in order to avoid unnecessary divisions of the data set that becomes problematic when an attribute has many values and only a few of them are relevant to the class labels. For a given value a_i of an attribute A, let us define two vectors made up of estimated and actual joint distribution of a_i and class labels over the data set, respectively. The attribute-value pair selection measure R is based on minimizing the proportion of standard error in estimating joint distribution over the geometric distance between these two vectors. The selection measure R indicates that the class distribution in its selected subset differs significantly from the class distribution in the original training set. Once the best attribute-value is

selected, the training set is split into two groups: one with $A = a_i$ and another with $A \neq a_i$.

3.5 Redundant Data

As opposed to incomplete data, the given data set may contain redundant or insignificant attributes with respect to the problem at the hand. This case might arise in several situations. For example, combining relational tables to gather relevant data set may result in redundant attributes that the user is not aware of, since un-normalized relational tables may involve redundant features in their contents. Fortunately, there exist many near-optimal solutions, or optimal solutions in special cases, with reasonable time complexity that eliminate insignificant (or redundant) attributes from a given attribute set by using weights for either individual attributes or combination of some attributes. These type of algorithms are known as feature selection (or reduction).

Feature selection, a pre-pruning process in inductive learning, is the problem of choosing a small subset of features that is necessary and sufficient to describe target concept(s). The importance of feature selection in a broader sense is not only to reduce the search space, but also to speed up the processes of both concept learning and classifying objects and to improve the quality of classification [37, 38, 39, 40]. It is well known that searching for the smallest subset of features in the feature space takes time that is bounded by $O(2^l J)$, where: l is the number of features, and J is the computational effort required to evaluate each subset. This type of exhaustive search would be appropriate only if l is small and J is computationally inexpensive. Greedy approaches like stepwise backward/forward techniques [20, 35], dynamic programming [41], and branch and bound algorithm [42] are non-exhaustive and efficient search techniques, which can be applied with some feature selection criterion. For near-optimal solutions or optimal solutions in special cases, weights of either individual features or combinations of features are computed with respect to some feature selection criteria (or measures) such as Bhattacharya coefficient, divergence, Kolmogorov variational distance, etc., in statistics [43, 44]; Shannon's entropy criterion, classification accuracy, or classification quality based on *dice coefficient* in pattern recognition and machine learning [37, 45, 46].

Projection Pursuit technique can also be used on the data to find "interesting low dimensional projections of a high dimensional point cloud by numerically maximizing a certain objective function or projection index" [47]. These "interesting" projections could then be further analyzed to check for some unspec-

ified, unanticipated structures in the data. The projection pursuit methods are unaffected by the curse of dimensionality; however, they are poorly suited to deal with non-linear structures. Many of the classical multivariate analysis techniques, viz., principal components, factor analysis, discriminant analysis are special cases of projection pursuit method. As a final note, it may be worth pointing out that one could also use random sampling methods [14], along with the horizontal pruning methods [22].

3.6 Dynamic Data

A fundamental characteristic of databases that are online is that they are dynamic; that is, their contents are ever changing. This situation has several important implications for the Knowledge Discovery (KD) method. First, if a knowledge discovery model is implemented as a database application then the run time efficiency of a knowledge discovery method within the KD model and its use of retrieval functions of the DBMS become important factors for the performance evaluation of the KD method, because the KD methods are strictly read-only, long-running transactions. Second, if we regard the knowledge obtained from dynamic data to be persistent, then the knowledge discovery method should have the capability of evolving derived knowledge incrementally as the data changes over time. Active database systems have already provided trigger facilities (or *if-then* action rules) that can be used for implementing incremental knowledge discovery methods.

4 DATA MINING METHODS

Knowledge is usually represented in the form of rules— rules indicating the degree of association between two variables, rules mapping data into predefined classes, rules that identify a finite set of categories or clusters to describe the data, etc. These rules support specific tasks and are generated by repeated application of a certain technique, or more generally an algorithm, on the data. The quality of these rules and hence the knowledge discovered is heavily dependent on the algorithms used to analyze the data. Thus, central to the problem of knowledge extraction are the techniques/methods used to generate such rules.

The core of an algorithm constitutes the model upon which the algorithm is built on. The issue of knowledge representation has been studied in the context

of various models, mainly relational, propositional or restricted first-order logic models. Choosing the appropriate model, realizing the assumptions inherent in the model and using a proper representational form are some of the factors that influence a successful knowledge discovery. For example, an overly powerful representation of the model might increase the danger of overfitting the training data resulting in reduced prediction accuracy on unseen data. In addition the search becomes highly complex and the interpretation of the model becomes difficult.

Model evaluation is concerned with estimating how well a particular model and its parameters meet the criteria of the KDD process. This step may also include the assessment of the relative degree of interest of the extracted patterns and decide which to present and which order. Many measures associated with rules (or knowledge units) have been proposed for model evaluation. Confidence factor (also known as accuracy of a rule) is a quantitative measure reflecting the strength of an induced rule. It is defined as the fraction of objects in a training set that satisfies both the antecedent and consequent parts of the rule. Classification accuracy (or classification error) is the fraction of objects/instances in test data that are incorrectly classified. The specific factors that influence the impact and interestingness of a pattern and hence the criteria of model evaluation will vary for different databases and tasks. In this section we present an overview of the popular methods used to discover patterns (or knowledge) in ultra large data sets in the light of model representation and evaluation.

Data Dependency Query: Data dependencies (also known as functional dependencies) in DBMSs are defined during the design of conceptual schema, whereas in machine learning they are induced from given data. Depending on how data dependencies are perceived, their use in these two disciplines is different. For example, data dependencies in DBMSs ARe used for normalizing relations and indexing relations, whereas in machine learning they are used as a preprocessing step of a knowledge discovery technique to reduce the number of attributes in a given data set, to quantize continuous values of an attribute, for testing a hypothesis (i.e., finding associations among values of certain attributes), or for constructing a data dependency graph.

In KDW [14], Shapiro & Matheus have utilized the idea of probabilistic dependency between two discrete attributes. This information provides the weight and direction of the arc between nodes characterized by the two attributes. An acyclic dependency network has been built based on statistical significance of probabilistic dependencies between pairs of discrete attributes. Concept hierarchies (or more generally dependency networks) are based on a partial ordering of propositions (or predicates), which are usually expressed as unary formulas.

Such structures may be a part of the background knowledge. Han et al. [22], for example, utilize generalization hierarchies of attributes' values in their inductive learning method to characterize a concept or discriminate it from other concepts. In another approach, Zhong & Ohsuga [13] have focused on the conditional distributions of two discrete attributes to form a basis for hierarchical model learning. They have transformed the instance space of two discrete attributes to a probability space, represented by a probability distribution matrix. After diagonalizing this probability distribution matrix, by selecting either a special attribute or a row, concept clusters have been formed. In the process of decomposing the database (i.e., while forming concept clusters) noisy data is filtered out.

It is sometimes useful to determine *associations* among values of an attribute. For example, planning department at a supermarket may like to know if the customer who purchase 'bread' and 'butter' also tends to purchase 'milk', where 'butter', 'bread', and 'milk' are usually part of the same multi-valued attribute of a sales transaction. This type of query along with interval classification has been suggested by Agrawal et al. in [48]. They represent knowledge as a set of rules, denoted by $r : F(o) \Rightarrow G(o)$, where: F is a conjunction of unary formulas, G is a unary formula. Each rule r is associated with a confidence factor c, $0 \leq c \leq 1$, which shows the strength of the rule r. The knowledge units considered in [48] are equivalent to the notion of ID3 trees, except that continuous values are partitioned into intervals in contrast to ID3 that uses binary splitting for this purpose. It is, however, worth pointing out that, given the set of objects O, the rules are generated in a way that they satisfy certain additional constraints of two different forms: *syntactic* and *support* constraints. Syntactic constraints involve restrictions on predicates and methods that can appear in the rule. For example, a user may be interested in all associations that have 'milk' in the consequent and 'bread' in the antecedent. Support constraints concern statistical significance of a rule, which is the fraction of objects in O that satisfy the conjunction of the consequent and antecedent of the rule. Finally, note that we use the dependencies among attributes in their narrow sense; however many data mining queries can, in broader sense, be viewed as an application or variation of data dependency analysis.

Classification Query: This kind of query involves inducing a classification function (also known as inducing a classifier, supervised learning, concept learning or discriminating description of classes) that partitions a given set of tuples into meaningful disjoint subclasses with respect to user defined labels or the values of some decision attributes. When a relation is used as a knowledge structure the set of attributes are partitioned into two groups. The first group is called the set of condition attributes or the feature set, depending on the

application domain. The second group is called the set of decision attributes. A block in the partition induced by the decision attribute(s) is called a concept (or a class). Typically, the IF part is specified by values of condition attributes, while the $THEN$ part identifies a concept. Difference between two classes may be described by discriminating descriptions such as decision trees and decision lists. Many empirical learning algorithms, such as decision tree inducers, neural networks and genetic algorithms are designed to produce discriminating descriptions. This subject has extensively been investigated in the literature [49, 50, 51, 52, 53] and is the primary task in inductive learning.

Note that this type of inductive learning can potentially help in predicting the future. In order to predict the future, known results from the past should be used as much as possible. In experimental environments, the validation of a decision algorithm is accomplished by splitting the available set of labeled samples into training and test sets. The training set is then used to generate a decision algorithm and the test set is used to determine the validity of that decision model. Classification accuracy (or classification error) is then measured as the fraction of objects/instances in test data that are incorrectly classified. There have been indications that the accuracy of a rule (as measured on training set) may not be a good indicator of its accuracy in general [54]. This is especially true on noisy data; DNF concept learners typically learn a few reliable disjuncts and many unreliable disjuncts each of which covers a small number of positive training examples [55]. If the evaluation criterion to derive the decision model is monotonic, then the training error can be controlled [37, 42]. In the process of estimating validation error, the concept of bootstrapping over test set may be used [12, 56]. Note that dividing the samples into training and test sets is an important problem and must be solved in a way that the distributions of the two sets are close to each other. The ratio of the sizes of the training set to the test set is then determined from the bias and the variance of the estimated error [57].

For classification with mixed mode data [23], the mutual information, between a class and an attribute, can be combined to determine the membership of an unknown object under the assumption that the given attributes are independent.

Clustering Query: We call unsupervised partitioning of tuples of a relational table a clustering query (also known as unsupervised learning in the context of inductive learning). There are numerous clustering algorithms ranging from the traditional methods of pattern recognition to clustering techniques in machine learning [43, 58]. User-defined parameters such as the number of clusters or the maximum number of tuples within a cluster can influence the result of

a clustering query. Clustering queries may be helpful for the following two reasons. First, the user may not know the nature or structure of the data. Second, even if the user have some domain knowledge, labeling a large set of tuples can be surprisingly costly and time consuming. Instead, a classifier may be designed on a small, labeled set of samples, and then *tuned up* by allowing it to run without supervision on a large and unlabeled set of tuples. Unfortunately such technique does not work well when the patterns are time varying. Alternatively, interactive cluster techniques may be applied, which combine the computer's computational power with a human's knowledge. In Shapiro & Matheus's paper on knowledge discovery workbench [14], a tool for line clustering of points involving numerical values of two attributes is discussed, as a part of data visualization. That is an example of the kind of interaction that can take place between a human expert and a data mining tool.

The problem of determining the exact number of clusters can be analyzed using some measure of the goodness of fit which expresses how well a given set of clusters matches the data. The curse of dimensionality usually forces the analyst to choose a simple quadratic optimizing function instead of using the chi-square or Kolmogorov-Smirnov statistic as the traditional measurement criterion. A test of hypothesis is then performed to determine whether to accept or reject the initial guess (null hypothesis).

Characterization Query: A classification query emphasizes the finding of features that distinguish different classes. On the other hand, the characterization query describes common features of a class regardless of the characteristics of other classes. The former kind of description is called discriminating while the latter is called characterizing. A typical example of characterization method can be found in [22]. Han et al., in their attribute based learning framework called DBLEARN [22], utilize concept hierarchies, which constitute background knowledge, during the generalization process. A relation that represents intermediate (or final) learning results is called an intermediate (or a final) generalized relation. A special attribute, *vote*, has been added to each generalized relation to keep track of the number of tuples in the original relation that got generalized to the current tuple in the generalized relation. The extent of the generalization is determined by a human user using a threshold value, which actually controls the number of tuples in a final generalized relation. A quantitative measure, e.g., percentage of votes, is associated with a final generalized rule, which is the disjunctive normal form of a final generalized relation, and is used to visualize the result of learning process.

5 ROUGH SETS AND DATA MINING

Even though it has been more than a decade since the introduction of the rough set theory, there is still a continueing need for further development of rough functions and for extending rough set model to new applications. We believe that the investigation of the rough set methodology for data mining in relational DBMSs is a challenging research area with promise of high payoffs in many business and scientific domains. Additionally, such investigations will lead to the integration of the rough set methodology with other knowledge discovery methodologies, under the umbrella of data mining applications. In this section, we assess the current status of and trends in the data mining problem from the point of the rough set theory.

5.1 An Introduction to Rough Set Theory

Let the pair $A = (U, R)$ be an approximation space, where U is a finite set, a subset of the universe of discourse, and R is a set of equivalence classes on U. A member of R is called an elementary (or atomic) set. A definable set in A is obtained by applying a finite number of union operations on R. Let R^* be a family of subsets of R. Then, R^* generates a topological space $T_A = (U, R^*)$. We call each member of U an object. A concept of interest, X, is a subset of U. The least definable set in A containing X, $Cl_A(X)$, is called *closure set* (also known as *upper set*) of X in A. Similarly, the greatest definable set in A that is contained in X, $Int_A(X)$, is called *interior set* (also known as *lower set*) of X in A.

A concept X is *definable* in A if for some $Y \in R^*$, X is equal to the union of all the sets in Y; otherwise X is said to be a *rough* set or *non-definable*. We would like to generate a decision algorithm, denoted by $D_A(X)$, in A such that, for a given $x \in U$, it yields one of these three answers: a) x is in X, b) x is not in X, c) *unknown*. In the following, we define corresponding sets of X in A for each answer. Let $POS_A(X)$ be a set of objects each of which is considered as a member of the concept X by $D_A(X)$. Let $BND_A(X)$ be a set of objects for which $D_A(X)$ gives the answer *unknown*. Finally, let $NEG_A(X)$ be a set of objects that are not regarded as members of X by $D_A(X)$. It is easy to see that $NEG_A(X) = U - (POS_A(X) \bigcup BND_A(X))$ by definition. In other words, the decision algorithm utilizes following rules to answer if $x \in X$:

i. $x \in POS_A(X) \Longrightarrow x \in X$,

ii. $x \in BND_A(X) \Longrightarrow$ *unknown*, and

iii. $x \in NEG_A(X) \Longrightarrow x$ *is not in* X.

Note that if x is not in one of regions, then a decision may be taken on using *closeness* heuristic [59], provided that each region and object have some type of descriptions. For the sake of simplicity, the decision algorithm $D_A(X)$ is assumed to be a set of decision rules, where each rule gives positive answer.

There are two approximation methods defined in algebraic approximation spaces:

a. Lower Approximation: $POS_A^l(X) = \underline{A}(X) = Int_A(X)$, and

b. Upper Approximation: $POS_A^u(X) = \overline{A}(X) = Cl_A(X)$.

In both methods, the boundary region of the concept X is equal to $Cl_A(X) - POS_A(X)$. The degree of imprecision is expressed by the accuracy measure

$$\mu_A(X) = \frac{|Int_A(X)|}{|Cl_A(X)|}$$

The classification Problem

Let $F = \{X_1, X_2, \ldots, X_k\}$, where $X_i \subseteq U$, be a partition of U. Interior and closure sets of F in A is defined as the family

$$Int_A(F) = \{Int_A(X_1), Int_A(X_2), \cdots, Int_A(X_k)\}$$

and

$$Cl_A(F) = \{Cl_A(X_1), Cl_A(X_2), \cdots, Cl_A(X_k)\}$$

respectively.

A classification problem is described as generating a decision algorithm, $D_A(R, F)$, that relates definable sets to concepts. If $D_A(R, F)$ is a relation then it is called *an inconsistent decision algorithm;* otherwise, it is said to be *a consistent decision algorithm.* Since $POS_A(R, F) = \bigcup_{X \in F} POS_A(R, X)$, the extension of an approximation method to its counterpart in classification problem is straightforward. Similarly, the classification accuracy $\beta_A(F)$ is equal to

$$\frac{\sum_{i=1}^{k} |Int_A(X_i)|}{\sum_{i=1}^{k} |Cl_A(X_i)|}.$$

In the classification problem, it is usual to define a second measure, *quality* of the classification F in A as shown in the below:

$$\eta_A(F) = \frac{\sum_{i=1}^{k} |Int_A(X_i)|}{|U|}.$$

If $\eta_A(F) = \beta_A(F)$ the classification is said to be *definable* (or *perfect*); otherwise it is called *roughly definable* classification.

The Notion of Decision Tables

Information system (also known attribute system) can be viewed as an application of rough set theory such that each object is described by a set of attributes. It is defined as a quadruple $S = (U, Q, V, \rho)$ where: U is the finite set of objects; Q is the set of attributes; denoted and $\rho : U \times Q \Rightarrow V$ is a total description function. For all $x \in U$ and $a \in Q$, $\rho(x, a) = \rho_x(a)$. The set of attributes in S is considered as the 'union of' condition and decision attributes when classification of objects is emphasized. The condition and decision attributes are denoted by CON, and DEC respectively. In this context, the information system is called a *decision table.* For given $P \subseteq Q$, let $U/\widetilde{P}$ denote the set of natural equivalence classes on U by the values of P.

A decision algorithm, induced from S, relates the elements of $U/\widetilde{CON}$ to that of $U/\widetilde{DEC}$. Note that every approximation problem in an algebraic space can be converted to the one in a decision table.

5.2 Data Mining Issues in Rough Sets

In rough set theory, accuracy measure is used for approximation of a concept, and both accuracy and quality measures are used for a classification problem. Deogun at al. in [60] have proposed a unification of these two measures, which is the normalized size of intersection between approximated concept, X, and its positive region in an approximation space A, $POS_A(X)$, as formalized below.

$$\mu_A(X) = \frac{|X \cap POS_A(X)|}{s_1 |X| + s_2 |POS_A(X)|},$$

where s_1 and s_2 are scaling factors and their sum must be equal to one. These scaling factors quantify the user's preference as to amount of increment in accu-

racy of $D_A(X)$ desired relative to a certain loss in accuracy of X (or vice versa). Note that when $s_1 = s_2 = 0.5$, the measure $\mu_A(X)$ becomes equal to *Dice's coefficient* in information retrieval systems. Note that the unified quality measure takes into account not only positive coverage, but also negative training examples that the condition part of a decision rule may cover. It is, however, worth pointing out that these measures are used to quantify accuracy (or quality) of an induced rough classifier and none of them are used during induction part of a process, except that, as explained later in this section, elementary classifiers and rough classification methods in probabilistic approximation spaces utilize accuracy measure to select a conjunct (or an elementary set). According to a reported study [55], DNF concept learning algorithms may induce many unreliable disjuncts each of which covers a small number of positive training examples. Since rough classifiers can be viewed as a DNF concept learner, and the study to incorporate the unified quality measure into post-pruning process can be well justified.

Ultra large data

Knowledge discovery with an *ultra large data set* is a novel area for the rough set methodology. As stated earlier, one of the plausible approaches to tackle ultra large data is to reduce the data set horizontally, which is not unknown to the rough set community. For example, in KDD-R system, the data preprocessing unit discretizes the numerical attributes either by applying user-supplied discretization formula or by using an automatic discretization algorithm [61]. Alternatively, horizontal reduction of a very large data set table may use a generalization hierarchy of attributes to merge identical tuples, after the substitution of an attribute value, by its higher level concept in the generalization hierarchy. This is one of the strategies used in the *attribute oriented* approach for inductive concept learning [22]. Since an attribute-oriented learning technique operates on relations, its strategies can be easily adapted to rough classifiers to reduce the size of some categorical attributes.

Uncertainty in data

In the algebraic space, rough set theory approximates given concept(s) using lower and upper sets of the concept(s). Given that the uncertainty in a data set is caused by *noisy* or *incomplete* data, this approach is not always desirable because it does not exercise oppurtunities to discover/generalize a valuable pattern that is perturbed by noise. This problem has been the subject of numerous studies on developing rough approximation methods based on different defini-

tions of positive (and boundary) regions [60, 62, 63, 64]. For example, in the *elementary set approximation* of an unknown concept [60], an elementary set is mapped to the positive region of an unknown concept if its degree of membership is bigger than a user defined threshold value. Alternatively, another approach would be to shift the domain of the problem from algebraic space to the probabilistic space, if one can assign prior probabilistic measures to the definable sets.

In rough set based classification, inconsistent rough classifiers (or decision algorithms) have not received as much attention as consistent rough classifiers. In the rough set literature, the terms 'inconsistent' and 'nondeterministic' decision algorithms (or rules) are used interchangeably, though they are different concepts. The 'inconsistency' is attributed to the result of a classification method while the 'nondeterminism' is attributed to the interpretation of that result. As shown in [60], inconsistent decision algorithms, under an appropriate representation structure, can be interpreted deterministically as well as nondeterministically. This is an important result, particularly when the background knowledge is *incomplete* and *dynamic.*

Redundant data

Redundant data can be eliminated by pruning insignificant attributes with respect to a certain problem at hand. In the rough set terminology, the emphasis, however, is given to more restrictive version of the redundancy problem that is called reduction of an information system (also known as attribute-value system). It is the process of reducing an information system such that the set of attributes of the reduced information system is independent and no attribute can be eliminated further without losing some information from the system, the result of which is called *reduct* [62, 65]. Given the fact that exhaustive search over the attribute space is exponential in the number of attributes it might not always be computationally feasible to search for the minimum size reduct of attributes. Furthermore, finding just a single reduct of the attributes may be too restrictive for some data analysis problems, which is one of the arguments stated in Kohavi & Frasca's paper [66]. One plausible approach is to utilize the idea of *θ-reduct* as described below.

Let $S(P)$ denote a substructure of S such that $S(P) = (U, Q' = P \cup DEC, \bigcup_{a \in P} V_a, \rho')$, where $P \subseteq CON$, ρ' is a restriction of ρ to set $U \times Q'$. It is said that $CON - P$ is *θ-superfluous* in S iff

$$\varphi_{S(P)}(U/\widetilde{DEC}) = \varphi_S(U/\widetilde{DEC})(1 - \theta),$$

where $0 \leq \theta \leq 1$. Similarly, P is a *θ-reduct* of CON iff $CON - P$ is a *θ-superfluous* in S and no $P' \subset P$ is *θ-superfluous* in $S(P)$. As stated before, the feature selection problem is to choose a small subset of features that is necessary and sufficient to define the target concept(s). In terms of these new definitions, feature selection problem can be re-expressed as finding a θ-reduct of CON in S. A stepwise backward algorithm to find a θ-reduct of a given feature set was introduced by Deogun et al. in [37] on the premise that the quality of upper classifier decreases as the feature set is pruned down.

Dynamic data

The theory of rough sets is based on the premise that the universe of discourse (or the set of objects) is *finite*; that is, it considers a snapshot of a database, which may not be a valid assumption if the background knowledge is indeed *dynamic*. A plausible remedy for this problem is to design an incremental method and separate the summary and the result of a method from one to another. Ziarko, in [18], has used the strength of a decision rule as a part of the summary of the decision algorithm. Similarly, a further refinement of antecedent parts of rules in a decision algorithm is a part of the summary if the decision algorithm is *persistent* in the system and the background knowledge from which the decision algorithm has been induced is dynamic. Deogun et al. in [60] extended decision tables to represent upper classifiers such that each tuple contains a special and composed field, called *incremental information,* which contains the number of objects that satisfy condition part of a decision rule and the number of objects being classified correctly by the same decision rule. The extended decision table evolves over time, provided that the incremental information is updated correspondingly as the background knowledge, from which the upper classifier had been induced, changes.

Data mining methods

When we inspect *the data mining queries* with respect to the rough set methodology, we see that attribute dependency analysis and classification are well investigated subjects among others. The *hypothesis testing* and *association between values of an attribute* can easily be solved by the rough set methodology (see Deogun et al. [67]). A recent theoretical paper by Kent [68] extends the notions of approximation and rough equality to formal concept analysis. An immediate result of this study, in our data mining context, is to be able to use the rough set methodology for the *characterization of a concept* (or more generally for concept exploration). As a final note, for handling an interesting

subset of data mining queries by the rough set methodology, the rough classifiers face a problem when a new object (coming from outside of the data set) is introduced and the description of the object is not found in the corresponding classifier. In other words, the problem is to find the closeness of given object to known concepts at hand. The usual remedy for this problem is to map non-quantitative (nominal) values into a numerical scale and use a distance function for the evaluation. For example, Kira & Rendell suggested a binary scale and the they used it in their *Relief* algorithm for feature selection [38]. Using more domain knowledge, Slowinski & Stefanowiski in [59] have suggested a distance measure based on mapping the difference between two values of an attribute into a well-ordered scale consisting of *indifferent, weakly indifferent, strictly different, and excessively different symbols (or intervals).* For quantitative (or cardinal) attributes, a decision maker compares the absolute difference between two values with three threshold values in order to decide which interval should be assigned. In the case of nominal attributes, all pairs of values are assigned an interval in advance. Then the closeness of an object to a rule is determined over the interval $[0, 1]$ by using partial differences of attribute values.

6 KNOWLEDGE DISCOVERY SYSTEMS

A knowledge discovery system that is capable of operating on large, real-world databases, is referred to as a knowledge discovery in databases (KDD) system. Knowledge discovery in databases is changing the face of today's business world and has opened up new frontiers in the realm of science. In the business world, one of the most successful and widespread application of KDD is "Database Marketing" [69]. Marketers are collecting mountains of information about customers, looking for patterns among existing customer preferences and using that knowledge to predict future customer behavior and to craft a marketing message that targets such potential customers. Not only is database marketing opening up new avenues for reaching out to customers, but it is also helping a faceless, distant marketer to recreate a personal relationship with its customers. In the scientific domain, KDD has a wide range of applications— from mass spectroscopy, to prognosis of breast cancer recurrence and the location of primary tumor, to automatic analysis and cataloging of celestial data.

The development of a KDD system is a complex process and is influenced by many factors including the extent of user involvement in controlling the discovery process, the tasks it can support, the number and variety of tools provided to support these tasks and the kinds of output that is being generated by the

system. In this section, a number of KDD systems are described and compared in terms of the characteristics mentioned above. For ease of exposition, we classify these systems into two broad categories:

a) *generic systems*, which support either a wide range of application areas or support a variety of data mining tasks, or

b) *application-specific systems*, which have been developed with a specific application domain in mind.

It is interesting to note that the versatility of a system decreases as one goes from systems supporting many data mining tools and/or many possible applications, to systems solving a specific application problem. However, systems that support many tools place heavy reliance on the judgement of the user and hence are less autonomous than the ones that have been developed for specific applications.

The algorithms used to extract the underlying patterns in the data set form the core of any knowledge discovery system. Providing a wide variety of knowledge discovery methods may cause unnecessary increase in the number of distinct knowledge structures maintained in a knowledge base and hence careful considerations must be given to the choice of a set of knowledge structures that is orthogonal, simple and minimal.

6.1 Generic Systems

These systems are versatile in the sense that a variety of tools are embedded in the system and/or that these can support a wide spectrum of applications.

The INLEN system [70], which is partially operational, combines database, knowledge base, and a wide spectrum of machine learning techniques to assist a data analysis expert to extract new or better knowledge from the database or/and knowledge base and discover interesting regularities in the database. The representation of data in the database and of information in the knowledge base are relational tables and knowledge segments respectively. A knowledge segment (KS) can be simple or compound. Simple KSs include rule sets, equations, networks, and hierarchies. Compound KSs consist of combinations of either simple KSs or KSs and relational tables.

The INLEN system employs four sets of operators: Data Management Operators (DMOs), Knowledge Management Operators (KMOs), Knowledge Generation Operators (KGOs), and macro operators. Instead of interfacing the INLEN system to a DBMS, the designers of INLEN has chosen to equip it with DMOs that have capabilities of a typical relational query language. KMOs have analogously been implemented to manipulate knowledge segments. The KGOs take input from both the database and the knowledge base, and invoke various machine learning programs. Macro operators allow a user to encompass a sequence of INLEN operators as a single operator.

INLEN-1, the first stage of implementing the INLEN system, was built on a knowledge base of simple decision rules, a relational database, and a user-oriented and menu based graphical interface. Characterization of a class, classification of a set of examples, improving the knowledge through new examples, suggesting likely values for unknown value of an attribute, and testing the performance of a rule set on a set of examples comprise implemented subset of KGOs. The INLEN approach lacks of orthogonality principle in designing both knowledge structures and built-in KGOs. For instance, rules in disjunctive normal form, decision trees, and relational tables are typical knowledge structure in INLEN's knowledge base, though they are equivalent in terms of modeling real-world objects. Similarly, it would have been better if more complex KGOs had been implemented on the kernel consisting of primitive KGOs, rather than collecting all KGOs in one menu.

The Knowledge Discovery Workbench (KDW) [14] is a collection of tools for interactive analysis of large databases. Many of its design principles and characteristics are similar to those of INLEN. The pattern extraction algorithms range from clustering to classification to deviation detection. Dependency analysis for finding and displaying probabilistic relationships, and summarization for characterizing classes are also incorporated in KDW. All these have made KDW a versatile and domain independent system. However, owing to this reason control in KDW is provided exclusively by the user, "who must decide what data to access, how to focus the analysis, which discovery algorithm to apply and how to evaluate and interpret the result". KDW is "ideal for exploratory data analysis by a user knowledgeable in both data and operation of discovery tools". However such heavy reliance on the user has given the system a low ranking on the autonomy scale.

Explora [4, 71] is another KDD system that incorporates a variety of search strategies to adapt discovery processes to the requirements of applications. It operates by performing a graph search through a network of patterns, searching for instances of interesting patterns. Interestingness is evaluated locally by the

verification method and is in the form of filtering redundant rules, finding something that is useful to the user, or some characteristic that is unique to a fraction of the population. The pattern templates can assume three forms— rule searcher, change detector and trend detector. Explora is specifically designed to work with data that changes regularly and often. Its knowledge extraction tool is fundamentally a deviation detector that identifies significant differences between populations or across time periods. A user of Explora experiences a moderately high degree of versatility and autonomy.

One of the leading data mining toolkit of modern era, that has been subjected to diverse applications, is Clementine [72]. Clementine is built on the technologies of neural networks and rule induction and hence can automatically identify the relationships in the data and generate rules to apply to future cases. It is essentially a classifier system and includes functions which can handle a sequence of records— ideal for handling time series data. Clementine has been applied to verify incoming foreign exchange stock price data, model skin corrosivity, select locations for retail outlets, anticipating toxic health hazards, and predicting audiences for television programs for the British Broadcasting Corporation (BBC).

DataLogic/R [73] is another software designed to perform multiple tasks in data analysis, knowledge discovery and reasoning from data and is based on the concept of rough set analysis. The analysis and pattern discovery involves elimination of redundant attributes, elimination of redundant data and generation of classification rules. These rules are non-redundant and can be either probabilistic or deterministic. The system also provides a series of quality indicators for these rules, viz., strength, confidence factors, supporting cases, train and test validation, etc. By varying the roughness, DataLogic/R can discover knowledge at different levels of detail. DataLogic/R has been used successfully in the "automated discovery of control rules for NO_X and SO_2 emissions from utility boilers", and market analysis.

The system LERS (Learning from Examples based on Rough Sets) [74, 75] induces a set of rules from examples given.in the form of a decision table. The examples in the table are described by the values of attributes and are characterized by a value of a decision, as assigned by an expert. The output is a set of rules that is minimal and provides a description of the concept defined only by the examples supporting it (positive examples). Besides the machine learning rules from examples, LERS also contains options for knowledge acquisition. The sets of rules generated by these options are called All Coverings and All Rules and are usually bigger than the set of rules given by the machine learning options. Experiments have been performed to test the efficacy of LERS system

for a variety of examples that differ substantially from each other in terms of the number of examples, attributes and concepts. In terms of completeness of the rules, it is noted that All Rules provide the most complete set while the error rates of the rule sets induced by the machine learning options are the worst.

System KDD-R [61] is a software providing a collection of rough sets based tools for comprehensive data analysis. It is based on the idea of variable precision rough sets (VPRS) model and investigates the relationship between two sets of user defined attributes, condition and decision attributes, that characterize the objects in a relational table. Control in the search space is provided by the user by specifying whether the data analysis will be focused on the lower bound or on the upper bound of each value of the decision attribute. The relationship between the discretized condition attributes and the binary decision attributes can be measured in terms of dependency between the sets of attributes, or the degree of accuracy, or the core attributes with respect to the given dependency function, or all the relative reducts of condition attributes with respect to reservation of lower bound. The algorithm for computation of all relative reducts is accomplished by the use of decision matrix. Computation of rules, besides the computation of reducts, is the other most important activity carried out by KDD-R. Minimal length rules for the lower bound (or upper bound) are computed using the decision matrix technique. These rules act synergistically in the decision process— individually each rule is treated as a piece of uncertain evidence and hence worth a little in the process of decision making; however, along with similar other rules, it can provide a substantial input to the decision making process.

6.2 Application-specific Systems

Commercial systems, like CoverStory [4], Spotlight [76] and KEFIR [4], have been developed to discover knowledge in databases using the concept of deviations. Deviations are powerful because they provide a simple way of identifying interesting pattern in the data. All these systems perform an automatic drill-down through data to determine the most important deviations and then rank these deviations according to some measure of interestingness. The interestingness of a deviation is generally measured from the relevant action that can be taken in response to that deviation. The systems then generate explanations for the most interesting deviations and, where appropriate, generates simple recommendations for actions in response to such deviations. CoverStory and Spotlight have been used successfully in supermarket sales analysis and KEFIR

has provided the healthcare analysts with an early warning system. The systems are fully automated once the initial domain knowledge has been set up. However, limited applicability has forced them a low ranking on the versatility scale.

R-MINI [4] is yet another system that primarily utilizes classification techniques and also deviation detection, to some extent, to extract useful information from noisy domains, such as financial markets. It utilizes a logic minimization technique to generate a minimum-sized rule set that is complete and consistent with all the examples in the database. Complete means that the rules cover all the examples in the database, while consistent means that the rules do not misclassify any examples.

R-MINI starts by making every example into a rule. Minimality of the rule set is then achieved by iterating over the following two steps:

1. Generalization Step— For each rule, find some way to expand it without allowing it to cover any counter-examples, shrink other rules to the smallest size that will not leave out any examples and delete any other rules that do not contain any examples (empty).
2. Specialization Step— Take each rule and replace it with a rule that is not larger and that will not cause any examples to become uncovered. Delete any empty rules.

The exact dimensions along which expansion and reduction will take place is randomized at each step. Since an iteration cannot increase the number of rules, an arbitrary number of iterations with random expansion and reduction methods at each step will result in monotonically non-increasing number of classification rules that are consistent and complete at every stage of their minimization.

The feasibility of the system is determined using the S&P 500 data for a contiguous period of 78 months. The data spans 774 securities and comprised of 40 variables for each month for each security. Only one of these 40 variables is categorical and the rest are numerical. The decision variable is the difference between the return of a given portfolio and the S&P average return for the same period. This was discretized into "strongly performing" (6% above average or more), "moderately performing" (2%—6% above average), "neutral" (2% below to 2% above average), "moderately underperforming" (2% below to 6% below average), and "strongly underperforming" (6% below average or

more). The data is arranged in temporal sequence and the classification rules are generated from consecutive 12 months of data. The performance of these rules is then tested on the following sets of 12-month stream. This gives an idea of the temporal rate of decline of the predictive power of the classification rules. Once this rate is known, rules can be regenerated "every n years from the immediate past data so as to continue holding up the predictive performance".

Knowledge Discovery techniques using associative rules has been explored in TASA (Telecommunication Network Alarm Sequence Analyzer) [77]. It is an experimental knowledge discovery system developed for predicting faults in a telecommunication network. An alarm in a telecommunication network occurs whenever a part of the system behaves abnormally. A network typically generates 200–1000 alarms per day. The TASA system seeks rules of the following form: "if a certain combination of alarms occur within a certain time period, then an alarm of another type will occur within a time period". The time periods are selected by the user and the rules being sought describe a temporal relationship between alarms. Once the rules are derived, the user can select a subset of rules to display or remove from display, specify an ordering of the rules or specify a grouping or clustering of the rules.

In the scientific domain SKICAT [4] has been developed for automating the reduction and analysis of large astronomical data. The SKICAT system employs a supervised classification technique and is intended to automatically catalog and analyze celestial objects, given digitized sky images (plates). The initial step is to identify, measure and catalog the detected objects in the image into their respective classes. Initial feature extraction is carried out by an image processing software known as FOCAS. Once these features are extracted, it is necessary to derive additional features that exhibit sufficient invariance within and across plates so that classifiers trained to make accurate predictions on one plate will perform equally well on others.

One of the motivations for developing SKICAT is the need for classifying objects too faint for astronomers to recognize by sight. In order that SKICAT might classify objects that astronomers cannot, a set of faint objects is selected from plates. A second telescope, with higher resolution power and a higher signal-to-noise ratio is used to classify the faint objects and rules are generated on the classified set of faint objects from the lower resolution image. These rules can then be applied to other faint objects for which no high resolution images are available.

Classification is done by repeatedly dividing the data set randomly into training and test sets. A decision tree is generated from each training set and its rules

are tested on the corresponding test set. "By gathering a large number of rules through iterating on a randomly subsampled training parts," a large collection of robust rules is derived. These rules collectively cover the entire original data set of examples. A greedy covering algorithm is then employed to select a minimum subset of rules that covers the examples.

When subjected to data consisting of objects from different plates, the SKICAT system gave a classification accuracy of 94.2% and was superior to the results obtained from existing decision tree algorithms (ID3, GID3, O-Btree). The accuracy dropped noticeably for all methods when the derived attributes are left out.

7 FUTURE RESEARCH DIRECTIONS

In this chapter we have surveyed the state of the art in data mining, including research trends related to rough set theory. Since a major focus of this book is data mining as related to rough set theory, in this section we present future research directions in data mining as related to rough set theory. We strongly believe that rough set based approaches to data mining present an excellent and fertile area for research. As mentioned in the Section 5, some aspects of the nature of data (i.e., incomplete, redundant, and uncertain data) have already been investigated in the rough set methodology, but they need to be tested in large databases. Towards this direction, there have already been some reported works on using the rough set methodology based knowledge discovery tools on off-line data; KDD-R, an experimental open tool box [61]; LERS, a machine learning system from examples [74]; and DataLogic/R [73], a commercial product for data mining and decision support. In the following, we present future research directions that are critical for data mining applications.

- **Incremental rough approximation:** This is a must feature that has to be provided for if the decision algorithm is to be persistent in the rough set model and the background knowledge is dynamic. One of the claims made by Deogun et al. in [60] is that evolving rough classifier schemes can be developed, if the decision table is accommodated with a composite *increment* field that contains frequencies of rows.

- **Closeness of two rules:** Slowinski & Stefonowski's study on determining the nearest rule, in the case that the description of a given object does not match to those of known concepts, is a key contribution in enhancing the

performance of a rough classifier when the data set is poorly designed or sampled from a large data. Even though it is not stated in the paper, such a measure can make the rough set methodology usable for *clustering queries*. This is a very important subject that needs to be studied by the rough set community.

- **Null values:** As stated before, a null value of an attribute is more general than unknown value of that attribute, and the reasoning about null values remains an open problem in the studies of data mining. A less restrictive version of the problem, which is known as *unknown attribute values*, has been studied by Grzymala-Busse and implemented in the LERS, a machine learning system [74].

- **Characterization query:** Even though data dependency analysis within the rough set methodology can be applied to characterize concepts, it lacks of an explicit context dimension that is very important notion when a knowledge model contains a set/hierarchy of persistent concepts. For example, characterization of the concept 'Windows' within the context of 'product' is certainly different from that of the one within the context of 'sale'. This subject has been formally studied by Wille [78] and used for concept modeling. We believe that this study can be further extended to capture approximate characterization of concepts.

In summary, data mining is a practical problem that drives theoretical studies toward understanding and reasoning about *large and existing* data. Matheus et al. used the tradeoff between 'versatility' and 'autonomy' for evaluating a KDD system [3]. They have argued that an ideal KDD system would handle knowledge discovery tasks autonomously while being applicable across many domains. While progress is being made in the direction of automatically acquiring knowledge needed for for guiding and controlling the knowledge discovery process, the ideal system remains far from reach. At the system level, more research is needed in how to derive domain knowledge from databases and how to represent domain knowledge and derived knowledge in a uniform manner. At the level of methods for extracting patterns, we believe that data mining is an important application area where the theoretical results of rough set theory can be tested, in order to help us understand its strengths and weaknesses.

REFERENCES

[1] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, "Knowledge discovery

databases: An overview," in *Knowledge Discovery in Databases* (G. Piatetsky-Shapiro and W. J. Frawley, eds.), pp. 1–27, Cambridge, MA: AAAI/MIT, 1991.

[2] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, and B. Swami, "An interval classifier for database mining applications," in *Proceedings of the 18th VLDB Conference*, (Vancouver, British Columbia, Canada), pp. 560–573, 1992.

[3] C. J. Matheus, P. K. Chan, and G. Piatetsky-Shapiro, "Systems for knowledge discovery in databases," *IEEE Trans. on Knowledge and Data Engineering*, vol. 5, no. 6, pp. 903–912, 1993.

[4] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*. Cambridge, MA: MIT Press, 1996.

[5] R. Krishnamurty and T. Imielinski, "Research directions in knowledge discovery," *SIGMOD RECORD*, vol. 20, pp. 76–78, 1991.

[6] A. Silberschatz, M. Stonebraker, and J. Ullman, "Database systems: achievements and opportunities," Tech. Rep. TR-90-22, University of Texas at Austin, Department of Computer Science, 1990.

[7] K. C. C. Chan and A. K. C. Wong, "A statistical technique for extracting classificatory knowledge from databases," in *Knowledge Discovery in Databases* (G. Piatetsky-Shapiro and W. J. Frawley, eds.), pp. 107–123, Cambridge, MA: AAAI/MIT, 1991.

[8] V. V. Raghavan, H. Sever, and J. S. Deogun, "A system architecture for database mining applications," in *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, (Banff, Alberta, Canada), pp. 73–77, 1993.

[9] S. K. Lee, "An extended relational database model for uncertain and imprecise information," in *Proceedings of the 18th VLDB conference*, (Vancouver, British Columbia, Canada), pp. 211–218, 1992.

[10] B. P. Buckles and F. E. Petry, "A fuzzy model for relational databases," *Journal of Fuzzy Sets and Systems*, vol. 7, no. 3, pp. 213–226, 1982.

[11] D. Barbara, H. Garcia-Molina, and D. Porter, "The management of probabilistic data," *IEEE Trans. on Knowledge and Data Engineering*, vol. 4, no. 5, pp. 487–502, 1992.

[12] C. Corinna, H. Drucker, D. Hoover, and V. Vapnik, "Capacity and complexity control in predicting the spread between barrowing and lending interest rates," in *The First International Conference on Knowledge Discovery and Data Mining* (U. Fayyad and R. Uthurusamy, eds.), (Montreal, Quebec, Canada), pp. 51–76, aug 1995.

[13] N. Zhong and S. Ohsuga, "Discovering concept clusters by decomposing databases," *Data & Knowledge Engineering*, vol. 12, pp. 223–244, 1994.

[14] G. Piatetsky-Shapiro and C. J. Matheus, "Knowledge discovery workbench for exploring business databases," *International Journal of Intelligent Systems*, vol. 7, pp. 675–686, 1992.

[15] U. M. Fayyad and K. B. Irani, "Multi interval discretization of continuous attributes for classification learning," in *Proceedings of 13th International Joint Conference on Artificial Intelligence* (R. Bajcsy, ed.), pp. 1022–1027, Morgan Kauffmann, 1993.

[16] J. F. Elder-IV and D. Pregibon, "A statistical perspective on KDD," in *The First International Conference on Knowledge Discovery and Data Mining* (U. Fayyad and R. Uthurusamy, eds.), (Montreal, Quebec, Canada), pp. 87–93, aug 1995.

[17] S. K. M. Wong, W. Ziarko, and R. L. Ye, "Comparison of rough set and statistical methods in inductive learning," *International Journal of Man-Machine Studies*, vol. 24, pp. 53–72, 1986.

[18] W. Ziarko, "The discovery, analysis, and representation of data dependencies in databases," in *Knowledge Discovery in Databases* (G. Piatetsky-Shapiro and W. J. Frawley, eds.), Cambridge, MA: AAAI/MIT, 1991.

[19] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[20] M. James, *Classification Algorithms.* John Wiley & Sons, 1985.

[21] T. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, pp. 203–226, 1982.

[22] J. Han, Y. Cai, and N. Cercone, "Knowledge discovery in databases: An attribute-oriented approach," in *Proceedings of the 18th VLDB Conference*, (Vancouver, British Columbia, Canada), pp. 547–559, 1992.

[23] J. Ching, A. Wong, and K. Chan, "Class-dependent discretization for inductive learning from continuous and mixed mode data," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 7, pp. 641–651, 1995.

[24] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA: Morgan Kaufmann Publishers, 1988.

[25] D. Stashuk and R. Naphan, "Probabilistic inference based classification applied to myoelectric signal decomposition," *IEEE Trans. on Biomedical Engineering*, June 1992.

[26] J. Quinlan and R. Rivest, "Inferring decision trees using the minumum description length principle," *Information and Computation*, vol. 80, pp. 227–248, 1989.

[27] J. R. Quinlan, "The effect of noise on concept learning," in *Machine Learning: An Artificial Intelligence Approach* (R. Michalski, J. Carbonell, and T. Mitchell, eds.), vol. 2, pp. 149–166, San Mateo, CA: Morgan Kauffmann Inc., 1986.

[28] T. Luba and R. Lasocki, "On unknown attribute values in functional dependencies," in *Proceedings of the International Workshop on Rough Sets and Soft Computing*, (San Jose, CA), pp. 490–497, 1994.

[29] J. W. Grzymala-Busse, "On the unknown attribute values in learning from examples," in *Proceedings of Methodologies for Intelligent Systems* (Z. W. Ras and M. Zemankowa, eds.), Lecture Notes in AI, 542, pp. 368–377, New York: Springer-Verlag, 1991.

[30] B. Thiesson, "Accelerated quantification of bayesian networks with incomplete data," in *The First International Conference on Knowledge Discovery and Data Mining* (U. Fayyad and R. Uthurusamy, eds.), (Montreal, Quebec, Canada), pp. 306–311, aug 1995.

[31] J. R. Quinlan, "Unknown attribute values in induction," in *Proceedings of the Sixth International Machine Learning Workshop* (A. M. Segre, ed.), (San Mateo, CA), pp. 164–168, Morgan Kaufmann Pub., 1989.

[32] S. K. M. Wong and W. Ziarko, "Comparison of the probabilistic approximate classification and fuzzy set model," *Fuzzy Sets and Systems*, no. 21, pp. 357–362, 1982.

[33] Y. Y. Yao and K. M. Wong, "A decision theoretic framework for approximating concepts," *International Journal Man-Machine Studies*, vol. 37, pp. 793–809, 1992.

[34] J. Mingers, "An empirical comparison of selection measures for decision tree induction," *Machine Learning*, vol. 3, pp. 319–342, 1989.

[35] M. Modrzejewski, "Feature selection using rough sets theory," in *Machine Learning: Proceedings of ECML-93* (P. B. Brazdil, ed.), pp. 213–226, Springer-Verlag, 1993.

[36] R. Uthurusamy, U. Fayyad, and S. Spangler, "Learning useful rules from inconclusive data," in *Knowledge Discovery in Databases* (G. Piatetsky-Shapiro and W. J. Frawley, eds.), Cambridge, MA: AAAI/MIT, 1991.

[37] J. S. Deogun, V. V. Raghavan, and H. Sever, "Exploiting upper approximations in the rough set methodology," in *The First International Conference on Knowledge Discovery and Data Mining* (U. Fayyad and R. Uthurusamy, eds.), (Montreal, Quebec, Canada), pp. 69–74, aug 1995.

[38] K. Kira and L. Rendell, "The feature selection problem: Tradational methods and a new algorithm," in *Proceedings of AAAI-92*, pp. 129–134, AAAI Press, 1992.

[39] H. Almuallim and T. Dietterich, "Learning with many irrelevant features," in *Proceedings of AAAI-91*, (Menlo Park, CA), pp. 547–552, AAAI Press, 1991.

[40] Z. Pawlak, K. Slowinski, and R. Slowinski, "Rough classification of patients after highly selective vagotomy for duodenal ulcer," *International Journal of Man-Machine Studies*, vol. 24, pp. 413–433, 1986.

[41] C. Y. Chang, "Dynamic programming as applied to feature subset selection in a pattern recognition system," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 166–171, 1973.

[42] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. on Computers*, vol. c-26, no. 9, pp. 917–922, 1977.

[43] R. A. Devijver and J. Kittler, *Pattern Recognation: A statistical approach.* London: Prentice Hall, 1982.

[44] A. J. Miller, *Subset Selection in Regression.* Chapman and Hall, 1990.

[45] U. M. Fayyad and K. B. Irani, "The attribute selection problem in decision tree generation," in *Proceedings of AAAI-92*, pp. 104–110, AAAI Press, 1992.

[46] P. Baim, "A method for attribute selection in inductive learning systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 888–896, 1988.

[47] P. J. Huber, "Projection pursuit," *Annals of Statistics*, vol. 13, no. 2, pp. 435–475, 1985.

[48] R. Agrawal, T. Imielinski, and A. Swami, "Database mining: A performance perspective," *IEEE Trans. Knowledge and Data Eng.*, vol. 5, no. 6, pp. 914–924, 1993.

[49] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[50] S. Salzberg, *Learning with Nested Generalized Exemplars.* Boston, MA: Kluwer Academic Publishers, 1990.

[51] S. M. Weiss and C. A. Kulikowski, *Computer Systems that Learn.* San Mateo, CA: Morgan Kaufmann, 1991.

[52] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach.* Palo Alto, CA: Tioga, 1983.

[53] J. Shavlik and T. Diettrich, *Readings in Machine Learning.* San Mateo, CA: Morgan Kaufmann, 1990.

[54] S. Muggleton, A. Srinivasan, and M. Bain, "Compression, significance and accuracy," in *Proceedings of 9th International Workshop on Machine Learning, (ML92)*, (Aberdeen, Scotland), Morgan Kauffmann, 1992.

[55] R. Holte, L. Acker, and B. Porter, "Concept learning and the problem of small disjuncts," in *Proceedings of 11th International Joint Conference on Artificial Intelligence*, (Detroit, MI), Morgan Kauffmann, 1989.

[56] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap.* Chapman & Hall, 1993.

[57] K. Fukunaga and R. Hayes, "Effects of sample size in classifier design," *IEEE Trans. on Pattern analysis and Machine Intelligence*, vol. 11, no. 8, pp. 873–885, 1985.

[58] M. P. D. Fisher and P. Langley, *Concept Formation, Knowledge and Experience in Unsupervised Learning.* San Mateo, CA: Morgan Kaufmann, 1991.

[59] R. Slowinski and J. Stefanowiski, "Rough classification with valued closeness relation," in *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, (San Jose, CA), 1995.

[60] J. S. Deogun, V. V. Raghavan, and H. Sever, "Rough set based classification methods and extended decision tables," in *Proceedings of the International Workshop on Rough Sets and Soft Computing*, (San Jose, California), pp. 302–309, 1994.

[61] W. Ziarko and N. Shan, "KDD-R: a comprehensive system for knowledge discovery in databases using rough sets," in *Proceedings of the International Workshop on Rough Sets and Soft Computing*, (San Jose, California), pp. 164–173, 1994.

[62] J. D. Katzberg and W. Ziarko, "Variable precision rough sets with asymmetric bounds," in *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, (Banff, Alberta, Canada), pp. 163–190, 1993.

[63] Y. Y. Yao and X. Li, "Uncertainty reasoning with interval-set algebra," in *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, (Banff, Alberta, Canada), pp. 191–201, 1993.

[64] R. R. Hashemi, B. A. Pearce, W. G. Hinson, M. G. Paule, and J. F. Young, "IQ estimation of monkeys based on human data using rough sets," in *Proceedings of the International Workshop on Rough Sets and Soft Computing*, (San Jose, California), pp. 400–407, 1994.

[65] Z. Pawlak, "Rough classification," *International Journal of Man-Machine Studies*, vol. 20, pp. 469–483, 1984.

[66] R. Kohavi and B. Frasca, "Useful feature subsets and rough set reducts," in *Proceedings of the International Workshop on Rough Sets and Soft Computing*, (San Jose, California), pp. 310–317, 1994.

[67] J. S. Deogun, V. V. Raghavan, and H. Sever, "Rough set model for database mining applications," Tech. Rep. TR-94-6-10, The University of Southwestern Louisiana, The Center for Advanced Computer Studies, 1994.

[68] R. E. Kent, "Rough concept analysis," in *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, (Banff, Alberta, Canada), pp. 245–253, 1993.

[69] J. Berry, "Database marketing," *Business Week*, pp. 56–62, September 5 1994.

[70] K. A. Kaufmann, R. S. Michalski, and L. Kerschberg, "Mining for knowledge in databases: Goals and general description of the INLEN system," in *Knowledge Discovery in Databases* (W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, eds.), Cambridge, MA: MIT Press, 1991.

[71] P. Hoschka and W. Klosgen, "A support system for interpreting statistical data," in *Knowledge Discovery in Databases* (G. Piatetsky-Shapiro and W. J. Frawley, eds.), pp. 325–345, Cambridge, MA: AAAI/MIT, 1991.

[72] Integrated Solutions, Ltd., Hampshire, England, *Clementine - Software for Data Mining.*

[73] A. J. Szladow, "DataLogic/R: for database mining and decision support," in *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, (Banff, Alberta, Canada), p. 511, 1993.

[74] J. W. Grzymala-Busse, "The rule induction system LERS Q: a version for personal computers," in *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, (Banff, Alberta, Canada), p. 509, 1993.

[75] D. M. Grzymala-Busse and J. W. Grzymala-Busse, "Comparison of machine learning and knowledge acquisition methods of rule induction based on rough sets," in *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, (Banff, Alberta, Canada), pp. 297–306, 1993.

[76] T. Anand and G. Kahn, "Spotlight: A data explanation system," in *Proceedings of the Eighth IEEE Conference on Applied AI*, (Washington, D.C.), pp. 2–8, IEEE Press, 1992.

[77] K. Hatonen, M. Klemettinen, H. Mannila, and P. Ronkinen, "Knowledge discovery from telecommunications network alarm databases," in *Proceedings of the 12th International Conference on Data Engineering* (C. Bogdan, ed.), (New Orleans, LA), feb/mar 1996.

[78] R. Wille, "Restructuring lattice theory: An approach based on hierarchies on concepts," in *Ordered Sets* (I. Rival, ed.), Dordrecht-Boston: Reidel, 1982.

3

A REVIEW OF ROUGH SET MODELS

Y.Y. Yao*, S.K.M. Wong, and T.Y. Lin*****

** Department of Computer Science, Lakehead University*
Thunder Bay, Ontario, Canada P7B 5E1
*** Department of Computer Science, University of Regina*
Regina, Saskatchewan, Canada S4S 0A2
**** Department of Mathematics and Computer Science*
San Jose State University, San Jose, California 95192

ABSTRACT

Since introduction of the theory of rough set in early eighties, considerable work has been done on the development and application of this new theory. The paper provides a review of the Pawlak rough set model and its extensions, with emphasis on the formulation, characterization, and interpretation of various rough set models.

1 INTRODUCTION

In early eighties, Pawlak [22] introduced the theory of rough sets as an extension of set theory for the study of intelligent systems characterized by insufficient and incomplete information [22, 23, 26]. It is motivated by the practical needs in classification and concept formation [27]. One may regard the theory of rough sets to be complementary to other generalizations of set theory, such as fuzzy sets and multisets [6, 24, 27, 42]. In recent years, there has been a fast growing interest in this new emerging theory. The successful applications of the rough set model in a variety of problems have amply demonstrated its usefulness and versatility [13, 15, 25, 33, 50].

The main objective of this paper is to present a review of the standard rough set model and its extensions, and to give some new results. Our emphasis will be on the formulation, characterization, and interpretation of various rough set models. We group existing rough set models into two major classes, the algebraic and probabilistic rough set models, depending on whether statistical information is used. In the algebraic class, we examine different rough set mod-

els in relation to modal logic, graded rough set models, rough set models over two universes, and rough set models over Boolean algebras. In the probabilistic class, we analyze rough membership functions and variable precision rough set models. More importantly, the probabilistic rough set models are justified based on the framework of decision theory.

In this paper, binary relations are used as a primitive notion. Rough set models are built and investigated based on various binary relations. Our aim is not to provide a complete and exhaustive summary of all works on rough set models. We only review existing works that fall in the framework we intent to establish based on binary relations. Many important studies, such as the construction of rough set model based on a covering of the universe [48] and algebraic study of rough set models [30, 37], are not covered in this paper.

2 ALGEBRAIC ROUGH SET MODELS

This section reviews the Pawlak rough set model and presents its extensions and interpretations.

2.1 Pawlak rough set model

Let U denote a finite and non-empty set called the universe, and let $\Re \subseteq U \times U$ denote an equivalence relation on U. The pair $apr = (U, \Re)$ is called an approximation space. The equivalence relation $\Re$ partitions the set U into disjoint subsets. Such a partition of the universe is denoted by $U/\Re$. If two elements x, y in U belong to the same equivalence class, we say that x and y are indistinguishable. The equivalence classes of $\Re$ and the empty set $\emptyset$ are called the elementary or atomic sets in the approximation space $apr = (U, \Re)$. The union of one or more elementary sets is called a composed set. The family of all composed sets, including the empty set, is denoted by $\mathrm{Com}(apr)$, which forms a Boolean algebra.

The equivalence relation and the induced equivalence classes may be regarded as the available information or knowledge about the objects under consideration. Given an arbitrary set $X \subseteq U$, it may be impossible to describe X precisely using the equivalence classes of $\Re$. That is, the available information is not sufficient to give a precise representation of X. In this case, one may

characterize X by a pair of lower and upper approximations:

$$\begin{aligned} \underline{apr}(X) &= \bigcup_{[x]_{\Re} \subseteq X} [x]_{\Re}, \\ \overline{apr}(X) &= \bigcup_{[x]_{\Re} \cap X \neq \emptyset} [x]_{\Re}, \end{aligned} \tag{1.1}$$

where

$$[x]_{\Re} = \{y \mid x \Re y\}, \tag{1.2}$$

is the equivalence class containing x. The lower approximation $\underline{apr}(X)$ is the union of all the elementary sets which are subsets of X. It is the largest composed set contained in X. The upper approximation $\overline{apr}(X)$ is the union of all the elementary sets which have a non-empty intersection with X. It is the smallest composed set containing X. An element in the lower approximation necessarily belongs to X, while an element in the upper approximation possibly belongs to X. We can also express lower and upper approximations as follow:

$$\begin{aligned} \underline{apr}(X) &= \{x \mid [x]_{\Re} \subseteq X\} \\ \overline{apr}(X) &= \{x \mid [x]_{\Re} \cap X \neq \emptyset\}. \end{aligned} \tag{1.3}$$

That is, an element of U necessarily belongs to X if all its equivalent elements belong to X; it is possibly belongs to X if at least one of its equivalent elements belongs to X.

For any subsets $X, Y \subseteq U$, the lower approximation $\underline{apr}$ satisfies properties:

(AL1) $\underline{apr}(X) = \sim\overline{apr}(\sim X)$,
(AL2) $\underline{apr}(U) = U$,
(AL3) $\underline{apr}(X \cap Y) = \underline{apr}(X) \cap \underline{apr}(Y)$,
(AL4) $\underline{apr}(X \cup Y) \supseteq \underline{apr}(X) \cup \underline{apr}(Y)$,
(AL5) $X \subseteq Y \Longrightarrow \underline{apr}(X) \subseteq \underline{apr}(Y)$,
(AL6) $\underline{apr}(\emptyset) = \emptyset$,
(AL7) $\underline{apr}(X) \subseteq X$,
(AL8) $X \subseteq \underline{apr}(\overline{apr}(X))$,
(AL9) $\underline{apr}(X) \subseteq \underline{apr}(\underline{apr}(X))$,
(AL10) $\overline{apr}(X) \subseteq \underline{apr}(\overline{apr}(X))$,

and the upper approximation $\overline{apr}$ satisfies properties:

(AU1) $\overline{apr}(X) = \sim\underline{apr}(\sim X)$,

$$
\begin{aligned}
&\text{(AU2)} && \overline{apr}(\emptyset) = \emptyset, \\
&\text{(AU3)} && \overline{apr}(X \cup Y) = \overline{apr}(X) \cup \overline{apr}(Y), \\
&\text{(AU4)} && \overline{apr}(X \cap Y) \subseteq \overline{apr}(X) \cap \overline{apr}(Y), \\
&\text{(AU5)} && X \subseteq Y \Longrightarrow \overline{apr}(X) \subseteq \overline{apr}(Y), \\
&\text{(AU6)} && \overline{apr}(U) = U, \\
&\text{(AU7)} && X \subseteq \overline{apr}(X), \\
&\text{(AU8)} && \overline{apr}(\underline{apr}(X)) \subseteq X, \\
&\text{(AU9)} && \overline{apr}(\overline{apr}(X)) \subseteq \overline{apr}(X), \\
&\text{(AU10)} && \overline{apr}(\underline{apr}(X)) \subseteq \underline{apr}(X),
\end{aligned}
$$

where $\sim X = U - X$ denotes the set complement of X. The lower and upper approximations may be viewed as two operators on the universe [14]. Properties (AL1) and (AU1) state that two approximation operators are dual operators. Hence, properties with the same number may be regarded as dual properties. These properties are not independent.

Based on the lower and upper approximations of a set $X \subseteq U$, the universe U can be divided into three disjoint regions, the positive region $\text{POS}(X)$, the negative region $\text{NEG}(X)$, and the boundary region $\text{BND}(X)$:

$$
\begin{aligned}
\text{POS}(X) &= \underline{apr}(X), \\
\text{NEG}(X) &= U - \overline{apr}(X), \\
\text{BND}(X) &= \overline{apr}(X) - \underline{apr}(X).
\end{aligned} \tag{1.4}
$$

Figure 1 illustrates the approximation of a set X, and the positive, negative and boundary regions. Each small rectangle represent an equivalence class. From this figure, we have the following observations. One can say with certainty that any element $x \in \text{POS}(X)$ belongs to X, and that any element $x \in \text{NEG}(X)$ does not belong to X. The upper approximation of a set X is the union of the positive and boundary regions, namely, $\overline{apr}(X) = \text{POS}(X) \cup \text{BND}(X)$. One cannot decide with certainty whether or not an element $x \in \text{BND}(X)$ belongs to X. For arbitrary element $x \in \overline{apr}(X)$, one can only conclude that x possibly belongs to X.

An important concept related to lower and upper approximations is the accuracy of the approximation of a set [22]. Yao and Lin [44] have shown that the accuracy of approximation can be interpreted using the well-known Marczewski-Steinhaus metric, or MZ metric for short. For two sets X and Y,

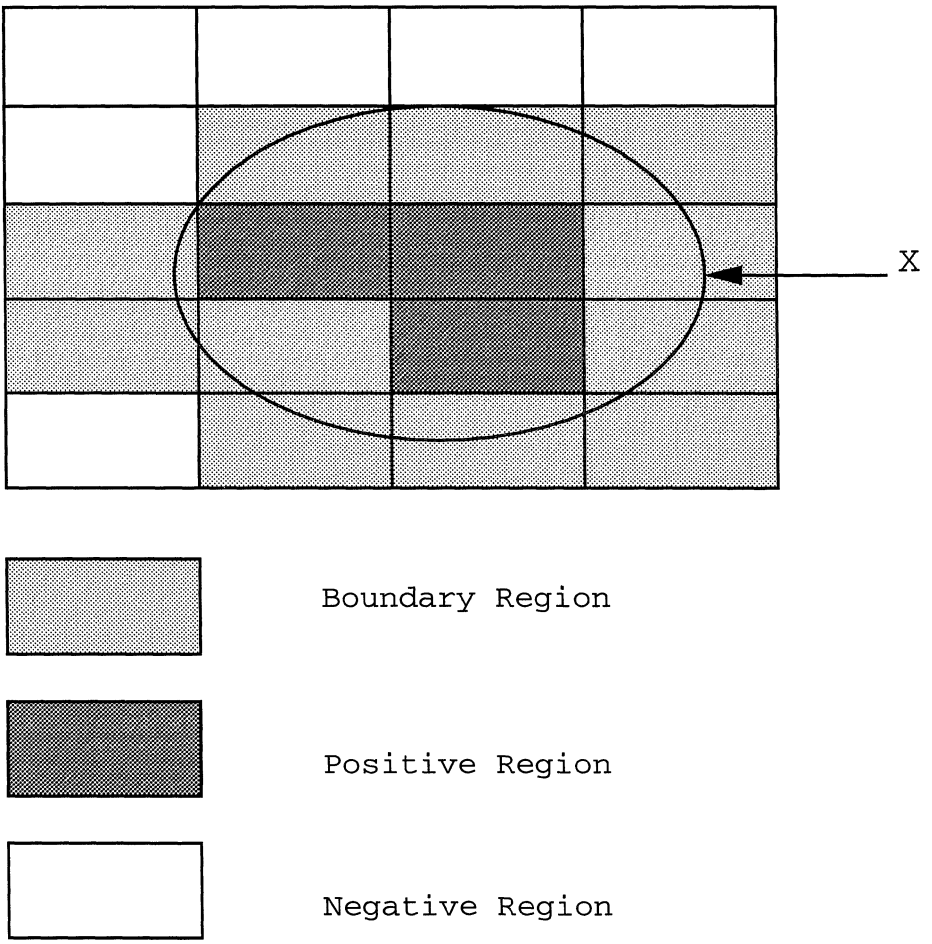


Figure 1 Positive, boundary and negative regions of a set X

where $X \Delta Y = (X \cup Y) - (X \cap Y)$ denotes the symmetric difference between two sets X and Y, and $|\cdot|$ the cardinality of a set. It reaches the maximum value of 1 if X and Y are disjoint, i.e., they are totally different, and it reaches the minimum value of 0 if X and Y are exactly the same. The quantity,

$$S(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}, \tag{1.6}$$

may be interpreted as a measure of similarity or closeness between X and Y. By applying the MZ metric to the lower and upper approximations, we have:

$$\begin{aligned} D(\underline{apr}(X), \overline{apr}(X)) &= 1 - \frac{|\underline{apr}(X) \cap \overline{apr}(X)|}{|\underline{apr}(X) \cup \overline{apr}(X)|} \\ &= 1 - \frac{|\underline{apr}(X)|}{|\overline{apr}(X)|}, \end{aligned} \tag{1.7}$$

The distance function defined above is indeed the *inverse* function of the accuracy of rough set approximation proposed by Pawlak [22], namely,

$$\begin{aligned} \rho(X) &= 1 - D(\underline{apr}(X), \overline{apr}(X)) \\ &= \frac{|\underline{apr}(X)|}{|\overline{apr}(X)|} \\ &= S(\underline{apr}(X), \overline{apr}(X)). \end{aligned} \tag{1.8}$$

For the empty set $\emptyset$, we define $\rho(\emptyset) = 1$. If X is a composed set, then $\rho(X) = 1$. If X is not composed set, then $0 \leq \rho(X) < 1$.

In the Pawlak rough set model, an arbitrary set is described by a pair of lower and upper approximations. Several different interpretations of the concepts of rough sets have been proposed. The interpretation suggested by Iwinski [11] views a rough set as a pair of composed sets, and the original proposal of Pawlak regards a rough set as a family of sets having the same lower and/or upper approximation. Rough sets may also be described by using the notion of rough membership functions, which will be discussed in Section 3.

Given two composed sets $X_1, X_2 \in \mathrm{Com}(apr)$ with $X_1 \subseteq X_2$, Iwinski called the pair (X_1, X_2) an rough set [11]. In order to distinguish it from other definition, we call the pair an I-rough set. Let $R(apr)$ be the set of all I-rough sets. Set-theoretic operators on $R(apr)$ can be defined component-wise using standard set operators. For a pair of I-rough sets, we have:

$$\begin{aligned} (X_1, X_2) \cap (Y_1, Y_2) &= (X_1 \cap Y_1, X_2 \cap Y_2), \\ (X_1, X_2) \cup (Y_1, Y_2) &= (X_1 \cup Y_1, X_2 \cup Y_2). \end{aligned} \tag{1.9}$$

The intersection and union of two composed sets are still composed sets. The above operators are well defined, as the results are also I-rough sets. The system $(R(apr), \cap, \cup)$ is complete distributive lattice [11], with zero element $(\emptyset, \emptyset)$ and unit element (U, U). The associated order relation can be interpreted as I-rough set inclusion, which is defined by:

$$(X_1, X_2) \subseteq (Y_1, Y_2) \iff X_1 \subseteq Y_1 \text{ and } X_2 \subseteq Y_2. \tag{1.10}$$

The difference of I-rough sets can be defined as

$$(X_1, X_2) - (Y_1, Y_2) = (X_1 - Y_2, X_2 - Y_1), \tag{1.11}$$

which is an I-rough set. Finally, the I-rough set complement is given as:

$$\sim (X_1, X_2) = (U, U) - (X_1, X_2) = (\sim X_2, \sim X_1). \tag{1.12}$$

The complement is neither a Boolean complement nor a pseudocomplement in the lattice $(R(apr), \cap, \cup)$. The system $(R(apr), \cap, \cup, \sim, (\emptyset, \emptyset), (U, U))$ is called an I-rough set algebra.

In Pawlak's seminal paper, another interpretation of rough sets was introduced. Using lower and upper approximations, we define three binary relations on subsets of U:

$$\begin{aligned} X \approx_* Y &\iff \underline{apr}(X) = \underline{apr}(Y), \\ X \approx^* Y &\iff \overline{apr}(X) = \overline{apr}(Y), \\ X \approx Y &\iff \underline{apr}(X) = \underline{apr}(Y) \text{ and } \overline{apr}(X) = \overline{apr}(Y). \end{aligned} \tag{1.13}$$

Each of them defines an equivalence relation on 2^U, which induces a partition of 2^U. By interpreting an equivalence, say $[X]_\approx$ containing X, as a rough set, called a P-rough set, we obtain three algebras of rough sets.

Consider the equivalence relation $\approx$. The set of all P-rough sets is denoted by $R_\approx(apr) = 2^U/\approx$. Given two sets $X_1, X_2 \in \text{Com}(apr)$ with $X_1 \subseteq X_2$, if there exists at least a subset $X \subseteq U$ such that $\underline{apr}(X) = X_1$ and $\overline{apr}(X) = X_2$, the following family of subsets of U,

$$\langle X_1, X_2 \rangle = \{X \in 2^U \mid \underline{apr}(X) = X_1, \overline{apr}(X) = X_2\}, \tag{1.14}$$

is called a P-rough set. A set $X \in \langle X_1, X_2 \rangle$ is said to be a member of the P-rough set. Given a member X, a P-rough set can also be more conveniently expression as $[X]_\approx$, which is the equivalent class containing X. A member is also referred to as a generator of the P-rough set [3]. Rough set intersection

$\sqcap$, union $\sqcup$, and complement $\neg$ are defined by set operators as follows: for two P-rough sets $\langle X_1, X_2\rangle$ and $\langle Y_1, Y_2\rangle$,

$$\begin{aligned}
&\langle X_1, X_2\rangle \sqcap \langle Y_1, Y_2\rangle \\
&\quad = \{X \in 2^U \mid \underline{apr}(X) = X_1 \cap Y_1, \overline{apr}(X) = X_2 \cap Y_2\} \\
&\quad = \langle X_1 \cap Y_1, X_2 \cap Y_2\rangle, \\
&\langle X_1, X_2\rangle \sqcup \langle Y_1, Y_2\rangle \\
&\quad = \{X \in 2^U \mid \underline{apr}(X) = X_1 \cup Y_1, \overline{apr}(X) = X_2 \cup Y_2\} \\
&\quad = \langle X_1 \cup Y_1, X_2 \cup Y_2\rangle, \\
&\neg\langle X_1, X_2\rangle \\
&\quad = \{X \in 2^U \mid \underline{apr}(X) = \sim X_2, \overline{apr}(X) = \sim X_1\}, \\
&\quad = \langle \sim X_2, \sim X_1\rangle. \qquad (1.15)
\end{aligned}$$

The results are also P-rough sets. The induced system $(R_{\approx}(apr), \sqcap, \sqcup)$ is a complete distributive lattice [1, 30], with zero element $[\emptyset]_{\approx}$ and unit element $[U]_{\approx}$. The corresponding order relation is called P-rough set inclusion given by:

$$\langle X_1, X_2\rangle \sqsubseteq \langle Y_1, Y_2\rangle \iff X_1 \subseteq Y_1 \text{ and } X_2 \subseteq Y_2. \qquad (1.16)$$

The system $(R_{\approx}(apr), \sqcap, \sqcup, \neg, [\emptyset]_{\approx}, [U]_{\approx})$ is called a P-rough set algebra. If equivalence relations $\approx_*$ and $\approx^*$ are used, similar structures can be obtained.

Example 1 This example illustrates the main ideas developed so far. Consider a universe consisting of three elements $U = \{a, b, c\}$ and an equivalence relation $\Re$ on U:

$$a\Re a, \quad b\Re b, \quad b\Re c, \quad c\Re b, \quad c\Re c.$$

The equivalence relation induces two equivalence classes $[a]_{\Re} = \{a\}$ and $[b]_{\Re} = [c]_{\Re} = \{b, c\}$. Table 1 summarizes the lower and upper approximations, the positive, negative and boundary regions, and the accuracy of approximations for all subsets of U. The family of all composed sets is $\mathrm{Com}(apr) = \{\emptyset, \{a\}, \{b, c\}, U\}$. It defines nine I-rough sets. Figure 2 shows the lattice formed by these I-rough sets. Based on the lower and upper approximations, a relation $\approx$ on 2^U is given by:

$$\begin{aligned}
&\emptyset \approx \emptyset, \\
&\{a\} \approx \{a\}, \\
&\{b, c\} \approx \{b, c\}, \\
&\{b\} \approx \{b\}, \quad \{b\} \approx \{c\}, \quad \{c\} \approx \{c\}, \quad \{c\} \approx \{b\}, \\
&\{a, b\} \approx \{a, b\}, \quad \{a, b\} \approx \{a, c\}, \quad \{a, c\} \approx \{a, b\}, \quad \{a, c\} \approx \{a, c\}, \\
&U \approx U.
\end{aligned}$$

X	$\underline{apr}(X)$	$\overline{apr}(X)$	POS(X)	NEG(X)	BND(X)	$\rho(X)$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	U	$\emptyset$	1
$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$	$\{b,c\}$	$\emptyset$	1
$\{b\}$	$\emptyset$	$\{b,c\}$	$\emptyset$	$\{a\}$	$\{b,c\}$	0
$\{c\}$	$\emptyset$	$\{b,c\}$	$\emptyset$	$\{a\}$	$\{b,c\}$	0
$\{a,b\}$	$\{a\}$	U	$\{a\}$	$\emptyset$	$\{b,c\}$	1/3
$\{a,c\}$	$\{a\}$	U	$\{a\}$	$\emptyset$	$\{b,c\}$	1/3
$\{b,c\}$	$\{b,c\}$	$\{b,c\}$	$\{b,c\}$	$\{a\}$	$\emptyset$	1
U	U	U	U	$\emptyset$	$\emptyset$	1

Table 1 Basic notions in Pawlak rough set model

This relation induces the following equivalence classes, i.e., P-rough sets:

$$\begin{aligned}
\langle\emptyset,\emptyset\rangle &= \{\emptyset\},\\
\langle\{a\},\{a\}\rangle &= \{\{a\}\},\\
\langle\emptyset,\{b,c\},\rangle &= \{\{b\},\{c\}\},\\
\langle\{b,c\},\{b,c\}\rangle &= \{\{b,c\}\},\\
\langle\{a\},U\rangle &= \{\{a,b\},\{a,c\}\},\\
\langle U,U\rangle &= \{U\}.
\end{aligned}$$

Figure 3 is the lattice formed by these P-rough sets. From this example, one can see that I-rough set algebra is different from the P-rough set algebra. In general, the lattice formed by P-rough sets is isomorphic to a sublattice of the lattice formed by I-rough sets.

2.2 Non-standard rough set models

The Pawlak rough set model may be extended by using an arbitrary binary relation [41, 43]. Given a binary relation $\Re$ and two elements $x, y \in U$, if $x\Re y$, we say that y is $\Re$-related to x. A binary relation may be more conveniently represented by a mapping $r : U \longrightarrow 2^U$:

$$r(x) = \{y \in U \mid x\Re y\}. \tag{1.17}$$

That is, $r(x)$ consists of all $\Re$-related elements of x. It may be interpreted as a neighborhood of x [12, 14]. If $\Re$ is an equivalence relation, $r(x)$ is the equivalence class containing x. By using the notion of neighborhoods to replace

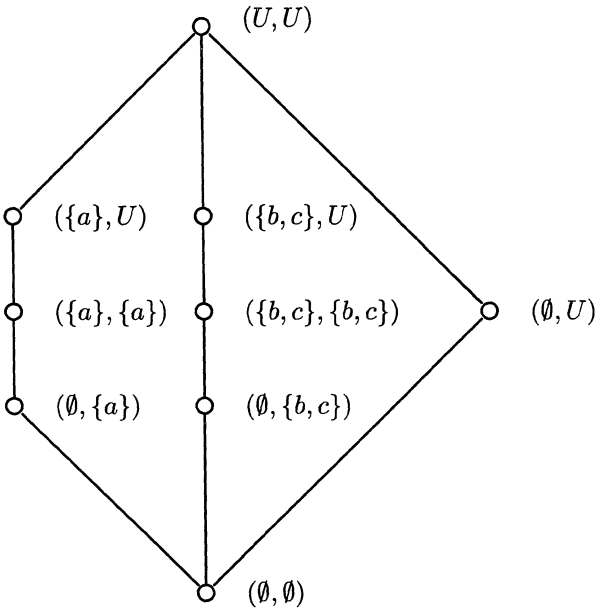


Figure 2 An example of I-rough set algebra

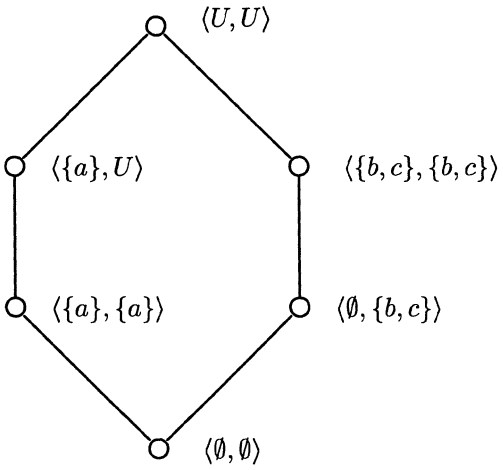


Figure 3 An example of P-rough set algebra

equivalence classes, we can extend equation (1.3) as follows:

$$\begin{aligned} \underline{apr}(X) &= \{x \mid r(x) \subseteq X\}, \\ \overline{apr}(X) &= \{x \mid r(x) \cap X \neq \emptyset\}. \end{aligned} \tag{1.18}$$

The set $\underline{apr}(X)$ consists of those elements whose $\Re$-related elements are all in X, and $\overline{apr}(X)$ consists of those elements such that at least one of whose $\Re$-related elements is in X. They are referred to as generalized approximations of X.

Generalized approximation operators do not necessarily satisfy all the properties in Pawlak rough set models. Nevertheless, properties (AL1)-(AL5) and (AU1)-(AU5) hold independent of the properties of the binary relation. Properties (AL7)-(AL10) may be used to characterize various rough set models. Such a classification of rough set models is similar to the classification of modal logics. For this purpose, we use the following properties, adopting the same labeling system from Chellas [4]:

(K)	$\underline{apr}(\sim X \cup Y) \subseteq \sim \underline{apr}(X) \cup \underline{apr}(Y)$,
(D)	$\underline{apr}(X) \subseteq \overline{apr}(X)$,
(T)	$\underline{apr}(X) \subseteq X$,
(B)	$X \subseteq \underline{apr}(\overline{apr}(X))$,
(4)	$\underline{apr}(X) \subseteq \underline{apr}(\underline{apr}(X))$,
(5)	$\overline{apr}(X) \subseteq \underline{apr}(\overline{apr}(X))$.

Property (K) does not depend on any particular binary relation. In order to construct a rough set model so that other properties hold, it is necessary to impose certain conditions on the binary relation $\Re$.

Each of the properties (D)-(5) corresponds to a property of the binary relation. Property (D) holds if $\Re$ is a serial relation, i.e., for all $x \in U$, there exists at least an element y such that $x\Re y$. Property (T) holds if $\Re$ is a reflexive relation, i.e., for all $x \in U$, $x\Re x$. Property (B) holds if $\Re$ is a symmetric relation, i.e., for all $x, y \in U$, $x\Re y$ implies $y\Re x$. Property (4) holds if $\Re$ is a transitive relation, i.e., for all $x, y, z \in U$, $x\Re y$ and $y\Re z$ imply $x\Re z$. Property (5) holds if the $\Re$ is an Euclidean relation, i.e., for all $x, y, z \in U$, $x\Re y$ and $x\Re z$ imply $y\Re z$. By combining these properties, one can construct distinct rough set models. Various rough set models are named according to the properties of the binary relation or the properties of the approximation operators. For example, a rough set model constructed from a symmetric relation is referred to as a symmetric rough set model or the KD model. If $\Re$ is reflexive and symmetric, i.e., $\Re$ is

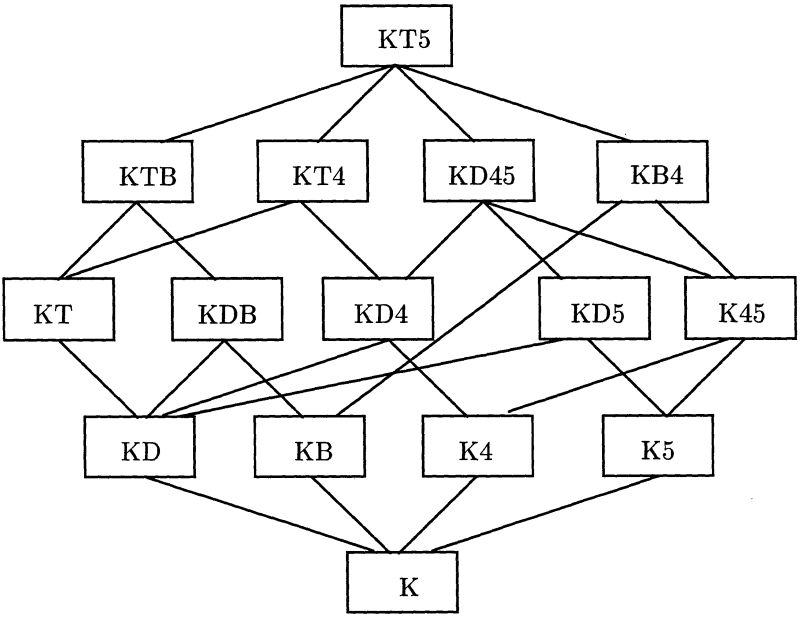


Figure 4 Rough set models

a compatibility relation, properties (K), (D), (T) and (B) hold. This model is labeled by KTB, which is the set of properties satisfied by operators $\underline{apr}$ and $\overline{apr}$. Property (D) does not explicitly appear in the label because it is implied by (T). Similarly, Pawlak rough set model is labled by KT5, which is also commonly known as S5 in modal logic.

By the results from modal logic, it is possible to construct at least fifteen distinct classes of rough set models based on the properties satisfied by the binary relation [4, 17, 43]. Figure 4, adopted from Chellas [4] and Marchal [17], summarizes the relationships between these models. A line connecting two models indicates the model in the upper level is a model in the lower level. These lines that can be derived from the transitivity are not explicitly shown. The model K may be considered as the basic and the weakest model. It does not require any special property on the binary relation. All other models are built on top the model K. The model KT5, i.e., the Pawlak rough set model, is the strongest model.

In the above formulation of rough set model, one considers only two special kinds of relationships between the neighborhood $r(x)$ of an element x and a set X to be approximated. An element belongs to the lower approximation of a subset X if *all* its $\Re$-related elements belong to X, it belongs to the upper approximation if there exists *one* elements belonging to X. The degree of overlap of X and $r(x)$ is not taken into consideration. By employing such information, graded rough set models can be obtained, in the same way graded modal logic is developed [2, 7, 9, 35, 36, 43].

Given the universe U and a binary relation $\Re$ on U, a family of graded approximation operators are defined as:

$$\begin{aligned} \underline{apr}_n(X) &= \{x \mid |r(x)| - |X \cap r(x)| \leq n\}, \\ \overline{apr}_n(X) &= \{x \mid |X \cap r(x)| > n\}. \end{aligned} \tag{1.19}$$

An element of U belongs to $\underline{apr}_n(X)$ if at most n of its $\Re$-related elements are not in X, and belongs to $\overline{apr}_n(X)$ if more than n of its $\Re$-related elements are in X. Based on the properties of binary relation, we can similarly define different classes of graded rough set models.

2.3 Rough sets in information systems

Following Lipski [16], Orlowska [20], Pawlak [21], Vakarelov [34], and Yao and Noroozi [45], we define a set-based information system to be a quadruple,

$$S = (U, At, \{V_a \mid a \in At\}, \{f_a \mid a \in At\}),$$

where

U is a nonempty set of objects,

At is a nonempty set of attributes,

V_a is a nonempty set of values for each attribute $a \in At$,

$f_a : U \longrightarrow 2^{V_a}$ is an information function for each attribute $a \in At$.

The notion of information systems provides a convenient tool for the representation of objects in terms of their attribute values. If all information functions map an object to only singleton subsets of attribute values, we obtain a degenerate set-based information system commonly used in the Pawlak rough set model. In this case, information functions can be expressed as $f_a : U \longrightarrow V_a$. In the following discussions, we only consider this kind of information systems.

We can describe relationships between objects through their attribute values. With respect to an attribute $a \in At$, a relation $\Re_a$ is given by: for $x, y \in U$,

$$x\Re_a y \Longleftrightarrow f_a(x) = f_a(y). \tag{1.20}$$

That is, two objects are considered to be indiscernible, in the view of single attribute a, if and only if they have exactly the *same* value. $\Re_a$ is an equivalence relation. The reflexivity, symmetry and transitivity of $\Re_a$ follow trivially from the properties of the relation $=$ between attribute values. For a subset of attributes $A \subseteq At$, this definition can be extended as follows:

$$x\Re_A y \Longleftrightarrow (\forall a \in A) f_a(x) = f_a(y). \tag{1.21}$$

That is, in terms of all attributes in A, x and y are indiscernible, if and only if they have the same value for every attribute in A. The extended relation is still is an equivalence relation [20].

The above discussion provides a convenient and practical method for constructing a binary relation, and in turn a Pawlak rough set model. All other notions can be easily defined. For an element $x \in U$, its equivalence class is given by:

$$r_A(x) = \{y \mid x\Re_A y\}. \tag{1.22}$$

For any subset $X \subseteq U$, the lower and upper approximations can be constructed as:

$$\begin{aligned} \underline{apr}_A(X) &= \{x \mid r_A(x) \subseteq X\}, \\ \overline{apr}_A(X) &= \{x \mid r_A(x) \cap X \neq \emptyset\}. \end{aligned} \tag{1.23}$$

As shown in the following example, different subsets of attributes may induce distinct approximation space, and hence different approximations of the same set.

Example 2 Consider the information system given in Table 2, taken from Quinlan [31]. Each object is described by three attributes. If the attribute $A = \{\text{Hair}\}$ is chosen, we can partition the universe into equivalence classes $\{o_1, o_2, o_6, o_8\}$, $\{o_3\}$, and $\{o_4, o_5, o_7\}$, reflecting the colour of Hair being blond, red and dark, respectively. With respect to the class $+ = \{o1, o_3, o_6\}$, the following approximations are obtained:

$$\begin{aligned} \underline{apr}_A(+) &= \{o_3\}, \\ \overline{apr}_A(+) &= \{o_1, o_2, o_3, o_6, o_8\}. \end{aligned}$$

Object	Height	Hair	Eyes	Classification
o_1	short	blond	blue	+
o_2	short	blond	brown	-
o_3	tall	red	blue	+
o_4	tall	dark	blue	-
o_5	tall	dark	blue	-
o_6	tall	blond	blue	+
o_7	tall	dark	brown	-
o_8	short	blond	brown	-

Table 2 An information system

Hence,

$$\begin{aligned} \mathrm{POS}_A(+) &= \underline{apr}_A(+) = \{o_3\}, \\ \mathrm{BND}_A(+) &= \overline{apr}_A(+) - \underline{apr}_A(+) = \{o_1, o_2, o_6, o_8\}, \\ \mathrm{NEG}_A(+) &= U - \overline{apr}_A(+) = \{o_4, o_5, o_7\}. \end{aligned}$$

If a set of two attributes $A' = \{\text{Hair}, \text{Eyes}\}$ is used, we have equivalence classes $\{o_1, o_6\}$, $\{o_2, o_8\}$, $\{o_3\}$, $\{o_4, o_5\}$ and $\{o_7\}$. The lower and upper approximation of + are:

$$\begin{aligned} \underline{apr}_{A'}(+) &= \{o_1, o_3, o_6\}, \\ \overline{apr}_{A'}(+) &= \{o_1, o_3, o_6\}. \end{aligned}$$

Three regions are:

$$\begin{aligned} \mathrm{POS}_{A'}(+) &= \underline{apr}_{A'}(+) = \{o_1, o_3, o_6\}, \\ \mathrm{BND}_{A'}(+) &= \overline{apr}_{A'}(+) - \underline{apr}_{A'}(+) = \emptyset, \\ \mathrm{NEG}_{A'}(+) &= U - \overline{apr}_{A'}(+) = \{o_2, o_4, o_5, o_7, o_8\}. \end{aligned}$$

From this example, it is clear that some approximation spaces are better than others.

The Pawlak rough set model can be easily generalized in information system by considering any binary relations on attribute values, instead of the trivial equality relation $=$. Suppose R_a is a binary relation on the values of an attribute $a \in At$. By extending equation (1.20), for $a \in At$ we define a binary relation on U:

$$x \Re_a y \iff f_a(x) R_a f_a(y). \tag{1.24}$$

Similarly, by extending equation (1.21), for $A \subseteq At$ we define a relation on U:

$$\begin{aligned} x\Re_A y &\iff (\forall a \in A) f_a(x) R_a f_a(y) \\ &\iff (\forall a \in A) x \Re_a y. \end{aligned} \tag{1.25}$$

An object x is related to another object y, based on an attribute a, if their values on a are related. With respect to a subset A of attributes, x is related to y if their values are related for every attribute in A. When all relations R_a are chosen to be $=$, the proposed definition reduced to the definition in the Pawlak rough set model.

The empty set $\emptyset$ produces the coarsest relation, i.e., $\Re_\emptyset = U \times U$, where $\times$ denotes the Cartesian product of sets. If the entire attribute set is used, one obtains the finest relation $\Re_{At}$. Moreover, if each object is described by an unique description, $\Re_{At}$ becomes the identity relation. The algebra $(\{\Re_A\}_{A \subseteq At}, \cap)$ is a lower semilattice with the zero element $\Re_{At}$ [20].

The relation $\Re_a$ preserves properties of R_a. Suppose R_a is a binary relation on V_a, and $\Re_a$ a binary relation on U defined by equation (1.24). Then,

a). R_a is serial $\Longrightarrow$ $\Re_a$ is serial;
b). R_a is reflexive $\Longrightarrow$ $\Re_a$ is reflexive;
c). R_a is symmetric $\Longrightarrow$ $\Re_a$ is symmetric;
d). R_a is transitive $\Longrightarrow$ $\Re_a$ is transitive;
e). R_a is Euclidean $\Longrightarrow$ $\Re_a$ is Euclidean.

The set of $\Re_A$-related objects, $r_A(x) = \{y \mid x\Re_A y\}$, can be regarded as a neighborhood of x. Likewise, the set of R_a-related values, $r_a(v) = \{v' \mid vR_a v'\}$, can be viewed as a neighborhood of v [12]. By definition, a neighborhood of objects is defined according to neighborhoods of its attribute values:

$$\begin{aligned} r_A(x) &= \{y \mid x\Re_A y\} \\ &= \bigcap_{a \in A} \{y \mid x \Re_a y\} \\ &= \bigcap_{a \in A} \{y \mid f_a(x) R_a f_a(y)\} \\ &= \bigcap_{a \in A} \{y \mid f_a(y) \in r_a(f_a(x))\}. \end{aligned} \tag{1.26}$$

This suggests that the notion of generalized rough sets is useful for approximate retrieval in information systems.

2.4 Rough set model over two universes

Recently, Wong, Wang and Yao generalized the rough set model using two distinct but related universes [38, 39, 47]. Let U and V represent two finite universes of interest. Suppose the relationships between elements of the two universes are described by a compatibility relation [32]. The formulation and interpretation of U and V and the compatibility relation between the two universes depend very much on the available knowledge and the domain of applications. For example, in a medical diagnosis system, U can be a set of symptoms and V a set of diseases. A symptom $u \in U$ is said to be compatible with a disease $v \in V$ if any patient with symptom u may have contracted the disease v. An element $u \in U$ is compatible with an element $v \in V$, written $u \; \mathcal{C} \; v$, if the u is related to v. Without loss of generality, we may assume that for any $u \in U$ there exists a $v \in V$ with $u \; \mathcal{C} \; v$, and vice versa.

A compatibility relation $\mathcal{C}$ between U and V can be equivalently defined by a multi-valued mapping, $\gamma : U \longrightarrow 2^V$, as [5, 32]:

$$\gamma(u) = \{v \in V \mid u \; \mathcal{C} \; v\}. \tag{1.27}$$

That is, $\gamma(u)$ is a subset of V consisting of all elements compatible with u. Based on this multi-valued mapping, a subset $X \subseteq V$ may be represented in terms of these elements of U compatible with the elements in X. For example, a particular group of diseases may be described by the symptoms compatible with them. Since the induced multi-valued mapping is not necessarily an one-to-one mapping, one may not be able to derive an exact representation for any subset $X \subseteq V$. By extending notion of approximation operators in rough set model, we define a pair of lower and upper approximations:

$$\begin{aligned} \underline{apr}(X) &= \{u \in U \mid \gamma(u) \subseteq X\}, \\ \overline{apr}(X) &= \{u \in U \mid \gamma(u) \cap X \neq \emptyset\}. \end{aligned} \tag{1.28}$$

The set $\underline{apr}(X)$ consists of the elements in U compatible with only those elements in X, while the set $\overline{apr}(X)$ consists of the elements in U compatible with at least one element in X. Therefore, the lower approximation $\underline{apr}(X)$ can be interpreted as the pessimistic description and the upper approximation $\overline{apr}(X)$ as the optimistic description of X. These approximation operators satisfy properties similar to (AL1)-(AL6) and (AU1)-(AU6). Since two universes are involved, there do not exist properties similar to (AL7)-(AL10) and (AU7)-(AU10).

2.5 Rough set model over Boolean algebras

Recall that the set of composed sets Com(apr) forms a sub-algebra of the Boolean algebra of the power set. One can easily formulate Pawlak rough set model in a wider context of Boolean algebra. Suppose $(\mathcal{A}, \wedge, \vee, \neg, 0, 1)$ is a Boolean algebra and $(\mathcal{B}, \wedge, \vee, \neg, 0, 1)$ is a sub-algebra. In terms of elements of $\mathcal{B}$, one may approximate any element of $\mathcal{A}$ using a pair of lower and upper approximations: for $a \in \mathcal{A}$,

$$\begin{aligned} \underline{apr}(a) &= \bigvee\{b \mid b \in \mathcal{B}, b \preceq a\}, \\ \overline{apr}(a) &= \bigwedge\{b \mid b \in \mathcal{B}, a \preceq b\}. \end{aligned} \tag{1.29}$$

Clearly, this definition reduce to Pawlak's original proposal if $\mathcal{A}$ is chosen to be 2^U and $\mathcal{B}$ is chosen to be Com(apr).

Wong, Wang and Yao [39] extended the above formulation further by considering two arbitrary Boolean algebras. Suppose $\underline{f}\colon \mathcal{A} \longrightarrow \mathcal{B}$ and $\overline{f}\colon \mathcal{A} \longrightarrow \mathcal{B}$ are two mappings from a Boolean algebra $(\mathcal{A}, \vee, \wedge, \neg, 0, 1)$ to another Boolean algebra $(\mathcal{B}, \vee, \wedge, \neg, 0, 1)$. We say that $\underline{f}$ and $\overline{f}$ are dual mappings if $\overline{f}(a) = \neg\underline{f}(\neg a)$ for every $a \in \mathcal{A}$. The pair of dual mappings form an interval structure if they satisfy the following axioms:

$$\begin{aligned} &\text{(IL1)} && \underline{f}(a) \wedge \underline{f}(b) = \underline{f}(a \wedge b), \\ &\text{(IL2)} && \underline{f}(0) = 0, \\ &\text{(IL3)} && \underline{f}(1) = 1, \\ &\text{(IU1)} && \overline{f}(a \vee b) = \overline{f}(a) \vee \overline{f}(b), \\ &\text{(IU2)} && \overline{f}(0) = 0, \\ &\text{(IU3)} && \overline{f}(1) = 1. \end{aligned}$$

These properties indeed correspond to properties (AL3), (AL2), (AL6), (AU3), (AU2) and (AU6).

An alternate way of defining an interval structure is through another mapping $j\colon \mathcal{A} \longrightarrow \mathcal{B}$ satisfying the axioms:

$$\begin{aligned} &\text{(A1)} && j(0) = 0, \\ &\text{(A2)} && \bigvee_{a \in \mathcal{A}} j(a) = 1, \\ &\text{(A3)} && a \neq b \Longrightarrow j(a) \wedge j(b) = 0. \end{aligned}$$

This mapping is called a *basic assignment*, and an element $a \in \mathcal{A}$ with $j(a) \neq 0$ is called a *focal element*. From a given j, one can define a mapping $\underline{f}$: for all

$a \in \mathcal{A}$,

$$\underline{f}(a) = \bigvee_{b \preceq a} j(b), \tag{1.30}$$

and another mapping $\overline{f}$ by the relationship $\overline{f}(a) = \neg \underline{f}(\neg a)$. The mapping $\overline{f}$ can be equivalently defined by:

$$\overline{f}(a) = \bigvee_{a \wedge b \neq 0} j(b). \tag{1.31}$$

Conversely, given an interval structure $(\underline{f}, \overline{f})$, we can construct the basic assignment j by the formula: for all $a \in \mathcal{A}$,

$$j(a) = \underline{f}(a) \wedge \neg(\bigvee_{b \prec a} \underline{f}(b)). \tag{1.32}$$

Rough set models on the same universe and on two universes are only special cases of this general framework. Based on the axioms of an interval structure, the above developed relationships hold in any rough set model that is stronger than the KD model. More specifically, we have the following connections:

$$\begin{aligned} j(X) &= \{x \mid r(x) = X\}, \\ j(X) &= \underline{apr}(X) - \bigcap_{Y \subset X} \underline{apr}(Y), \\ \underline{apr}(X) &= \bigcup_{Y \subseteq X} j(X), \\ \overline{apr}(X) &= \bigcup_{Y \cap X \neq \emptyset} j(X). \end{aligned} \tag{1.33}$$

Therefore, the basic assignment provides another representation of approximation operators.

3 PROBABILISTIC ROUGH SET MODELS

Based on the notion of rough membership functions, we review two different approaches for the construction of probabilistic rough set model. One is related to probabilistic modal logic and the other is based on decision theory.

3.1 Rough membership functions

Pawlak and Skowron [28], Pawlak *et al.* [29] and Wong and Ziarko [40] proposed another way to characterize a rough set by a single membership function. For any $X \subseteq U$, a rough membership function is defined by:

$$\mu_X(x) = \frac{|X \cap [x]_{\Re}|}{|[x]_{\Re}|}. \tag{1.34}$$

By definition, elements in the same equivalent class have the same degree of membership. The rough membership $\mu_X(x)$ may be interpreted as the probability of x belonging to X given that x belongs to an equivalence class. This interpretation leads to probabilistic rough sets [29, 40]. Like the algebraic rough set model, the intersection and union of probabilistic rough sets are not truth-functional. Nevertheless, we have:

(m1) $\mu_X(x) = 1 \Longleftrightarrow x \in \text{POS}(X)$,

(m2) $\mu_X(x) = 0 \Longleftrightarrow x \in \text{NEG}(X)$,

(m3) $0 < \mu_X(x) < 0 \Longleftrightarrow x \in \text{BND}(X)$,

(m4) $\mu_{\sim X}(x) = 1 - \mu_X(x)$,

(m5) $\mu_{X \cup Y}(x) = \mu_X(x) + \mu_Y(x) - \mu_{X \cap Y}(x)$,

(m6) $\max(0, \mu_X(x) + \mu_Y(x) - 1) \leq \mu_{X \cap Y}(x) \leq \min(\mu_X(x), \mu_Y(x))$,

(m7) $\max(\mu_X(x), \mu_Y(x)) \leq \mu_{X \cup Y}(x) \leq \min(1, \mu_X(x) + \mu_Y(x))$.

They follow from the property of probability. The definition in equation (1.34) can be easily extended by using an arbitrary binary relation.

3.2 Variable precision rough set model

In the definition of graded rough set models, the size of $r(x)$ is not taken into consideration. By using such information, we can define variable precision, or probabilistic, rough set model [40, 49], in parallel to probabilistic modal logic [8, 10, 19, 43].

With respect to the universe U and a binary relation $\Re$ on U, we define a family of probabilistic rough set operators:

$$\begin{aligned} \underline{apr}_{\alpha}(X) &= \{x \mid \frac{|X \cap r(x)|}{|r(x)|} \geq 1 - \alpha\}, \\ \overline{apr}_{\alpha}(X) &= \{x \mid \frac{|X \cap r(x)|}{|r(x)|} > \alpha\}. \end{aligned} \tag{1.35}$$

By definition, for a serial binary relation and $\alpha \in [0,1]$, probabilistic rough set operators satisfy the following properties:

$$
\begin{array}{ll}
\text{(PL0)} & \underline{apr}(X) = \underline{apr}_0(X), \\
\text{(PL1)} & \underline{apr}_\alpha(X) = \sim\overline{apr}_\alpha(\sim X), \\
\text{(PL2)} & \underline{apr}_\alpha(U) = U, \\
\text{(PL3)} & \underline{apr}_\alpha(X \cap Y) \subseteq \underline{apr}_\alpha(X) \cap \underline{apr}_\alpha(Y), \\
\text{(PL4)} & \underline{apr}_\alpha(X \cup Y) \supseteq \underline{apr}_\alpha(X) \cup \underline{apr}_\alpha(Y), \\
\text{(PL5)} & X \subseteq Y \Longrightarrow \underline{apr}_\alpha(X) \subseteq \underline{apr}_\alpha(Y), \\
\text{(PL6)} & \alpha \geq \beta \Longrightarrow \underline{apr}_\alpha(X) \supseteq \underline{apr}_\beta(X), \\
\text{(PU0)} & \overline{apr}(X) = \overline{apr}_0(X), \\
\text{(PU1)} & \overline{apr}_\alpha(X) = \sim\underline{apr}_\alpha(\sim X), \\
\text{(PU2)} & \overline{apr}_\alpha(\emptyset) = \emptyset, \\
\text{(PU3)} & \overline{apr}_\alpha(X \cup Y) \supseteq \overline{apr}_\alpha(X) \cup \overline{apr}_\alpha(Y), \\
\text{(PU4)} & \overline{apr}_\alpha(X \cap Y) \subseteq \overline{apr}_\alpha(X) \cap \overline{apr}_\alpha(Y), \\
\text{(PU5)} & X \subseteq Y \Longrightarrow \overline{apr}_\alpha(X) \subseteq \overline{apr}_\alpha(Y), \\
\text{(PU6)} & \alpha \geq \beta \Longrightarrow \overline{apr}_\alpha(X) \subseteq \overline{apr}_\beta(X).
\end{array}
$$

Moreover, for $0 \leq \alpha < 0.5$,

$$\text{(PD)} \qquad \underline{apr}_\alpha(X) \subseteq \overline{apr}_\alpha(X),$$

which may be interpreted as a probabilistic version of axiom (D). In this case, one can also partition the into three regions based on the value of α:

$$
\begin{aligned}
\mathrm{POS}_\alpha(X) &= \underline{apr}_\alpha(X), \\
\mathrm{NEG}_\alpha(X) &= U - \overline{apr}_\alpha(X), \\
\mathrm{BND}_\alpha(X) &= \overline{apr}_\alpha(X) - \underline{apr}_\alpha(X).
\end{aligned} \tag{1.36}
$$

They may be referred to as the probabilistic positive, negative and boundary regions. In the following subsection, we will show that the value of α can be determined within the framework of decision theory.

3.3 Rough set model based on decision theory

In the variable precision rough set model, the universe is partitioned into three regions. The same goal can be achieved by using rough membership functions in the framework of decision theory [46]. In terms of decision-theoretic language,

we have a set of states $\Omega = \{X, \neg X\}$, indicating that an element belongs to and does not belong to X, and the set of actions $A = \{a_1, a_2, a_3\}$, representing the three actions, deciding POS(X), deciding NEG(X), and deciding BND(X), respectively.

Let $\lambda(a_i|X)$ denote the loss incurred for taking action a_i when an object in fact belongs to X, and let $\lambda(a_i|\neg X)$ denote the loss incurred when the object actually belongs to $\neg X$. $P(X \mid r(x))$ and $P(\neg X \mid r(x))$ are the probabilities that an object with neighborhood $r(x)$ belongs to X and $\neg X$, respectively. They are in fact the rough membership functions with respect to X and $\neg X$. Thus, the expected loss $R(a_i|r(x))$ associated with taking the individual actions can be expressed as:

$$\begin{aligned} R(a_1|r(x)) &= \lambda_{11}P(X \mid r(x)) + \lambda_{12}P(\neg X \mid r(x)), \\ R(a_2|r(x)) &= \lambda_{21}P(X \mid r(x)) + \lambda_{22}P(\neg X \mid r(x)), \\ R(a_3|r(x)) &= \lambda_{31}P(X \mid r(x)) + \lambda_{32}P(\neg X \mid r(x)), \end{aligned} \tag{1.37}$$

where $\lambda_{i1} = \lambda(a_i|X)$, $\lambda_{i2} = \lambda(a_i|\neg X)$, and $i = 1, 2, 3$. The Bayesian decision procedure leads to the following minimum-risk decision rules:

(P) Decide POS(X) if
$$R(a_1|r(x)) \leq R(a_2|r(x)) \text{ and } R(a_1|r(x)) \leq R(a_3|r(x));$$

(N) Decide NEG(X) if
$$R(a_2|r(x)) \leq R(a_1|r(x)) \text{ and } R(a_2|r(x)) \leq R(a_3|r(x));$$

(B) Decide BND(X) if
$$R(a_3|r(x)) \leq R(a_1|r(x)) \text{ and } R(a_3|r(x)) \leq R(a_2|r(x)).$$

Since $P(X \mid r(x)) + P(\neg X \mid r(x)) = 1$, the above decision rules can be simplified so that only the probabilities $P(X \mid r(x))$ are involved. Thus, we can classify any object with neighborhood $r(x)$ based only on the probabilities $P(X \mid r(x))$, i.e., the rough membership function, and the given loss function λ_{ij} ($i = 1, 2, 3$; $j = 1, 2$).

Consider a special kind of loss functions with $\lambda_{11} \leq \lambda_{31} < \lambda_{21}$ and $\lambda_{22} \leq \lambda_{32} < \lambda_{12}$. The loss of classifying an object x belonging to X into the positive region POS(X) is less than or equal to the loss of classifying x into the boundary region BND(X), and both of these losses are strictly less than the loss of classifying x into the negative region NEG(X). We obtain the reverse order of losses by classifying an object that does not belong to X. For this type of loss functions, the minimum-risk decision rules (P)-(B) can be written as:

(P) Decide POS(X) if $P(X \mid r(x)) \geq \beta$ and $P(X \mid r(x)) \geq \gamma$;

(N) Decide NEG(X) if $P(X \mid r(x)) \leq \gamma$ and $P(X \mid r(x)) \leq \delta$;
(B) Decide BND(X) if $\delta \leq P(X \mid r(x))$ and $P(X \mid r(x)) \leq \beta$;

where

$$\begin{aligned} \beta &= \frac{\lambda_{12} - \lambda_{32}}{(\lambda_{31} - \lambda_{11}) + (\lambda_{12} - \lambda_{32})}, \\ \gamma &= \frac{\lambda_{12} - \lambda_{22}}{(\lambda_{21} - \lambda_{11}) + (\lambda_{12} - \lambda_{22})}, \\ \delta &= \frac{\lambda_{32} - \lambda_{22}}{(\lambda_{21} - \lambda_{31}) + (\lambda_{32} - \lambda_{22})} \end{aligned} \tag{1.38}$$

From the assumptions, $\lambda_{11} \leq \lambda_{31} < \lambda_{21}$ and $\lambda_{22} \leq \lambda_{32} < \lambda_{12}$, it follows that $\beta \in (0,1]$, $\gamma \in (0,1)$, and $\delta \in [0,1)$. Decision rules (P)-(B) depend only on the parameters β, γ, and δ computable from the λ_{ij}'s directly supplied by the user.

If $\delta \leq \beta$, $\delta \leq \gamma \leq \beta$. By decision rules (P)-(B), three regions can be determined by δ and β. If $\beta < \delta$, we have $\beta < \gamma < \delta$. According to (P)-(B), the boundary region is empty, and both positive and negative region can be determined by γ. To be consistent with the variable precision rough set model, we assume $\delta < \beta$, which implies $\delta < \gamma < \beta$. Furthermore, we choose a tie-breaking rule to differentiate actions producing the same risk. If the risk of deciding POS(X) or BND(X) is the same, we decide POS(X); if the risk of deciding NEG(X) or BND(X) is the same, we decide NEG(X). Under these assumptions, (P)-(B) can be simplified into:

(P) Decide POS(X) if $P(X \mid r(x)) \geq \beta$;
(N) Decide NEG(X) if $P(X \mid r(x)) \leq \delta$;
(B) Decide BND(X) if $\delta < P(X \mid r(x)) < \beta$.

The positive, negative, and boundary regions can be explicitly expressed in terms of the pair of parameters δ and β, namely:

$$\begin{aligned} \mathrm{POS}_{\beta,\delta}(X) &= \{x \mid P(X \mid r(x)) \geq \beta\}, \\ \mathrm{NEG}_{\beta,\delta}(X) &= \{x \mid P(X \mid r(x)) \leq \delta\}, \\ \mathrm{BND}_{\beta,\delta}(X) &= \{x \mid \delta < P(X \mid r(x)) < \beta\}. \end{aligned} \tag{1.39}$$

The lower and upper approximations $\underline{apr}_{\beta,\delta}(X)$ and $\overline{apr}_{\beta,\delta}(X)$ of X can be defined as:

$$\begin{aligned} \underline{apr}_{\beta,\delta}(X) &= \mathrm{POS}_{\beta,\delta}(X) \\ &= \{x \mid P(X \mid r(x)) \geq \beta\}, \\ \overline{apr}_{\beta,\delta}(X) &= \mathrm{POS}_{\beta,\delta}(X) \cup \mathrm{BND}_{\beta,\delta}(X) \\ &= \{x \mid P(X \mid r(x)) > \delta\}. \end{aligned} \tag{1.40}$$

Now assume the following condition:

$$\frac{\lambda_{12} - \lambda_{32}}{\lambda_{31} - \lambda_{11}} = \frac{\lambda_{21} - \lambda_{31}}{\lambda_{32} - \lambda_{22}}. \tag{1.41}$$

We have $\beta = 1 - \delta$. Let $\alpha = \delta$. The lower and upper approximations can be expressed by:

$$\begin{aligned} \underline{apr}_{1-\alpha,\alpha}(X) &= \{x \mid P(X \mid r(x)) \geq 1 - \alpha\}, \\ \overline{apr}_{1-\alpha,\alpha}(X) &= \{x \mid P(X \mid r(x)) > \alpha\}. \end{aligned} \tag{1.42}$$

They are exactly the probabilistic approximations given in equation (1.35) if the required probabilities are estimated from the cardinalities of $X \cap r(x)$ and $r(x)$, namely, $P(X \mid r(x)) = |X \cap r(x)|/|r(x)|$. The approximations of in an algebraic rough set model can be easily derived. Consider the following loss function:

$$\lambda_{12} = \lambda_{21} = 1, \quad \lambda_{11} = \lambda_{22} = \lambda_{31} = \lambda_{32} = 0. \tag{1.43}$$

This means that there is a unit cost if an object belonging to X is classified into the negative region or if an object not belonging to X is classified into the positive region; otherwise there is no cost. For such a loss function, we obtain from equation (1.38) that $\beta = 1$ and $\delta = 0$. Hence, according to equation (1.40), we have:

$$\begin{aligned} \underline{apr}_{1,0}(X) &= \{x \mid P(X \mid r(x)) = 1\}, \\ \overline{apr}_{1,0}(X) &= \{x \mid P(X \mid r(x)) > 0\}. \end{aligned} \tag{1.44}$$

With the probabilities estimated by

$$P(X \mid r(x)) = \frac{|X \cap r(x)|}{|r(x)|}, \tag{1.45}$$

$\underline{apr}_{1,0}(X)$ and $\overline{apr}_{1,0}(X)$ can be expressed as:

$$\begin{aligned} \underline{apr}_{1,0}(X) &= \{x \mid r(x) \subseteq X\}, \\ \overline{apr}_{1,0}(X) &= \{x \mid r(x) \cap X \neq \emptyset\}. \end{aligned} \tag{1.46}$$

The results given here suggest that both algebraic rough set and probabilistic rough set models can be viewed as a special case of the decision theoretic framework.

4 CONCLUSION

In the Pawlak rough set model, an equivalent relation is used to define an approximation space. Following the argument of Pawlak and using an arbitrary binary relation, one can derive various type of generalized rough set models. Alternatively, one may also generalize Pawlak rough set model by using statistical information. Based on the properties of binary relation, one can identify the properties of lower and upper approximations. Generalized rough set models may be grouped into two classes, the algebraic and probabilistic rough set models, depending on whether statistical information is used. The algebraic class includes normal rough set models, graded rough set models, rough set models over two universes, and rough set models over Boolean algebras. The probabilistic rough set models may be interpreted based on rough membership functions.

The successful applications of the theory of rough sets depends to a large extent on the formulation, characterization, and interpretation of the theory. In this paper, existing works are reviewed using a very simple, and unified, view. That is, rough set models are constructed, classified, and interpreted based on the notion of binary relations. This view in may be useful in the applications of the theory of rough sets.

REFERENCES

[1] Bonikowski, Z., "Algebraic structures of rough sets," in: Rough Sets, Fuzzy Sets and Knowledge Discovery, edited by W.P. Ziarko, Springer-Verlag, London, pp. 242-247, 1994.

[2] de Caro, F., "Graded modalities II," Studia Logica, **XLVII**, pp. 1-10, 1988.

[3] Chanas, S. and Kuchta, D., "Further remarks on the relation between rough and fuzzy sets," Fuzzy Sets and Systems, **47**, pp. 391-394, 1992.

[4] Chellas, B.F., Modal Logic: An Introduction, Cambridge University Press, Cambridge, 1980.

[5] Dempster, A.P., "Upper and lower probabilities induced by a multivalued mapping," Annals of Mathematical Statistics, **38**, pp. 325-339, 1967.

[6] Dubois, D. and Prade, H. "Rough fuzzy sets and fuzzy rough sets," it International Journal of General Systems, **17**, pp. 191-209, 1990.

[7] Fattorosi-Barnaba, M. and de Caro, F., "Graded modalities I," Studia Logica, **XLIV**, pp. 197-221, 1985.

[8] Fattorosi-Barnaba, M. and Amati, G., "Modal operators with probabilistic interpretations I," Studia Logica, **XLVI**, pp. 383-393, 1987.

[9] Fattorosi-Barnaba, M. and de Caro, F., "Graded modalities III," Studia Logica, **XLVII**, pp. 99-110, 1988.

[10] Hart, W.D., "Probability as degree of possibility," Notre-Dame Journal of Formal Logic, **13**, pp. 286-288, 1972.

[11] Iwinski, T.B., "Algebraic approach to rough sets," Bulletin of the Polish Academy of Sciences, Mathematics, **35**, pp. 673-683, 1987.

[12] Lin, T.Y., "Neighborhood systems and approximation in database and knowledge base systems," Proceedings of the Fourth International Symposium on Methodologies of Intelligent Systems, 1989.

[13] Lin, T.Y. (Ed.) Proceedings of the CSC'95 Workshop on Rough Sets and Database Mining, San Jose State University, 1995.

[14] Lin, T.Y. and Liu, Q., "Rough approximate operators: axiomatic rough set theory," in: Rough Sets, Fuzzy Sets and Knowledge Discovery, edited by W.P. Ziarko, Springer-Verlag, London, pp. 256-260, 1994.

[15] Lin, T.Y. and Wildberger, A.M. (Ed.) Soft Computing, The Society for Computer Simulation, San Diego, 1995.

[16] Lipski, W. Jr., "On databases with incomplete information," Journal of the ACM, **28**, 41-70, 1981.

[17] Marchal, B. "Modal logic – a brief tutorial," in: Non-Standard Logics for Automated Reasoning, edited by Smets, P., Mamdani, A., Dubois, D. and Prade, H., New York, Academic Press, pp. 15-23, 1988.

[18] Marczewski, E. and Steinhaus, H., "On a certain distance of sets and the corresponding distance of functions," Colloquium Mathemmaticum, **6**, pp. 319-327, 1958.

[19] Murai, T., Miyakoshi, M. and Shimbo, M., "Fuzzification of modal operators from the standpoint of fuzzy semantics," Proceedings of the Second International Fuzzy Systems Association Congress, pp. 430-433, 1987.

[20] Orlowska, E., "Logic of indiscernibility relations," Bulletin of the Polish Academy of Sciences, Mathematics, **33**, pp. 475-485, 1985.

[21] Pawlak, Z., "Information systems – theoretical foundations," Information Systems, **6**, pp. 205-218, 1981.

[22] Pawlak, Z., "Rough sets," International Journal of Computer and Information Sciences, **11**, pp. 341-356, 1982.

[23] Pawlak, Z. "Rough classification," International Journal of Man-Machine Studies, **20**, pp. 469-483, 1984.

[24] Pawlak, Z. "Rough sets and fuzzy sets," Fuzzy Sets and Systems, **17**, pp. 99-102, 1985.

[25] Pawlak, Z. Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Boston, 1991.

[26] Pawlak, Z. "Rough sets: a new approach to vagueness," in: Fuzzy Logic for the Management of Uncertainty, edited by L.A. Zadeh and J. Kacprzyk, Eds., John Wiley & Sons, New York, pp. 105-118, 1992.

[27] Pawlak, Z. "Hard and soft sets," in: Rough Sets, Fuzzy Sets and Knowledge Discovery, edited by W.P. Ziarko, Springer-Verlag, London, pp. 130-135, 1994.

[28] Pawlak, Z. and Skowron, A., "Rough membership functions," in: Fuzzy Logic for the Management of Uncertainty, edited by L.A. Zadeh and J. Kacprzyk, John Wiley & Sons, New York, pp. 251-271, 1994.

[29] Pawlak, Z., Wong, S.K.M. and Ziarko, W., "Rough sets: probabilistic versus deterministic approach," International Journal of Man-machine Studies, **29**, pp. 81-95, 1988.

[30] Pomykala, J. and Pomykala, J.A., "The Stone algebra of rough sets," Bulletin of the Polish Academy of Sciences, Mathematics, **36**, pp. 495-508, 1988.

[31] Quinlan, J.R., "Learning efficient classification procedures and their application to chess endgames," in: Machine Learning: An Artificial Intelligence Approach, vol. 1, edited by Michalski, J.S., Carbonell, J.G., and Mirchell, T.M., Morgan Kaufmann, Palo Alto, CA, pp. 463-482, 1983.

[32] Shafer, G., "Belief functions and possibility measures," in: Analysis of Fuzzy Information, vol. 1: Mathematics and Logic, edited by Bezdek, J.C., CRC Press, Boca Raton, pp. 51-84, 1987.

[33] Slowinski, R. (Ed.) Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, Boston, 1992.

[34] Vakarelov, D., "A modal logic for similarity relations in Pawlak knowledge representation systems," Fundamenta Informaticae, **XV**, pp. 61-79, 1991.

[35] van der Hoek, W., "On the semantics of graded modalities," Journal of Applied Non-classical Logic, **2**, pp. 81-123, 1992.

[36] van der Hoek, W. and Meyer, J.-J.Ch., "Graded modalities in epistemic logic," Logique et Analyse, **133-134**, pp. 251-270, 1991.

[37] Wasilewska, A. Topological rough algebras. Manuscript, 1995.

[38] Wong, S.K.M., Wang, L.S. and Yao, Y.Y., "Interval structure: a framework for representing uncertain information," Uncertainty in Artificial Intelligence: Proceedings of the 8th Conference, pp. 336-343, 1993.

[39] Wong, S.K.M. and Wang, L.S. and Yao, Y.Y., "On modeling uncertainty with interval structures," Computational Intelligence, **11**, pp. 406-426, 1995.

[40] Wong, S.K.M. and Ziarko, W., "Comparison of the probabilistic approximate classification and the fuzzy set model," Fuzzy Sets and Systems, **21**, pp. 357-362, 1987.

[41] Wybraniec-Skardowska, U., "On a generalization of approximation space," Bulletin of the Polish Academy of Sciences, Mathematics, **37**, pp. 51-61, 1989.

[42] Yao, Y.Y. "On combining rough and fuzzy sets," Proceedings of the CSC'95 Workshop on Rough Sets and Database Mining, edited by Lin, T.Y., San Jose State University, 1995.

[43] Yao, Y.Y. and Lin, T.Y., "Generalization of rough sets using modal logic," Intelligent Automation and Soft Computing, An International Journal, to appear.

[44] Yao, Y.Y., and Lin, T.Y., "Approximate reasoning using rough-set theory," Proceedings of International Symposia on Soft Computing and Intelligent Industrial Automation, pp., 1996.

[45] Yao, Y.Y. and Noroozi, N., "A unified model for set-based computations," in: Soft Computing, edited by Lin, T.Y. and Wildberger, A.M., The Society for Computer Simulation, San Diego, pp. 252-255, 1995.

[46] Yao, Y.Y. and Wong, S.K.M., "A decision theoretic framework for approximating concepts," International Journal of Man-machine Studies, **37**, pp. 793-809, 1992.

[47] Yao, Y.Y., Wong, S.K.M. and Wang, L.S., "A non-numeric approach to uncertain reasoning," International Journal of General Systems, **23**, pp. 343-359, 1995.

[48] Zakowski, W. "Approximations in the space (U, Π)," Demonstratio Mathematica, **XVI**, pp. 761-769, 1983.

[49] Ziarko, W., "Variable precision rough set model," Journal of Computer and System Sciences, **46**, pp. 39-59, 1993.

[50] Ziarko, W.P. (Ed.) Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer-Verlag, London, 1994.

4

ROUGH CONTROL: A PERSPECTIVE

Toshinori Munakata

Computer and Information Science Department
Cleveland State University, Cleveland, OH 44115

ABSTRACT

Observing the current state of commercial and industrial AI, control and hybrid systems are said to have the highest potentials for massive practical applications of rough set theory. After a brief description of the control problem and fuzzy systems, the principles of rough control and a scenario of fine temperature control are discussed.

1 INTRODUCTION

The term "rough control" refers to applications of the rough set theory to control problems. The objective of this article is to show that there are compelling reasons to pursue rough control and to explore a preliminary survey for its potentials of truly practical applications.

The field of rough sets is relatively new and has remained unknown to most of the computing community. There appears, however, to be a growing interest among many researchers recently. Most research works in the field of rough sets so far have been in "symbolic" approaches such as data and decision analysis, databases, knowledge based systems and machine learning. Also, more emphasis appears to have been placed on theoretical aspects rather than everyday commercial and industrial applications. No one can over-emphasize the importance of the theoretical foundation of any field; without sound foundation, this field would be a castle on sand. On the other hand, one would doubt the true value of any theory if it does not offer any practical applications. For a field to prosper, we need balanced successes on both theory and applications.

There has been some research on applications of rough control [1-6]. However, their number and application domains have been relatively few. This fact has been confirmed by my recent contacts with experts in rough sets at the RSSC'94 [7]. Also, my search in a national literature database has yielded a relatively small number of publications. Furthermore, some of the works in rough control are published in form of technical reports, which are not very visible to the world computing community. To promote rough control, quality articles should be published in more widely circulated international journals. Also, after a preliminary version of this article was presented at the ACM CSC'95 in March 1995, a Rough Control Group was initiated and I have been named as the first Chair. The major objective of the group is to pursue research and development of rough control and coordinate such efforts. Currently about 60 researchers worldwide are in the group.

Whether rough control will succeed is yet to be seen, but there are several reasons for which at least one should pursue its potentials. For the past couple of years, I have been serving as the Guest Editor of two special issues for commercial and industrial AI for the Communications of the ACM [8, 9]. In the second issue, an article for rough sets by Zdzislaw Pawlak, et.al. will appear [10]. This should introduce the theory to a wide spectrum of audience worldwide. In these two issues, I have assembled some of the foremost minds in AI to author and/or review about 23 articles. Through this extensive experience, I have observed practical and not-so-practical application domains within AI. The following Table 1 is a summary of my speculation regarding ratings on potentials of successful everyday applications of rough set theory [11].

Control has been the most successful application domain for recently evolved AI areas, such as fuzzy sets and chaos theory [12, 13]. Control is also one of the most practical application domains for neural networks [14]. It would be too naive to assume that these success stories in fuzzy sets, chaos, and neural networks will also be repeated for rough sets. However, observing these successful applications, control definitely deserves attention. There are several reasons to further support this claim: for example, there are some similarities between fuzzy and rough sets. Simply stated, control is a mapping problem from inputs to outputs. When compared with symbolic AI, the effectiveness of control is often easier to prove than, say, symbolic expert systems. If a machine can operate 5%, or even 1%, more efficiently than before, we do not need to elaborate words to explain its validity.

Table 1 A Future Perspective on Successful Industrial Applications of Rough Sets. (On scale of 0 to 10, where 0 least and 10 most)

Application Area	Success Expectation
Use of common sense	0
Machine learning	3
Expert systems	5
Control	7
Hybrid with other existing systems (e.g., fuzzy systems)	8

2 THE CONTROL PROBLEM

"Control" in this article refers to control of the various physical, chemical, or other numeric characteristics, such as temperature, electric current, flow of liquid/gas, motion of machines, various business and financial quantities (e.g., flow of cash, inventory control), etc. A control system can be abstracted as a box for which inputs are flowing into it, and outputs are emerging from it. Parameters can be included as parts of inputs or within the box, i.e., the control system.

For example, consider a system that controls room temperature by a heat source. The inputs may be the current room temperature and a parameter representing a target temperature. The output can be the amount of the heat source to be applied. The control problem in general is to determine the numeric values of the outputs for given values of the inputs. That is, the problem is to develop a formula or algorithm for mapping from the inputs to the outputs.

Although the statement of the control problem is straightforward, achieving good control is not necessarily a simple matter for several reasons. For example, bad control can be time consuming or inefficient, it may unnecessarily fluctuate before reaching the target, or even worse, it may become unstable. A good

algorithm will be relatively simple yet it performs efficient and stable control. For easy problems, a simple mathematical formulation may be sufficient. When problems get harder, traditional control techniques such as PID (proportional, integral, and differential) may not work well. This is the type of problems where fuzzy control has been successful, and where our major target domain is centered for rough control as well.

3 THE CASE OF FUZZY CONTROL

Although fuzzy and rough sets are different, there are some similarities. Fuzzy control has already been successful in many applications, thus a reasonable approach exploring rough control would be to examine in what types of application domains and how or why fuzzy control has been successful.

Typical situations where fuzzy control are particularly successful are difficult cases where traditional control methods do not work well. For example, the control rules may be so complex that mathematical formulation is either impossible, or even if it is possible, it is too complicated or costly for practical applications. For such situations, fuzzy approaches allow us to represent descriptive or qualitative expressions such as "slow" or "moderately fast." These expressions are much closer in spirit of human thinking and natural language, and are easily incorporated with symbolic statements in form of fuzzy logic. Fuzzy systems are also suitable for uncertain or approximate reasoning. For example, the input and parameter values of a system may involve fuzziness, inaccuracy, or incompleteness. Similarly, the control rules that derive output values may also be incomplete or inaccurate. Fuzzy logic allows decision making with estimated values under incomplete information.

In the following, we will illustrate the basic ingredients of fuzzy control by a simple example. (For more on basics, see Tutorial of [12]; for more extensive coverage see [15].)

Given two input values, E = (the difference between the current temperature and the target temperature) and dE = (the time derivative of the difference), we are to determine output value, W = (the amount of heat or cooling source). We select *fuzzy variables*, such as, NB = Negative Big, NS = Negative Small, ZO= Zero, PS = Positive Small, and PB = Positive Big. Membership functions for the fuzzy variables, as functions of input and output values, are then defined (typically as a set of triangles or trapezoids in a graph representation). For

example, the membership function value or degree of variable PS is 1 and other variables, NB, NS, and so on, are 0, when E = 0.5. Generally, other membership functions can be defined and selections can affect the control performance. Fuzzy if-then rules that derive W from E and dE in terms of the fuzzy variables are given as the following table. For simplicity, assume the empty entries do not occur.

		dE				
		NB	NS	ZO	PS	PB
E	NB			PB		
	NS			PS		
	ZO	PB	PS	ZO	NS	NB
	PS			NS		
	PB			NB		

This table represents nine rules corresponding to the nine entries in the table. For example, "if E = ZO and dE = NB, then W = PB" may be called Rule 1. The remaining four entries in the same horizontal line of the table may be called Rules 2, 3, 4 and 5. The remaining four entries in the vertical line may be called Rules 6, 7, 8 and 9.

Now with all the predetermined information, we can compute W for given values of E and dE. Suppose that E = 0.75 and dE = 0. From the membership function defined earlier, E can be PB with degree = 0.5 and PS with degree = 0.5, dE is ZO with degree = 1.0. Hence, in the if-then table, two rules are applicable, Rules 8 and 9. Using each of these two rules we compute a membership function for W as follows, where $\wedge$ takes the minimum of the operand membership functions. The weight (firing strength) of each rule is determined as:

$$\alpha_8 = m_{PB}(E) \wedge m_{ZO}(dE) = 0.5 \wedge 1.0 = 0.5$$

$$\alpha_9 = m_{PS}(E) \wedge m_{ZO}(dE) = 0.5 \wedge 1.0 = 0.5$$

Then the membership function associated with each rule is determined as:

$$m_8(W) = \alpha_8 \wedge m_{NB}(W) = 0.5 \wedge m_{NB}(W)$$

$$m_9(W) = \alpha_9 \wedge m_{NS}(W) = 0.5 \wedge m_{NS}(W)$$

The membership function for W, m(W), is obtained as the max of the above two intermediate membership functions, $m_8(W)$ and $m_9(W)$. This m(W) gives the fuzzy version of the solution for W, but we need a specific single value W_0 as a system output to perform control. For this purpose, we compute the center of gravity of m(W) as W_0, which is called a difuzzification procedure.

4 ROUGH CONTROL TAXONOMY

To develop various potential types of rough control in an organized fashion, classification of such types would be helpful. Generally there are different ways of classification and the following is just one possibility. In the course of developing actual applications, a newer and more appropriate classification may emerge in the future.

1. Pure (rather than hybrid) Rough Control
   (a) Under an assumption of existing control rules, output values are determined for imprecise or incomplete input and parameter values and/or rules.
   (b) Deriving feasible control rules when the input-output relations are vague.
2. Hybrid systems
   (a) Rough + fuzzy systems.
   (b) Fuzzy + rough systems.
   (c) Rough + neural network systems.
   (d) Neural network + rough systems.
   (e) Rough + other systems (e.g., PID, chaos).

5 PRINCIPLES OF ROUGH CONTROL

1. Pure (rather than hybrid) Rough Control

(a) Under an assumption of existing control rules, output values are determined for imprecise or incomplete input and parameter values and/or rules. The basic concept here is to explore the most typical feature of rough set theory. Inputs and parameters correspond to condition attributes in standard rough set theory; outputs are decision or action attributes.

Typically, values of decision attributes in rough sets, i.e., outputs in rough control, are given in descriptive expression such as "slow." For control problems, we need to derive numeric values. Possible methods include the following.

Methods of deriving numeric output values:

i. Making the control rules fine enough to produce numeric values.

ii. Use of rough measures (e.g., dependency), possibly in conjunction with a simple arithmetic formula.

iii. As an extension of (2), a technique similar to defuzzification process discussed above for fuzzy control may be employed. The idea is to find "the center of gravity" or weighted mean of several possible output values. For example, "rough variables" in place of fuzzy variables may be defined. Their weights (firing strengths) may be evaluated based on their dependencies.

This type of rough control has high potentials when we look at proven successes of fuzzy control, the key elements contributed to those successes, and the similarities of the key elements in fuzzy and rough control. As stated earlier, the key elements for fuzzy control successes are the use of descriptive expressions and uncertain reasoning. Rough control also has these characteristics.

(b) In this approach, we derive feasible control rules when the input-output relations are vague. The basic principle again, is to tailor a typical application of rough sets for discovering relationships in data to particularly fit control problems. Inputs and parameters correspond to condition attributes in standard rough set theory as before, and outputs are decision or action attributes. A set of existing control rules, whether it is described by human experts or developed for PID or fuzzy control, may be incomplete, imprecise, or contain redundancy. By employing rough set theory, nonessential rules may be identified then deleted, or less important rules may be downgraded in priority or weighted by smaller factors.

2. Hybrid systems

This category is also one of the most promising for practical applications. The fundamental concept is to complement each other's weakness, thus

creating new approaches to solve problems. For example, fuzzy control has many established application cases while rough control has relatively few. Integrating rough control with successful fuzzy control cases could be relatively easy for accomplishing real world practicality of rough control. Fuzzy sets allow partial membership to deal with gradual changes or uncertainties, while rough sets allow multiple memberships to deal with indiscernibility. A fuzzy-rough hybrid system may allow multiple-partial membership (e.g., multiple membership where each can be partial) to deal with both indiscernibility and uncertainty. In rough + fuzzy systems, for example, the macroscopic, possibly symbolic, output is determined by rough control while fine tuning is carried out by fuzzy control. In fuzzy + rough systems, the roles will be reverse. Or, rough control lacks capabilities of pattern recognition or memory. A hybrid system of rough control and neural networks may work well for certain applications.

6 CASE STUDY

In the previous two sections, potentials of rough control are stated in an abstract manner. In order to relate these statements to real world applications, we will consider a fictitious control example. We note that although we use one specific hypothetical example for easy understanding, the basic idea can be applied to many other types of control problems.

Imagine we want to perform delicate room temperature control for a sophisticated experiment, perhaps for biomedical or solid state physics. In this scenario, we need fine temperature control: the allowable temperature deviation range is, say, within ± 0.02 $^{\circ}C$ of the target temperature throughout the room. Furthermore, the homogeneity of the temperature distribution is required, i.e., the temperature difference between any two points of one meter distance must be less than, say, 0.01 $^{\circ}C$. The difficulty of temperature control is compounded because of the various boundary conditions. For example, the current level of robotics is not good enough to make robots perform the experiment. That is, human technicians must be in the room, which themselves are complicated heat sources.

Solving the problem theoretically, for example, by the Navier-Stokes' equation for air flow, associated with thermodynamic equations for heat conduction, convection, and radiation, under such complicated boundary conditions is out of question. A practical approach is to develop empirical formula for control

from experimental data for temperature distributions and various heat/cooling sources. Rough control may be used in various stages of such development.

The major component of the inputs (attributes) is the measured temperatures throughout the room. Since the temperatures have to be measured three dimensionally, many sensors will be required at least initially. Other factors, such as the human body heat source, can be added as a part of the inputs. These inputs can be denoted as s1, s2, ... The outputs can be heat/cooling sources, which can be denoted as, t1, t2, ... The problem here is to deal with incomplete and imprecise data. Even if we use many sensors, there are still many points in the room where the true temperatures are never measured. Also, in addition to the sensor reliability problem, local temperature fluctuations due to various causes such as convection, radiation, and small turbulence, will make measurements inaccurate.

An input-output mapping table may look as follows:

An input-output mapping table

(Input)	s1	s2	s3 ...	(Output)	t1	t2 ...
		...				
	+.03	+.01	-.04 ...		-.6	+.2 ...
	+.03	+.01	-.03 ...		-.6	+.3 ...
		...				

Such a table can be constructed initially in various ways. For example, if there are any existing methods to approximate the mapping, they can be used. Or, experiments can be conducted by human experts, possibly involving trial-and-errors.

Since maintaining many temperature sensors is expensive, lesser sensors are desirable. During the first stage, rough control may be used to reduce the number of sensors required to achieve the required control. For example, suppose that contributions of Sensors No. 4, 7, and 23, to the outputs are found to be insignificant, then they may be deleted. Similarly, some of output elements may be found insignificant and thus deleted. Or, rough control may suggest other possible ways for achieving the same results.

After the initial construction of the input-output mapping table, the system becomes operational. However, the operations probably require much fine tuning. For example, there will be a certain limit to the number of the table entries because of the space and efficiency. In other words, all the possible

combinations of input-output values may not be included in the table. Also, the data are incomplete and inaccurate. Rough control may be used for fine tuning of such a circumstance. For example, the closest table entries are used as "zeroth approximation." Rough control then finds "superposing corrections" to the zeroth approximation. Rough control may compute the corrections by first determining the input-to-output dependencies, then taking "the center of gravity" as in case of defuzzification. Generally, the use of traditional method for zeroth approximation and rough control for fine tuning may be conservative but probably safer than relying the total control on a new technology.

The above illustrates a basic idea of rough control. Many other variations and extensions for employing rough control would be possible, depending on the types of applications.

7 CONCLUSIONS

This article has presented a preliminary study on rough control potentials. The topics are arranged in a top-down approach starting from a global overview of the subject to somewhat detailed specifications of rough control.

Once rough control is proved to be feasible, its implication can be enormous. As stated earlier, it can be applied to control various physical, chemical, or other numeric characteristics, such as temperature, electric current, flow of liquid/gas, motion of machines, various business and financial quantities, etc. This means that controlling these characteristics in turn can be applied to many areas involving various engineering, scientific and management problems. Again, a list of successful application areas of fuzzy control [12] would be a good reference source to consider potential application areas for rough control. The list may include: transportation, consumer electronics, robotics, computers, communications, agriculture, medicine, management, finance, and education.

REFERENCES

[1] S. Khasnabis, T. Arciszewski, S.K. Hoda, and W. Ziarko, "Urban rail corridor control through machine learning: an IVHS approach," in B.H.V. Topping, (Ed.), Knowledge Based Systems for Civil and Structural Engineering, Aug. 1993, Civil-Comp Press, Edinburgh, UK, pp.97-104.

[2] W. Ziarko, J.D. Katzberg, "Rough sets approach to system modelling and control algorithm acquisition," IEEE WESCANEX 93. Communications, Computers and Power in the Modern Environment Conference Proceedings (Cat. No.93CH3317-5), May, 1993, pp. 154-64.

[3] W. Ziarko, "Generation of control algorithms for computerized controllers by operator supervised training," in Hamza, M.H. (Ed.), Proceedings of the Eleventh IASTED International Conference. Modelling, Identification and Control, Innsbruck, Austria, Feb., 1992, pp.510-13.

[4] R. Nowicki, R. Slowinski, and J. Stefanowski, "Evaluation of vibroacoustic diagnostic symptoms by means of the rough sets theory," Computers in Industry, Vol.20, No.2, 1992, pp.141-152.

[5] E. Czogala, A. Mr??zek, Z. Pawlak, "The Idea of a Rough Fuzzy Controller and its Application to the Stabilization of a Pendulum-Car System," Institute of Computer Science Reports, Warsaw University of Technology, February 1994.

[6] A. Mrozek and L. Plonka, "Rough sets for controller synthesis," Institute of Computer Science Reports, Warsaw University of Technology, October, 1994.

[7] RSSC'94: The Third International Workshop on Rough Sets and Soft Computing, San Jose, CA, Nov., 1994.

[8] T. Munakata (Guest Ed.), "Commercial and Industrial AI," Communications of the ACM, Vol. 37, No. 3, March, 1994, pp. 23-119.

[9] T. Munakata (Guest Ed.), "New Horizons of Commercial and Industrial AI," Communications of the ACM, Vol. 38, No. 11, Nov. 1995.

[10] Z. Pawlak, J. Grzymala-Busse, R. Slowinski and W. Ziarko, "Rough sets," Communications of the ACM, Vol. 38, No. 11, Nov. 1995.

[11] T. Munakata, "Commercial and Industrial AI and Future Perspective on Rough Sets," Proc. RSSC'94: The Third International Workshop on Rough Sets and Soft Computing, San Jose, CA, Nov., 1994.

[12] T. Munakata and Y. Jani, "Fuzzy Systems: An Overview," Communications of the ACM, Vol. 37, No. 3, March, 1994, pp. 69-76.

[13] W. Ditto and T. Munakata, "Chaos Systems: Principles and Applications," Communications of the ACM, forthcoming.

[14] B. Widrow, D.E. Rumelhart, and M.A. Lehr, "Neural Networks: Applications in Industry, Business and Science," Communications of the ACM, Vol. 37, No. 3, March, 1994, pp. 93-105.

[15] Lee, C.C. Fuzzy logic in control systems: fuzzy logic controller, Parts I and II. IEEE Trans. Sys. Man. Cyb., 20, 2 (March/April, 1990) 404-435.

PART II

APPLICATIONS

5

MACHINE LEARNING & KNOWLEDGE ACQUISITION, ROUGH SETS, AND THE ENGLISH SEMANTIC CODE

Jerzy W. Grzymala-Busse, Sally Yeates Sedelow*, and Walter A. Sedelow, Jr.*

Department of Electrical Engineering and Computer Science,
University of Kansas,
Lawrence, KS 66045
USA

** Department of Computer and Information Science,*
University of Arkansas at Little Rock,
Little Rock, AR 72204
USA

ABSTRACT

Rough Set-based machine learning and knowledge acquisition, as embodied in the system LERS, are applied to the task of sorting out natural-language word sense relationships. The data for training and testing are derived from *The Oxford English Dictionary*; a subsequent objective of this research enterprise is automatically placing additional terms in *Roget's International Thesaurus*. The results of this research are promising, so that now we would like to employ this approach to providing a comprehensive whole-language base for general use in building varied natural-language computing applications.

1 INTRODUCTION

Computer-based natural language systems of all sorts—interfaces, message analyzers, question answerers, etc.—have been, and continue to be, limited by a

system's ability to deal with natural language semantics for the English language as a whole. Research by the Sedelows and their associates has concentrated on associational semantics—which is meant to indicate that word senses are heavily influenced through association with other semantically-similar or at least semantically- related words. The Sedelow research has drawn heavily upon the Bryan graph-theoretic model of abstract thesauri [1, 2] as applied to a conceptual thesaurus *Roget's International Thesaurus* [11] to study the effectiveness of such a thesaurus for the bete noire of all semantic systems: ambiguity. Using the Bryan model it has been shown possible to disambiguate among word senses without human intervention, solely by algorithm. The lattice representations employed in Formal Concept Analysis [14, 15] also have proved to be excellent for documenting precise semantic inter- relationships among word senses.

Given these encouraging research results involving the use of a thesaurus for fine-grained semantic distinctions, a next major goal is to enhance the coverage of a whole-language thesaurus such as *Roget's International Thesaurus* by automatically adding to it terms it does not currently contain. The computational availability of that most major of dictionaries for English, *The Oxford English Dictionary (OED)* [12], has prompted the experiment reported on in this paper. Here, the first task was to explore the use of Machine Learning as embodied in the LERS system to classify terms *within* the OED. Our objective was to see whether the philologically-analytic data for each word in the OED is sufficient for good internal classification, where by internal classification we mean the discriminant association of a word sense with the more semantically appropriate among two or more senses of other words. Success there would then prompt exploration of mapping OED terms into the *Thesaurus.*

The Rough Set-based system LERS first requires a decision table consisting of examples, their attributes, values for those attributes, and a classification assignment (decision) provided by an expert. The four words we chose to use as training examples were 1. concept; 2. conception; 3. conceptual; and 4. imagination. The decision table for these words provided the input to, first, a machine-learning rule- generating option of LERS. Those rules were, then, the basis for the rules LERS would generate for the words 1. conceit; and 2. image. This comparison, using Rough Sets, resulted in classifying senses of "conceit" and "image" as to whether they were more closely related, semantically, to "concept", "conception", and "conceptual"—all treated as instances of "concept"—or to "image". As any user of the English language knows, this classification is not a straightforward task; all the words are inter-related as to meanings and thus conceptually/semantically tangled. Hence, we have here a strong test.

Since a basic purpose of this experiment was to see how useful the OED data might be for automating this particular task, for Attribute categories we turned

to the "General Explanations" section in the OED. From that section we selected as Attributes the following philological characterizations: Part of Speech, Origin, Status, Word Type, Citizenship, Domain Specification, and Form History. In addition, we added as Attributes the following: Hypernym (a term up the abstraction ladder from, e.g., "concept"), Hyponym (a term down the abstraction ladder); Meronym (part-whole relationship); and Synonym. As an example, the first sense of Concept I in our OED [12] would have the following values for these attributes: Part of Speech - Substantive (all words in the OED for which Part of Speech is not specified are, by default, substantives); Origin - Latin; Status - Obsolete; Word Type - Single; Citizenship - Natural (a somewhat obscure classification which includes Naturals, Denizens, Aliens, and Casuals, etc. Naturals include all native words like "father", and all fully naturalized words like "street", "rose". Denizens are words such as "aide-de-camp" which are fully naturalized as to use but not as to form. Aliens are names of foreign objects for which we have no native equivalent, e.g., shah, and Casuals are also foreign words, but are not in habitual use.); Domain Specification - Concept I has no value for this attribute (the second sense of "concept", Concept II has two values for this attribute: Logic and Philosophy); Form History - adoption (again, a complicated attribute; adoption means the word comes from a form in another language, in this case, Latin); Hypernym - "conceit"; Hyponym - Concept I has no value for this attribute; Meronym - Concept I has no value for this attribute; Synonyms - idea, throught, disposition, imagination, opinion, fancy. Finally, the expert's Classification for Concept I is that it belongs to the group (we called it a semicolon group because of the way *Roget's* is subdivided) Concept.
It should be emphasized that when the data in the decision table is entered, no attribute may have more than one value. So that in fact the first two entries, both for the sense of concept we have labeled Concept I, are as follows:

Concept-1 Substantive Latin Obsolete Single Natural ? Adoption Conceit ? ? Idea concept.

Concept-2 Substantive Latin Obsolete Single Natural ? Adoption Conceit ? ? Thought concept.

Notice that these entries differ only in a single attribute value- – through the appearance of the synonym "idea" in Concept-1 and the synonym "thought" in Concept-2. Question marks stand for an attribute for which there is no value. In the OED, the entry for "concept" begins with etymological information indicating that the word's origin is chiefly Latin. This information applies to all senses of "concept" under the boldface dictionary heading: **Concept**. The expert extracted this information; to do so automatically would require the use

of an already parsed version of the OED on CD-ROM or calling a parsing algorithm for each word being considered. Following the etymological information for all the senses listed under the boldfaced heading, the definition for the first sense of "concept" appears. It begins with a symbol indicating that this sense is obsolete (Status). It then reads:

= CONCEIT, in various senses: a. A thought, idea: = CONCEIT sb. I. b. Disposition, frame of mind; ibid. 2 c. c. Imagination, fancy; ibid. 7. d. Opinion; ibid. 4. Obs.

There is no indication of part-of-speech, so the default value is Substantive; its origin has been established as Latin; the status is Obsolete; the word is Single (no hyphens or spaces in the headword); it is Natural (no indication of anything else; default is Natural); there is no specification of domain; the form history is Adoption from Latin (see etymology for this information). The next four attribute values required the expert. As a hypernym Conceit was chosen because the definition specifies that this sense of Concept "= CONCEIT, in various senses". Although the equal sign here might suggest synonymy, "various senses" suggests a partitioning or subdivision; hence, the assignment of Conceit as hypernym. The partitions, themselves, would seem to be in parallel, so the terms "thought", "idea," "disposition", "frame of mind", "imagination", "fancy", and "opinion" were all classified as synonyms of "concept". And finally, the value of the decision was "concept", itself. No values were chosen for the attributes hyponym and meronym.

For this experiment, the attributes Hypernym, Hyponym, and Meronym were included as is conventional in traditional semantic taxonomies [6] as well as in computer-accessible data bases such as WordNet [7]. For some empirical investigations using the Thesaurus we have found the equivalent of a Hypernym useful as a way of characterizing a semantic partition. That is, we could use a word in the explicit hierarchy in the *Thesaurus* as a name for a semantic partition which would discriminate among, for example, word senses. WordNet has been constructed on the premise that such taxonomic relationships will prove useful. Hence, for compatibility and potential utility we used such attributes. However, for the specific goal of adding terms to a lexicon—in our case, principally to a *Thesaurus*—it may be that synonyms will be sufficient. If so, that would obviate the need for the more fine-grained discriminations implied by the taxonomic attributes and would make parsing the definition algorithmically even more straightforward.

Perhaps a word of explanation as to the possible sufficiency of the Synonym attribute for this specific goal is in order. As indicated above, the *Thesaurus* has an explicit hierarchy. The uppermost level in the hierarchy is divided into eight classes. Each class is divided into several labeled sub-classes indicated

by Roman numerals, and each sub- class is divided into labeled sub-sub-classes designated by capital letters. Each sub-sub-class is divided into several of 1032 categories, which are numbered consecutively throughout the text and which are the lowest level in the hierarchy to be labeled with a word. Below this level are groupings of paragraphs, grouped by part of speech; next are paragraphs which are numbered; at the bottom level are what we have called semicolon groups (at least one boundary of such a group is marked by a semicolon) which contain those words which are most closely associated semantically. For example, here is one occurrence of the word "concept" in the Thesaurus:

Class Six: Intellect
I. Intellectual Faculties and Processes
Faculties
C. Functions of Mind
478. Idea
Nouns
478.1
478.1.1 idea, idee [dial.],
Idee [G.], thought, think [slang],
notion, fancy, concept, conception,
conceit, percept, perception,
impression, mental impression;

As is evident, a number of words occurring in the Concept I sense in the OED occur in semicolon group 478.1.1 [the *Thesaurus* does not provide numbering for the semicolon groups; for clarity we have added that in our data base and here]. Notice that "conceit" is here in synonymous relationship with "concept", "idea", "fancy", etc. This example suggests a reason for our speculation that the Synonym attribute may be sufficient for this specific task.
To return to the input Decision Table for the experiment, there were twelve examples based on the entries in the OED for "Concept"; fifty-one for "Conception"; two for "Conceptual"; and fourteen for "Imagination". (As an aside, it might be noted that the OED's greater emphasis upon "Conception", vis-a-vis "Concept" correlates very well with the *Thesaurus*. A recent article [13] using a Formal Concept Analysis approach, notes that although "concept", was the initiating term, the resulting lattice had "conception" as the Supremum, with "concept" and other senses of "conception" (e.g., "coming with child") below.) The input Decision Table described above was then used in classification runs with the Test Data, which consisted of twenty-one entries for "conceit", and

twenty-four for "image". It should be noted that in the Decision Table for the Test Data, the expert's "decision" was always "concept", (the classification covering "concept", "conceptual", and "conception") and never "image" (the classification covering "imagination" in the input control data)—thus creating more of a challenge for system LERS.

2 INDUCTION SYSTEM LERS

System LERS (Learning from Examples based on Rough Sets) has four options of rule induction [4]. Two of these options, called LEM1 and LEM2, entail machine learning. Using the LEM1 or the LEM2 option, the system induces a set of rules sufficient to describe all examples. The description is discriminant. Two other options, called All Global Coverings and All Rules, respectively, are for knowledge acquisition. When LERS is used in either of these options, it induces, in general, much larger rule sets. When using these bigger rule sets it is possible to tell to which concept an example belongs even when the given information is incomplete, i.e., when values of only some attributes are available. Other systems that also use the rough set approach to rule induction were presented in [10] and [16]. Rough set theory was introduced by Z. Pawlak in 1982, see [8, 9].

In the experiments on data derived from the OED, two LERS options were used: LEM2 and All Rules. LERS in the option LEM2 induces the simplest rules taking into account the users priorities. By contrast, LERS in the All Rules option induces all rules that can be induced from input data. In both cases, any induced rule is as simple as possible.

During classification of unseen (testing) examples LERS uses a new scheme similar to the bucket brigade algorithm of generic algorithms. In this scheme, for every example first an attempt is made for complete matching by induced rules. If the attempt is successful—i.e., there exists at least one rule that matches the example—the concept to which the example belongs is decided by voting, in which all rules supporting the same concept vote for the concept with their strength and specificity. The strength of a rule is the number of times the rule was successful for training data in the correct classification of examples. The specificity of a rule is the number of conditions in the rule. When complete matching is impossible, a partial matching is attempted. Here rules that partially match the example vote, each rule votes with the product of three numbers: strength, specificity, and the ratio of the number of matching conditions of the rule to the total number of conditions of the rule. This scheme was successfully used in medicine [5].

3 EXPERIMENTS

Both the training and testing data sets that were constructed from the OED are presented in the Appendix under names oedthes.tab and test.tab, respectively. Both sets contain missing attribute values, denoted by question marks. In our experiments the LERS system induced rules treating ? as a special value. Thus ?s were included in rule conditions.

In the first experiment, the quality of rule sets for two options, LEM2 and All Rules, was estimated by running a statistical test *leaving-one-out* for oedthes.tab data set. In this test rules are iteratively induced from all examples excluding one that is used for testing. The number of iterations is equal to the total number of examples. Thus every example is used—once—for testing. In the option LEM2 a total of 37 errors (incorrectly recognized examples) out of 81 examples was registered. Option All Rules produced only two errors during leaving- one-out (out of 81 examples).

LERS with the LEM2 option induced 13 rules; two of these rules contained missing attribute values. In this induction, the attributes: hypernym, hyponym, and meronomy were given higher priorities. To the attribute synonyms was assigned the highest priority. Moreover, LERS with the All Rules option induced 71 rules, among these, 23 rules were involved with missing values.

For the following experiments both rule sets, induced by LERS using LEM2 and All Rules options, were modified by removing all rules with conditions containing missing values. The obvious reason for doing so is that rules relating missing values should not participate in matching missing values of examples; rather, the classification should be exclusively based on matching actual attribute-value pairs. The modified rule set induced by the LEM2 option of LERS was called *mod-lem2.rul*, the modified rule set induced by All Rules option of LERS was called *mod-all.rul*.

Selected rules from the set *mod-all.rul*, with the strengths equal to or greater than six, are presented below:

(synonyms, Imagination) → (semicolon-group, concept) with strength = 9,
(synonyms, Apprehension) → (semicolon-group, concept) with strength = 8,
(meronomy, Mind) → (semicolon-group, image) with strength = 7,
(synonyms, Notion) → (semicolon-group, concept) with strength = 6,
(hypernym, Conceit) → (semicolon-group, concept) with strength = 6,
(hyponym, Concept) → (semicolon-group, image) with strength = 6.

Note that all of these rules are in the set *mod-all.rul*, however, *mod-all.rul* additionally contains one rule in this range (with strength equal to or greater than six):

(meronomy, Origination-in-the-Mind) → (semicolon-group, concept)
with strength = 6.

Two different experiments were done with rule sets *mod- lem2.rul* and *mod-all.rul*. First, both rule sets were tested on the original, training data set oedthes.tab, from which both rule sets were induced. This was done to check how modification affected the quality of rule sets. It turned out that the *mod-lem2.rul* rule set misclassified 34 examples (out of 81). On the other hand, the rule set *mod-all.rul* was still able to classify correctly all 81 examples.
The second experiment was running rule sets *mod-lem2.rul* and *mod-all.rul* against a testing data set, test.tab, containing 45 examples. The rule set *mod-lem2.rul* was able to recognize correctly only the following two examples: the third example, because this example was matched by the following rule:

(synonyms, Notion) → (semicolon-group, concept) with strength = 6;

and the eighth example, because this example was matched by the following rule:

(synonyms, Apprehension) → (semicolon-group, concept) with strength = 8

Rule set *mod-all.rul* correctly recognized five examples. This rule set correctly recognized the above two examples (for the same reasons) and, additionally, the following examples: the twentieth example, because among the two rules that match this example:

(hypernym, Production) → (semicolon-group, concept) with strength = 4,
(status, Current) & (synonyms, Fancy) → (semicolon-group, image)
with strength = 2,

the first rule is stronger and won in the voting; the twenty-first example, because this example was matched by the following rule:

(hypernym, Production) → (semicolon-group, concept) with strength = 4;

and the thirty-ninth example, because this example was matched by the following rule:

(synonyms, Plan) → (semicolon-group, concept) with strength = 3.

In all of the above cases, correct classification was made on the basis of com-

plete matching.

4 CONCLUSIONS

First, it is obvious that in this specific application of LERS, the option All Rules induced a much better rule set than the option LEM2. Not only the leaving-one-out test, but also testing rule sets against the training data and testing data all produced better results in the case of the All Rules option. This observation supports a significant claim by Grzymala-Busse and Grzymala-Busse [3] that rule sets induced by the knowledge acquisition options of LERS are of higher quality than rule sets induced by the machine learning options of LERS. Secondly, the rule set *mod-lem2.rul* incorrectly classified four examples (all of them during complete matching), and in the remaining 39 cases it could not classify examples from test.tab at all. By contrast, the much more complex rule set *mod-all.rul* was unable to classify at all only two examples, but it classified incorrectly 38 examples, eight during complete matching and 30 during partial matching.
Thirdly, in the natural language domain much bigger training data sets should be used for proper rule induction. In our study the training set was definitely too small. Further attempts should be made to induce rules from more representative data sets. One function of our work is to show that further research in this direction is promising.

REFERENCES

[1] Bryan, R. Abstract Thesauri and Graph Theory Applications to Thesaurus Research. In *Automated Language Analysis*, 1972–1973. Sedelow, S. (ed.), University of Kansas. Departments of Computer Science and Linguistics, Lawrence, Kansas, 45–89, 1973.

[2] Bryan, R. Modelling in Thesaurus Research. In *Automated Language Analysis*, 1973–1974. Sedelow, S. (ed.), University of Kansas Departments of Computer Science and Linguistics, Lawrence, Kansas, 44– 59, 1974.

[3] Grzymala-Busse, D. M., and Grzymala-Busse, J. W. Evaluation of machine learning approach to knowledge acquisition. *Proc. of the 14th Int. Avignon Conf., Paris*, 183–192, 1994.

[4] Grzymala-Busse, J. W. LERS—A system for learning from examples based on rough sets. In R. Slowinski (ed.), *Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory.* Kluwer Academic Publishers, Dordrecht, Boston, London, 3–18, 1992.

[5] Grzymala-Busse, J. W. and Woolery, L. Machine learning for an expert system to predict preterm birth risk. *Journal of the American Medical Informatics Association* 1, 439–446, 1994.

[6] Lyons, J. *Semantics.* Cambridge University Press, Cambridge, London, New York, Melbourne, 270–317, 1977.

[7] Miller, G. et al. *Introduction to WordNet, etc.* The EURALEX Bulletin, Vol. 3, No. 4, Winter, entire issue, 1990.

[8] Pawlak, Z. Rough sets. Int. *J. Comp. and Inf. Sciences* 11, 341–356, 1982.

[9] Pawlak, Z. *Rough Sets. Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.

[10] Slowinski, R. and Stefanowski, J. 'RoughDAS' and 'RoughClass' software implementations of the rough set approach. In *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.* Slowinski, R. (ed.), Kluwer Academic Publishers, Dordrecht, Boston, London, 445–456, 1992.

[11] *Roget's International Thesaurus.* Third Edition. Thomas Y. Crowell, New York, 1962.

[12] *The Oxford English Dictionary.* Murray, J. A. H., Bradley, H., Craigie, W. A., Onions, C. T. (eds.), Oxford University Press, Oxford, New York, 1933.

[13] Sedelow, S. Y. and Sedelow, W. A., Jr. Thesauri and Concept-Lattice Semantic Nets. In *Knowledge Organization and Quality Management.* Albrechtsen, H., et al. (eds.), Indeks Verlag, Frankfurt/Main, 350–357, 1994.

[14] Wille, R. Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In *Ordered Sets.* Rival, I. (ed.). Reidel, Dordrecht, Boston, 445–470, 1982.

[15] Wille, R. *ANACONDA and TOSCANA programs.* Darmstadt, 1995.

[16] Ziarko, W. P. Acquisition of control algorithms from operation data. In *Intelligent Decision Support. Handbook of Applications and Advances of*

the Rough Sets Theory. Slowinski, R. (ed.), Kluwer Academic Publishers, Dordrecht, Boston, London, 61–75, 1992.

APPENDIX A

TRAINING DATA SET OEDTHES.TAB (IN LERS FORMAT)

[name part-of-speech origin status word-type citizenship domain- specification form-history hypernym hyponym meronomy synonyms semicolon-group]
concept-1 Substantive Latin Obsolete Single Natural ? Adoption Conceit ? ? Idea concept
concept-2 Substantive Latin Obsolete Single Natural ? Adoption Conceit ? ? Thought concept
concept-3 Substantive Latin Obsolete Single Natural ? Adoption Conceit ? ? Imagination concept
concept-4 Substantive Latin Obsolete Single Natural ? Adoption Conceit ? ? Opinion concept
concept-5 Substantive Latin Obsolete Single Natural ? Adoption Conceit ? ? Fancy concept
concept-6 Substantive Latin Obsolete Single Natural ? Adoption Conceit ? ? Disposition concept
concept-7 Substantive Latin Current Single Natural Logic Adoption Idea Class-of-Objects ? Notion concept
concept-8 Substantive Latin Current Single Natural Logic Adoption Idea Class-of-Objects ? Idea concept
concept-9 Substantive Latin Current Single Natural Philosophy Adoption Idea Class-of-Objects ? Idea concept
concept-10 Substantive Latin Current Single Natural Philosophy Adoption Idea Class-of-Objects ? Notion concept
concept-11 Verb-Transitive Latin Obsolete Single Natural ? Adoption Conceive ? ? ? concept
concept-12 Verb-Transitive Latin Rare Single Natural ? Adoption Conceive ? ? ? concept
conception-1 Substantive Latin Current Single Natural ? Adoption Action Conceiving-in-the-Womb ? ? concept

conception-2 Substantive Latin Current Single Natural ? Adoption Fact Being-Conceived-in-the- Womb ? ? concept
conception-3 Substantive Latin Obsolete Single Natural Plants Adoption Generation ? ? ? concept
conception-4 Substantive Latin Obsolete Single Transferred Plants Adoption Generation ? ? ? concept
conception-5 Substantive Latin Obsolete Single Natural Plants Adoption Production ? ? ? concept
conception-6 Substantive Latin Obsolete Single Transferred Plants Adoption Production ? ? ? concept
conception-7 Substantive Latin Obsolete Single Natural Minerals Adoption Generation ? ? ? concept
conception-8 Substantive Latin Obsolete Single Transferred Minerals Adoption Generation ? ? ? concept
conception-9 Substantive Latin Obsolete Single Natural Minerals Adoption Production ? ? ? concept
conception-10 Substantive Latin Obsolete Single Transferred Minerals Adoption Production ? ? ? concept
conception-11 Substantive Latin Current Single Natural ? Adoption Embryo ? ? ? concept
conception-12 Substantive Latin Current Single Natural ? Adoption Foetus ? ? ? concept
conception-13 Substantive Latin Current Single Concretely ? Adoption Foetus ? ? ? concept
conception-14 Substantive Latin Current Single Concretely ? Adoption Embryo ? ? ? concept
conception-15 Substantive Latin Current Single Natural ? Adoption Action ? ? Apprehension concept
conception-16 Substantive Latin Current Single Natural ? Adoption Faculty ? ? Apprehension concept
conception-17 Substantive Latin Current Single Natural ? Adoption Forming-An-Idea ? ? Apprehension concept
conception-18 Substantive Latin Current Single Natural ? Adoption Forming-A-Notion ? ? Apprehension concept
conception-19 Substantive Latin Current Single Natural ? Adoption Action ? ? Imagination concept
conception-20 Substantive Latin Current Single Natural ? Adoption Faculty ? ? Imagination concept
conception-21 Substantive Latin Current Single Natural ? Adoption Forming-An-Idea ? ? Imagination concept
conception-22 Substantive Latin Current Single Natural ? Adoption Forming-A-Notion ? ? Imagination concept

conception-23 Substantive Latin Current Single Natural Philosophy Adoption Action ? ? Apprehension concept
conception-24 Substantive Latin Current Single Natural Philosophy Adoption Faculty ? ? Apprehension concept
conception-25 Substantive Latin Current Single Natural Philosophy Adoption Forming-An-Idea ? ? Apprehension concept
conception-26 Substantive Latin Current Single Natural Philosophy Adoption Forming-A-Notion ? ? Apprehension concept
conception-27 Substantive Latin Current Single Natural Philosophy Adoption Action ? ? Imagination concept
conception-28 Substantive Latin Current Single Natural Philosophy Adoption Faculty ? ? Imagination concept
conception-29 Substantive Latin Current Single Natural Philosophy Adoption Forming-An-Idea ? ? Imagination concept
conception-30 Substantive Latin Current Single Natural Philosophy Adoption Forming-A-Notion ? ? Imagination concept
conception-31 Substantive Latin Current Single Natural Philosophy Adoption Forming ? ? Notion concept
conception-32 Substantive Latin Current Single Natural Philosophy Adoption Faculty ? ? Notion concept
conception-33 Substantive Latin Current Single Natural ? Adoption ? ? A-Part-Of-Mind Notion concept
conception-34 Substantive Latin Current Single Natural ? Adoption ? ? A-Part-Of-Mind Idea concept
conception-35 Substantive Latin Current Single Natural Philosophy Adoption ? ? A-Part-Of-Mind Notion concept
conception-36 Substantive Latin Current Single Natural Philosophy Adoption ? ? A-Part-Of-Mind Idea concept
conception-37 Substantive Latin Current Single Natural Philosophy Adoption General-Notion ? ? ? concept
conception-38 Substantive Latin Current Single Natural Philosophy Adoption Concept ? ? ? concept
conception-39 Substantive Latin Current Single Natural ? Adoption ? ? Origination-in-the-Mind Designing concept
conception-40 Substantive Latin Current Single Natural ? Adoption ? ? Origination-in-the-Mind Planning concept
conception-41 Substantive Latin Current Single Natural ? Adoption ? ? A-Part-of-the-Mind Mental- Product concept
conception-42 Substantive Latin Current Single Natural ? Adoption ? ? A-Part-of-the-Mind Design concept
conception-43 Substantive Latin Current Single Natural ? Adoption ? ? A-Part-of-the-Mind Plan concept

conception-44 Substantive Latin Current Single Natural ? Adoption ? ? A-Part-of-the-Mind Original-Idea concept
conception-45 Substantive Latin Current Single Natural ? Adoption ? ? Mental-Product-of- Inventive-Faculty Mental-Product concept
conception-46 Substantive Latin Current Single Natural ? Adoption ? ? Mental-Product-of- Inventive-Faculty Design concept
conception-47 Substantive Latin Current Single Natural ? Adoption ? ? Mental-Product-of- Inventive-Faculty Plan concept
conception-48 Substantive Latin Current Single Natural ? Adoption ? ? Mental-Product-of- Inventive-Faculty Original-Idea concept
conception-49 Substantive Latin Current Single Natural ? Adoption ? ? Origination-in-the-Mind Mental-Product concept
conception-50 Substantive Latin Current Single Natural ? Adoption ? ? Origination-in-the-Mind Design concept
conception-51 Substantive Latin Current Single Natural ? Adoption ? ? Origination-in-the-Mind Plan concept
conception-52 Substantive Latin Current Single Natural ? Adoption ? ? Origination-in-the-Mind Original-Idea concept
conceptual-1 Adjective Latin Current Single Natural ? Adoption ? ? A-Part-of-Concept ? concept
conceptual-2 Adjective Latin Current Single Natural ? Adoption ? ? A-Part-of-Mental-Conception ? concept
Imagination-1 Substantive Latin Current Single Natural ? Adoption Action Concept ? Mental-Image image
Imagination-2 Substantive Latin Current Single Natural ? Adoption Action Concept ? Idea image
Imagination-3 Substantive Latin Current Single Natural ? Adoption Forming Concept ? Mental- Image image
Imagination-4 Substantive Latin Current Single Natural ? Adoption Forming Concept ? Idea image
Imagination-5 Substantive Latin Current Single Natural ? Adoption Faculty Image Mind ? image
Imagination-6 Substantive Latin Current Single Natural ? Adoption Faculty Concept Mind ? image
Imagination-7 Substantive Latin Current Single Natural ? Adoption Power Concept ? ? image
Imagination-8 Substantive Latin Current Single Natural ? Adoption Operation Fantastic-Thought ? Fancy image
Imagination-9 Substantive Latin Current Single Natural ? Adoption Creative-Faculty ? Mind Poetic-Genius image
Imagination-10 Substantive Latin Current Single Natural ? Adoption Power Conception ? Poetic- Genius image

Imagination-11 Substantive Latin Current Single Natural ? Adoption Mind ? Mind ? image
Imagination-12 Substantive Latin Current Single Natural ? Adoption Operation ? Mind Thinking image
Imagination-13 Substantive Latin Current Single Natural ? Adoption Operation ? Mind Thought image
Imagination-14 Substantive Latin Current Single Natural ? Adoption Operation ? Mind Opinion image
Imagination-15 Substantive Latin Current Single Natural ? Adoption Operation ? ? Opinion image

APPENDIX B

TESTING DATA SET TEST.TAB (IN LERS FORMAT)

[name part-of-speech origin status word-type citizenship domain- specification form-history hypernym hyponym meronomy synonyms semicolon-group]
conceit-1 Substantive Latin Current Single Natural ? Adoption Conception ? ? Conceiving concept
conceit-2 Substantive Latin Current Single Natural ? Adoption Conception ? ? Conception concept
conceit-3 Substantive Latin Current Single Natural ? Adoption Conception ? ? Notion concept
conceit-4 Substantive Latin Current Single Natural ? Adoption Conception ? ? Idea concept
conceit-5 Substantive Latin Current Single Natural ? Adoption Conception ? ? Thought concept
conceit-6 Substantive Latin Current Single Natural ? Adoption Conception ? ? Device concept
conceit-7 Substantive Latin Obsolete Single Natural ? Adoption Faculty ? ? Conception concept
conceit-8 Substantive Latin Obsolete Single Natural ? Adoption Faculty ? ? Apprehension concept
conceit-9 Substantive Latin Obsolete Single Natural ? Adoption Faculty ?

? Understanding concept
conceit-10 Substantive Latin Current Single Natural ? Adoption Opinion ? ? Overweening-Opinion concept
conceit-11 Substantive Latin Current Single Natural ? Adoption Opinion ? ? Overestimation concept
conceit-12 Substantive Latin Current Single Natural ? Adoption Opinion ? ? Vanity concept
conceit-13 Substantive Latin Current Single Natural ? Adoption Opinion ? ? Pride concept
conceit-14 Substantive Latin Current Single Natural ? Adoption Notion ? ? Fancy concept
conceit-15 Substantive Latin Current Single Natural ? Adoption Notion ? ? Whim concept
conceit-16 Substantive Latin Current Single Natural ? Adoption Opinion ? ? Fancy concept
conceit-17 Substantive Latin Current Single Natural ? Adoption Opinion ? ? Whim concept
conceit-18 Substantive Latin Current Single Natural ? Adoption Action ? ? Fancy concept
conceit-19 Substantive Latin Current Single Natural ? Adoption Action ? ? Whim concept
conceit-20 Substantive Latin Current Single Natural ? Adoption Production ? ? Fancy concept
conceit-21 Substantive Latin Current Single Natural ? Adoption Production ? ? Whim concept
image-1 Substantive Latin Current Single Natural ? Adoption Imitation Solid-Form ? Statue concept
image-2 Substantive Latin Current Single Natural ? Adoption Imitation Solid-Form ? Effigy concept
image-3 Substantive Latin Current Single Natural ? Adoption Imitation Solid-Form ? Sculptured- Figure concept
image-4 Substantive Latin Current Single Natural ? Adoption Representation Solid-Form ? Statue concept
image-5 Substantive Latin Current Single Natural ? Adoption Representation Solid-Form ? Effigy concept
image-6 Substantive Latin Current Single Natural ? Adoption Representation Solid-Form ? Sculptured-Figure concept
image-7 Substantive Latin Current Single Natural ? Adoption ? Painting-On-A-Surface ? Likeness concept
image-8 Substantive Latin Current Single Natural ? Adoption ? Painting-On-A-Surface ? Portrait concept
image-9 Substantive Latin Current Single Natural ? Adoption ? Painting-

On-A-Surface ? Picture concept
image-10 Substantive Latin Current Single Natural ? Adoption ? Painting-On-A-Surface ? Carving concept
image-11 Substantive Latin Current Single Natural ? Adoption Representation ? ? Idea concept
image-12 Substantive Latin Current Single Natural ? Adoption Representation ? ? Conception concept
image-13 Substantive Latin Current Single Natural ? Adoption Representation ? ? Mental-Picture concept
image-14 Substantive Latin Current Single Natural ? Adoption Representation ? ? Mental- Impression concept
image-15 Substantive Latin Current Single Natural ? Adoption Representation ? Mind ? concept
image-16 Verb-Transitive Latin Current Single Natural ? Adoption Form ? ? Conceived concept
image-17 Verb-Transitive Latin Current Single Natural ? Adoption Form ? ? Devise concept
image-18 Verb-Transitive Latin Current Single Natural ? Adoption Form ? ? Plan concept
image-19 Verb-Transitive Latin Current Single Natural ? Adoption ? ? Mind Imagine concept
image-20 Verb-Transitive Latin Current Single Natural ? Adoption ? ? Mind Picture concept
image-21 Verb-Transitive Latin Current Single Natural ? Adoption ? ? Mind Represent concept
image-22 Verb-Transitive Latin Current Single Natural ? Adoption ? ? Image Imagine concept
image-23 Verb-Transitive Latin Current Single Natural ? Adoption ? ? Image Picture concept
image-24 Verb-Transitive Latin Current Single Natural ? Adoption ? ? Image Represent concept

6

GENERATION OF MULTIPLE KNOWLEDGE FROM DATABASES BASED ON ROUGH SETS THEORY

Xiaohua Hu, Nick Cercone, Wojciech Ziarko

Dept. of Computer Science,
Univ. of Regina, Regina
Saskatchewan, S4S 0A2, Canada

ABSTRACT

In this paper we present a new approach to generate multiple knowledge using rough sets theory. The idea is to generate several knowledge bases instead of one knowledge base for the classification of new object, hoping that the combination of answers of multiple knowledge bases result in better performance. Multiple knowledge bases can be formulated precisely and in an unified way within the framework of rough sets theory. Our approach is based on the reducts and decision matrix of the rough set theory. Our method first eliminates the superfluous attributes from the databases, next, the minimal decision rules are obtained through decision matrices. Then a set of reducts which include all the indispensable attributes to the learning task are computed, finally, the minimal decision rules are grouped to the corresponding reducts to form different knowledge bases. We attempt to make a theoretical model by using rough sets theory to explain the generation of multiple knowledge. The distinctive feature of our method over other methods of generating multiple knowledge is that in our method, each knowledge base is as accurate and complete as possible and at the same time as different from the other knowledge bases as possible. The test result shows the higher classification accuracy produced by multiple knowledge bases than that produced by single knowledge base.

1 INTRODUCTION

Knowledge bases have been successfully applied in a lot of real-world applications where intelligent decisions have to be made. Usually, knowledge base can be represented as a set of decision rules. This kind of knowledge base can be

derived from human experts or a collected data. Most of the time the collected data is so huge that it is beyond the human experts' ability to analyze without automated analysis techniques. The analysis and extraction of useful information from a collected data has been a subject of active research in database mining.

Rough sets as a non-statistical methodology for data analysis was introduced by Pawlak [9]. It provides a powerful tool for data analysis and knowledge discovery from imprecise and ambiguous data. So far, the theory of rough sets has been successfully applied in many areas, such as knowledge acquisition, forecasting and predictive modeling, knowledge base system, and data mining [11, 14]. A number of algorithms and systems have been developed based on rough sets theory which may induce a set of decision rules from a given decision table, and may use induced decision rules to classify future examples. Most of them are attempting to find and select the best minimal set of decision rules that use only a minimal subset of attributes from the given data table. A minimal set of decision rules as a knowledge base represents the minimum knowledge necessary to support the classification of the given data.

A single knowledge base which utilizes a single minimal set of decision rules to classify future examples may lead to mistakes, because the minimal set of decision rules are more sensitive to noise and a small number of rules means that few alternatives exist when classifying unseen objects. Recently, in order to enhance the classification accuracy, the concept of multiple knowledge bases or redundant knowledge emerged. The idea is to generate several knowledge bases instead of one knowledge base for the classification of new objects, hoping that the combination of answers of multiple knowledge bases result in better performance. Many research results illustrated that such multiple rules, if appropriately combined during classification, can improve the classification accuracy [7, 8, 2, 1]. Gams [2] developed the inductive learning system GINESYS that generate multiple sets of decision rules. One set of rules consists of "main" rule and of several "confirmation" rule. Each instance is classified with one set of rules by combining the probability distribution returned by different rules. Although the combination rule used by Gams is rather ad-hoc, the reported results are encouraging. In the learning system YAILS [12], redundancy is used to deal with several types of uncertainty existing in real domains to achieve higher accuracy. YAILS uses a simple mechanism to control redundancy. This mechanism consists on splitting the learned rules into two sets by a user-definable parameter (minimal utility, which acts as a way of controlling redundancy) : *foreground* rules and *background* rules. All these methods lack a theoretical formalism about the generation of redundant knowledge. The focus of this paper is to make a theoretical model to explain the generation of

multiple knowledge bases (or redundant knowledge). The distinctive feature of our methods over other methods of generating multiple knowledges is that each knowledge base is as accurate and complete as possible and at the same time as different from the other knowledge bases as possible.

2 RELATED NOTATIONS OF ROUGH SETS

2.1 Knowledge Representation System

The basic component of a knowledge representation system (KRS) is a set of objects. A database is a special case of a knowledge representation system. Let $S =< U, C, D, VAL, f >$ be a *knowledge representation system*, where U is a non-empty set of objects (i.e., $U = \{u_1, u_2, ..., u_n\}$), C is a non-empty set of condition attributes, and D is a non-empty set of decision attributes. We have $A = C \cup D$ which is the set of all attributes and $C \cap D = \emptyset$. Let $VAL = \bigcup_{a \in A} VAL_a$, where for each $a \in A$, VAL_a is a finite attribute domain and the elements of VAL_a are called values of attribute a ($a \in A$). f is an information function such that $f(u_i, a) \in VAL_a$ for every $a \in A$ and $u_i \in U$. Every object which belongs to U is associated with a set of values corresponding to the condition attributes C and decision attributes D.

Suppose B is a nonempty subset of A, u_i, u_j are members of U, and R is an *equivalence relation* over U ($R = U \times U$). We define a binary relation, called an *indiscernibility relation* as $IND(B) = \{(u_i, u_j) \in R : \forall a \in B \quad f(u_i, a) = f(u_j, a)\}$. We say that u_i and u_j are indiscernible by a set of condition attributes B in a knowledge representation system iff $\forall a \in B$, $f(u_i, a) = f(u_j, a)$. The indiscernibility relation partitions U into *equivalence classes*. Equivalence classes of the relation R are called *elementary sets* in an *approximation space* $Apr = (U, R)$. For any object $u_i \subseteq U$, the equivalence classes of the relation R containing U_i be denoted $[u_i]_R$.

Let X be a subset of U, the *lower approximation* of X in Apr is the set $\underline{Apr}(X) = \{u_i \in U | [u_i]_R \subseteq X\}$. The *upper approximation* of X in Apr is the set $\overline{Apr}(X) = \{u_i \in U | [u_i]_R \cap X \neq \emptyset\}$

Table 1 shows an example of a knowledge representation system. $U = \{ u_1, u_2, ..., u_8 \}$. Each object is described by a set of condition attributes C = {S, H, E, C}, with attribute values VAL_S = {0, 1}, VAL_H = {0, 1, 2}, VAL_E = {1,2}, and VAL_C = {0, 1}. The set of values VAL_{CLASS} = { 0,

U	S	H	E	C	CLASS
u_1	0	0	1	0	0
u_2	1	0	2	1	1
u_3	1	1	1	0	0
u_4	0	2	1	1	1
u_5	1	2	1	0	1
u_6	1	0	1	0	0
u_7	1	2	2	1	1
u_8	0	0	2	1	1

Table 1 A knowledge representation system

1} of the decision attribute D represents the set of concept descriptions which are to be learned based on the attribute values of C. In our terminology, the concept is a subset of objects with a particular value of a decision attribute. All objects belonging to the concept are said to be positive, whereas all objects outside the concept are negative.

2.2 Attribute Reduction Techniques

Attribute reduction techniques aim at removing superfluous attributes and finding minimal subsets of attributes, each of which has the same discriminating power as the entire attributes. Let C^* denote the collection of equivalent classes of the relation $IND(C)$ and D^* be a family of equivalent classes of the relation $IND(D)$. The $POS(C, D)$ is a union of lower approximation of all elementary sets of the partition D^* in the approximation space $Apr = (U, IND(C))$ such as $POS(C, D) = \bigcup_{X \in D^*} \underline{Apr}(X)$

Definition 2.1 The *degree of dependency* between the condition attributes and the decision attributes D is denoted as $\gamma(C, D)$ and defined as $\gamma(C, D) = \frac{card(POS(C,D))}{card(U)}$, where *card* denotes set cardinality.

Definition 2.2 The set of attributes P ($P \subseteq C$) is a *reduct* of attributes C which satisfies the following conditions: (1) $POS(P, D) \neq POS(P', D)$, $\forall P' \subset P$ and (2) $\gamma(P, D) = \gamma(C, D)$.

Attribute reducts, denoted as $RED(D)$, are the minimal subsets of condition attributes C with respect to decision attributes D, none of the attributes of any minimal subsets can be eliminated without affecting the essential information. These minimal subsets can discern decision classes with the same discriminating power as the entire condition attributes. Any reduct $RED_i \in RED(D)$, can be used instead of the original system S. For each reduct, we can derive a reduct table from the original knowledge representation system by removing those attributes which are not in the reduct. For example, HE is a reduct of

U	H	E	Class
u_1, u_6	0	1	0
u_2, u_8	0	2	1
u_3	1	1	0
u_4, u_5	2	1	1
u_7	2	2	1

Table 2 A Reduct Table of Reduct HE

Table 1 (we discuss how to compute reducts in Section 3), then a reduct table is obtained as shown in Table 2.

3 GENERATING MULTIPLE KNOWLEDGE

Recently, the subject of multiple knowledge bases (or redundant knowledge) and multiple experts have received considerable attention [7]. A minimal knowledge base employs only the information necessary to represent the given data set without losing essential information. In other words, a minimal knowledge base is a set of decision rules without any redundant attributes and attribute values. Depending on the criterion, one minimal knowledge base can be more useful than another that employs different information. Empirical tests [7, 8, 6] indicate that multiple knowledge is more helpful if it is as accurate and reliable as possible and at the same time as different from the other knowledge as possible. This also seems plausible in real life. Adding a novice is probably counterproductive and adding an expert whose knowledge is too similar to some other members only give more importance to the previous expert [2]. The multiple knowledge bases concept matches the concept of reducts in rough set theory. One reduct table can be obtained from a knowledge representation system by removing those attributes which are not in the reduct without losing any essential information, thus simplify the knowledge representation system. ¿From a reduct table, we can derive a knowledge base which consists of the corresponding decision rules. Using different reducts of a knowledge representation, we can derive different knowledge bases, thus forming multiple knowledge bases.

Our approach uses reducts and decision matrix to construct multiple knowledge bases. The constructing task is to find multiple knowledge bases that can be used to predict the class of an unseen object as a function of its attribute values. One reduct corresponds to a minimal knowledge base in this

U	H	E	C	CLASS
u_1, u_6	0	1	0	0
u_2, u_8	0	2	1	1
u_3	1	1	0	0
u_4	2	1	1	1
u_5	2	1	0	1
u_7	2	2	1	1

Table 3 A simplified Table from Table 1

paper. Our method performs in three steps. First, superfluous attributes are eliminated from databases to improve the efficiency and accuracy of the learning process. Next, using the method first proposed by Ziarko in [15], decision matrices are used to compute the minimal decision rules of the knowledge representation system and then compute a set of reducts which include all the indispensable attributes of the databases to the learning task. Finally, construct different minimal knowledge bases corresponding to different reducts. A minimum knowledge base corresponding to a reduct is a set of decision rules which is *fully covered* by the attributes of a reduct. The fully cover means that all the condition attributes used by the decision rules is also the attributes of the reduct table.

3.1 Elimination of Superfluous Attributes

In the data collection stage, all the features believed to be useful and relevant are collected into the databases. In a database system, we describe each object by the attribute values of C. Very often it turns out that some of the attributes in C may be redundant in the sense that they do not provide any additional information about the objects in S. Thus it is necessary to eliminate those superfluous attributes to improve learning efficiency and accuracy.

Definition 3.1. An attribute $p \in C$ is superfluous in C with respect to D if $POS_C(D) = POS_{C-\{p\}}(D)$, otherwise p is indispensable in C with respect to D.

If an attribute is superfluous in the information system, it should be removed from the information system without changing the dependency relationship of the original system. For example, S is a superfluous attribute in Table 1. Table 3 is obtained by removing it from Table 1. As we can see, Table 3 is simple but has the same discernibility as Table 1.

3.2 Minimal Decision Rules

U	u_2, u_8	u_4	u_5	u_7
u_1, u_6	(E,1)(C,0)	(H,0)(C,0)	(H,0)	(H,0)(E,1)(C,0)
u_3	(H,1)(E,1)(C,0)	(H,1)(C,0)	(H,1)	(H,1)(E,1)(C,0)

Table 4 A decision matrix for class '0'

A *rule* is a combination of values of some condition attributes such that the set of all objects matching it is contained in the set of objects labeled with the same concept, and such that there exists at least one such object. Traditionally, the rule r is denoted as an implication

$$r : (C_1 = V_{i1}) \wedge (C_2 = V_{i2}) \wedge ... \wedge (C_m = V_{im}) \rightarrow (D = V_d),$$

where $C_1, C_2, ...,$ and C_m are the condition attributes and d is a decision attribute.

The process by which the maximum number of condition attribute values of a rule are removed without decreasing the classification accuracy of the rule is called *Value Reduction* [9] and the resulting rule is called *maximally general* or *minimal decision rule.* Thus, a *minimal decision rule* is optimal in the sense that no condition could be removed without decreasing the classification accuracy of the rule. This process checks whether a rule can be made more general by eliminating irrelevant attribute values. An attribute value in a rule is irrelevant if it can be removed from the rule without decreasing its expected classification accuracy, which is computed from the given data set. The *minimal decision rules* minimize the number of rule conditions and are optimal because their conditions are non-redundant. The computing of minimal decision rules is of particular importance with respect to knowledge discovery or data mining applications [14], since they represent the most general patterns existing in the data.

A decision matrix approach was first proposed by Ziarko et al. in [15] to compute all minimal decision rules of a knowledge representation system S and then extended in [6] to large relational databases by integrating attribute-oriented generalization. It provides a way to generate the simplest set of decision rules (i.e., minimum length decision rules) while preserving all essential information. The method is based upon the construction of a number of boolean functions [10, 15] from decision matrices. For more details, please refers to [15, 16].

Example 1: Table 4 and Table 5 depicts two decision matrices obtained from the knowledge representation system given in Table 3. Each cell (i, j) in a

U	u_1, u_6	u_3
u_2, u_8	(E,2)(C,1)	(H,0)(E,2)(C,1)
u_4	(H,2)(C,1)	(H,2)(C,1)
u_5	(H,2)	(H,2)
u_7	(H,2)(E,2)(C,1)	(H,2)(E,2)(C,1)

Table 5 A decision matrix for class '1'

decision matrix is a collection of attribute-value pairs distinguishing row i of the target class from column j of its complement.

¿From Table 4, we can get the following minimal decision rules for the class '0':

$(H = 0) \wedge (E = 1) \rightarrow (CLASS =' 0')$
$(H = 0) \wedge (C = 0) \rightarrow (CLASS =' 0')$
$(H = 1) \rightarrow (CLASS =' 0')$

Similarly, we can obtain the set of minimal rules for the class '1' from Table 5:

$(E = 2) \rightarrow (CLASS =' 1')$
$(C = 1) \rightarrow (CLASS =' 1')$
$(H = 2) \rightarrow (CLASS =' 1')$

3.3 Computing Multiple Reducts

A reduct is a minimal subset of attributes which has the same discernibility power as the entire condition attributes. Finding all the reducts is a NP-complete problem [5, 6]. Fortunately, it is usually not necessary to find all the reducts in a lot of applications including ours. A reduct uses a minimum number of attributes and represent a minimum and complete rules set to classify objects in the databases from "one angle". To classify unseen objects, it is optimal that different reducts use different attributes as much as possible and the union of these attributes in the reducts together include all the indispensable attributes in the databases and the number of reducts used for classification is minimum. Here we proposed a greedy algorithms to compute a set of reducts which satisfy this optimal requirement partially because our algorithm cannot guarantee the number of reducts is minimum. (It may be conjured that this problem is computationally intractable to solve). Our algorithm starts with the core attribute (CO) [6]. (Core is defined as the intersection of all reducts and can be computed easily from the discernibility matrix [10]. The core attributes

U	u_2, u_8	u_4	u_5	u_7
u_1, u_6	EC	HC	H	HEC
u_3	HEC	HC	H	HEC

Table 6 A discernibility matrix

are those entries in the discernibility matrix which have only one attribute. For example, Table 6 is the discernibility matrix for Table 3. The entry (1,3) and (2,3) has only one attribute in it, H is a core attribute.) Then through backtracking, a set of reducts are constructed. A reduct is computed by using forward stepwise selection and backward stepwise elimination based on the significance values of the attributes and the dependency between conditions attributes and decision attributes [6]. The algorithm terminates when the attributes in the union of the reducts includes all the indispensable attributes in the databases.

Algorithm 1: Computing Multiple Reducts

Input: A relation R after elimination of superfluous attributes

Output: A set of reducts $\cup REDU_i$ which have all the indispensable attributes in the relation

$AR = C - CO$; $REDU = CO$; $1 \rightarrow i$

Compute the significant value for each attribute $a \in AR$

Sort the set of attributes AR based on significant values

While the attributes in $\cup\ REDU_i$ does not include all the indispensable attributes in the
databases.

(forward selection:)
While $K(REDU, D) \neq K(C, D)$ **Do** /* Create a subset $REDU$ of attributes C by
adding attributes */
Select the next attribute a_j in AR based on the significant value;
$REDU = REDU \cup \{a_j\}$, $AR = AR - \{a_j\}$

compute the degree of dependency $K(REDU, D)$;
Endwhile

(backward elimination:)
$|REDU| \rightarrow N$

For j=0 to N-1 **Do** /* create a reduct by dropping redundant attributes */
If a_j is not in CO **Then** remove it from $REDU_i$
compute $K(REDU, D)$;
If $K(REDU, D) \neq K(C, D)$ **Then** $REDU \cup a_i \rightarrow REDU$
Endfor
$REDU_i = REDU$; $i + 1 \rightarrow i$; /* backtrack to compute the next reduct */
Endwhile

Using the algorithm, we can find two reducts which have all the three indispensable attributes in Table 3 is {HE, HC}. The complexity of the algorithm cannot be determined exactly since it is highly dependent on the nature of the input data. The number of iteration really varies from data sets. From the experiment on the test data set from [6], it normally terminates after a few iterations. To compute a single reduct, it takes $O(an + aloga)$ in every iteration in the worst case for n objects with a attributes because computing the degree of dependency using a hashing technique is $O(n)$, computing attribute significance value is $O(an)$, sorting the attributes based on the significance is value $O(aloga)$.

Let $RUL_{min} = \{r_1, r_2, ..., r_k\}$ be the set of all minimal decision rules generated by decision matrix method and let $RED = \{RED_1, RED_2, ..., RED_i\}$ be the attribute reducts computed from the algorithm. A minimal knowledge base denoted as $KBRED_i$ ($RED_i \in RED$) is defined as $KBRED_i = \bigcup\{Cond(r_k) \subseteq Cond(RED_i) : r_k \in RUL_{min}\}$, where $Cond()$ is the set of attribute names.

For example, in *Example 1*, we have the set of reducts $RED = \{HE, HC\}$ with respect to decision attribute. According to the above definition, the minimized knowledge bases corresponding to reducts "HE" and "HC" are the following sets of decision rules extracted from all minimal decision rules:

The minimal knowledge base $KBRED_1$ for reduct "HE" is

$(H = 0) \wedge (E = 1) \rightarrow (CLASS =' 0')$
$(H = 1) \rightarrow (CLASS =' 0')$
$(E = 2) \rightarrow (CLASS =' 1')$
$(H = 2) \rightarrow (CLASS =' 1')$

The minimal knowledge base $KBRED_2$ for reduct "HC" is

$(H = 0) \wedge (C = 0) \rightarrow (CLASS =' 0')$
$(H = 1) \rightarrow (CLASS =' 0')$
$(C = 1) \rightarrow (CLASS =' 1')$
$(H = 2) \rightarrow (CLASS =' 1')$

In summary, the algorithm of generating multiple knowledge is as follow:
Algorithm 2: Generating Multiple Knowledge Bases (DBMkbs)

Input: a relational system R

Output: Multiple knowledge bases $\cup KBREDU_i$

Step 1: Remove superfluous attributes from the databases

Step 2: Compute the minimal decision rules through decision matrices

Step 3: Compute a set of reducts which cover all the indispensable attributes in the databases.

Step 4: Group the minimal decision rules to the corresponding reducts to form a multiple knowledge bases

3.4 Test Results

Table 7 shows the test results of DBMaxi [6](which generate all the minimal decision rules), DBDeci [4, 5] (which generates only a set of minimal decision rules) and DBMKbs. For a detailed explanation of the data set, please refer to [13]. DBMaxi represent the upper bound of the classification accuracy. As can be see, the result of DBMkbs is very close to the upper bound. However, the problem of how to combine decisions of multiple knowledge bases remains. Currently, there are three strategies for combining multiple sets of knowledge rule: (1) Sum of distribution (2) Voting [7] (3) Naive Bayesian combination [8]. In the test, Voting is adopted to solve classification confliction. These three strategies are complementary to each other, each has its strong and weak point depending on the domain. A deep analysis and comparison of these strategies

Methods	Iris	Appendicitis	Thyroid	Cancer	Average
DBMaxi	96.33	91.05	98.86	98.30	96.14
DBMkbs	96.00	90.80	98.00	97.50	95.57
DBDeci	94.33	88.20	95.06	95.56	93.29

Table 7 The Comparative Performance

and developing new methods for combining multiple knowledge bases rules are one of our current research topics.

4 CONCLUSION

In this paper, an approach for constructing multiple knowledge bases based on rough set theory is presented. The concept of rough set offers a sound theoretical foundation for multiple sets of knowledge rules. Multiple knowledge base systems can be formulated precisely and in an unified way within the framework of rough set theory. In addition, by using the multiple knowledge bases, the classification accuracy may increase and the ability of explanation may improve as the same decision is explained by using many different "point of view", which is one of the weak points of the inductive generated (nonredundant) sets of decision rules.

REFERENCES

[1] B. Cestnik and I. Bratko, "Learning Redundant Rules in Noisy Domains" in *Prod. of European Conf. on AI*, Munich, Germany, pp348-350, 1988.

[2] M. Gams, "New Measurements Highlight the Importance of Redundant Knowledge" in *Proc. 4th Europe Working Session on Learning*, Momtpellier, pp71-80, 1989

[3] X. Hu, and N. Shan, "Rough Set and Multiple Knowledge Bases" in *Proc. of the 7th Florida AI Research Symposium*, Pensacola Beach, FL, USA, pp255-258, 1994

[4] X. Hu, N. Cercone,"Discovery of Decision Rules from Databases", in *the 3rd International Conference on Information and Knowledge Management (CIKM94)*, Nov., 1994

[5] X. Hu, N. Cercone, "Learning from Databases: A Rough Set Approach", in *Computational Intelligence: An International Journal*, 11(2), 323-338, 1995

[6] X. Hu, "Knowledge Discovery from Databases: An Attribute-Oriented Rough Set Approach" Ph.D. thesis, Dept. of Computer Science, University of Regina, Canada, June 1995

[7] I. Kononenko, "An Experiment in Machine learning of Redundant Knowledge" in *Proc. Intern Conf. MELECON*, Ljubljana, Slovenia, pp1146-1149

[8] I. Kononenko and M. Kovacie, "Learning as Optimization: Stochastic Generation of Multiple knowledge" in *Proc. of the 9th International Workshop on Machine Learning*, Aberden,Scotland, July 1-3, pp. 256-262, 1992

[9] Z. Pawlak, "Rough Sets" in *International Journal of Information and Computer Science*, Vol. 11(5), pp341-356, 1982

[10] A. Skowron and C. Rauszer, C, "The Discernibility Matrices and Functions in Information Systems" in *Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory*, edited by R. Slowinski, Kluwer Academic Publishers, 1992.

[11] R. Slowinski, *Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory*" Kluwer Academic Publishers, 1992

[12] L. Torgo, "Controlled Redundancy in Incremental Rule Learning" in *Proc. of European Conf. on Machine Learning*, pp185-195, 1993

[13] S.M. Weiss and I. Kapouleas, "An Empirical Comparison of Pattern Recognition Neural Nets, and Machine Learning Classification Methods", *Proc. of the 11th International Joint Conf. on AI*, pp781-787, 1989

[14] W. Ziarko, *Rough Sets, Fuzzy Sets and Knowledge Discovery* Springer-Verlag, 1994.

[15] W. Ziarko and N. Shan, "A Method For Computing All Maximally General Rules in Attribute-Value Systems" in *Computational Intelligence: An International Journal*, to appear.

[16] W. Ziarko, N. Cercone, X. Hu, "Rule Discovery from Databases with Decision Matrices", *to appear in proceeding of the 9th International Symposium on Methodologies for Intelligent Systems, 1996*

7

FUZZY CONTROLLERS: AN INTEGRATED APPROACH BASED ON FUZZY LOGIC, ROUGH SETS, AND EVOLUTIONARY COMPUTING

T. Y. Lin

Department of Mathematics and Computer Science,
San Jose State University, One Washington Square,
San Jose, California 95192-0103

ABSTRACT

There are two approaches to control theory. One is the classical hard computing approach. Its modern theory is based on differential geometry and topology. Another is fuzzy logic, a soft computing approach. Taking the views of both hard and soft computing, an integrated approach is proposed. The mathematical formalism for such an integrated structure is called the rough logic government. Intuitively, fuzzy *logic is viewed* as a methodology of constructing functions by *a grand scale interpolation guided by qualitative information.* Model theory of rough logic sysyem is used to formalized the design of classical fuzzy logic controllers. The design is formulated as a sequence of trasnformations of mathematical models of a control system. It starts from a symbolic model that consists of predicates or propositions in rough logic. Such a model is referred to as a theory in formal logic. By experts' suggestion, called fuzzificaton, t he symbolic model can be transformed into a fuzzy model that usually consists of rules of fuzzy sets (membership functions). In formal logic, such a trânsformation is called an interpretation of the theory. Of course, interpretations are usually not unique. The collection of all such interpretations is a higly structured set of membership functions; it is called a fibre space by differential geometers and topologists. Using one of the usual inference methods, the cross-sections of the fibre space is tranformed into a "virtual" space of trajectories or integral submanifold of a differentiable manifold. Conceptually, some of these "virtual" trajectories or integral submanifolds should correspond to some solutions of the system equations of a classical dynamic system. By veification and validation, part of the "virtual" space solidifies into a "real" space of trajectories or integral submanifolds of a differentiable manifold. These "real" trajectories or integral submanifolds are solutions of the system equations of a classical dynamic system; of course, in fuzzy logic design,

these equations are usually not expiliclty constructed. Several new applications are identified, most notably one is the stability problem.

1 INTRODUCTION

Rough set methodology is an emerging new technology in handling the classificatory analysis of imprecise, uncertain or incomplete information systems. In this paper, we integrate this new technology together with evolutionary computing technology into the design theory of fuzzy logic control (FLC).

One of the important novelties of FLC design is that the experts domain knowledge is the integral part of the system. FLC designer implicitly or explicitly believe that a control system has an underlying logic system behind it. So the essence of fuzzy logic control design is to search for such a underlying logic system. In this paper, we formalize such search.

2 MATHEMATICAL MODELS IN CONTROL

What is a mathematical models? The following paragraph about mathematical model is depict from a common text book of discrete mathematics [15].

> A mathematical model is a mathematical characterization of a phenomena or a process. So mathematical model has three essential parts: a process or phenomena which is to be modeled, a mathematical structure capable of expressing the important properties of the object to be modeled, an explicit correspondence between the two.

The classical controller(CC) and FLC, though, have the same goal, they focus on different aspects of the same process. So they use different mathematics, and hence have different methods for proving the correctness of the model (verification and validation).

In CC, one often starts from a model of the system [6]:

$$X' = F(t, U, X) \tag{1.1}$$

$$Y = H(t, U, X) \tag{1.2}$$

Then from the solutions of (1.1), one obtains the control function

$$Y = K(t, U) \tag{1.3}$$

where X, U, and Y represent respectively the state, the input, and the observable output variables in vector forms. The mathematical models (1.1) and (1.2) are often derived from the nature laws and/or engineering principles. Hence the correctness of the correspondence between the mathematical structure and the real world phenomena or process is obvious, and hence the verification and validation step is often brief or fell into background. On the other hand if systems are complex and precise (or even approximate) analytic descriptions of systems are unavailable, then CC approaches have to halt and to wait for new mathematical development. FLC designers then choose to model the solution directly (without the equations of a system). In other words, classical control models the system, while FLC models the solution without the system model (equations).

Their differences are fundamental and intrinsic, so are their respective methodologies. FLC often needs long and tedious experiments-called tuning- to prove that the solution model is indeed correct. The goal of this paper is to propose theoretical foundation so that such proofs (verification and validation) can be conducted in a more organized and systematic fashion. We are aware that some modern formulation of control has extend the differential equations to differential inclusions [2], so FLC designers may often attack more general problems than the solution of differential equations

3 AN OVERVIEW OF FUZZY LOGIC CONTROLS

We will express FLC theory in terms of the framework of fuzzy graphs [17], because it is equivalent, yet more concise. Let us set up our notations and terminology.

1. U is the input vector and Y is the output vector.
2. The desirable control function $Y = K(U)$.
3. $SymIn$ is a finite set of input symbols and $SymOut$ is a finite set of output symbols.
4. The mapping $\kappa : SymIn \rightarrow SymOut$, denoted by $\eta = \kappa(\upsilon)$, is a qualitative approximation of the control function described by experts. This mapping is called symbolic control function, or in graph notations, the graph $(\eta, \kappa(\upsilon))$ is called symbolic graph. In current practice, it is often expressed by a set of linguistic rules – a set of rules using linguistic variables.
5. The assignment of a membership function to a symbol is called fuzzy interpretation.
6. η, υ are variables in $SymIn$ or $SymOut$ respectively; they are called linguistic variables, however, no interpretations are assigned at this level.
7. Note that we did not use the term linguistic variable in Zadehs technical sense ([18], p. 132). We did not include the interpretation as a part of the definition of a linguistic variable. In our approach, we want to vary these fuzzy interpretations. In other words, the tie between symbols and membership functions are dynamic.

Now we will describe the FLC design procedure in our terminology.

3.1 Step 1: A symbolic graph – a set of linguistic rules

In this step, experts capture the unknown control function, $Y = K(U)$, qualitatively by a symbolic function, $\eta = \kappa(\upsilon)$, where υ and η are two variables that range respectively through $SymIn = \{Sin_1, Sin_2, \ldots Sin_i \ldots Sin_h\}$ and $SymOut = \{Sout_1, Sout_2, \ldots Sout_i \ldots Sout_k\}$. In current practice, the symbolic function $\eta = \kappa(\upsilon)$ is represented by a set of linguistic rules. Abstractly, $Y = K(U)$ is a function of vector spaces, while $\eta = \kappa(\upsilon)$ is a function of finite sets. Intuitively, the two functions are related by experts interpretations of the symbolic input and output. The symbolic graph (υ, η) is a mathematical representation of the set of linguistic rules.

In the Figure 1. below, it illustrated the interpreted symbolic graph. The curve is the unknown control function, $Y = K(U)$. Experts use the symbols(represented by boxes) to capture its behavior qualitatively.

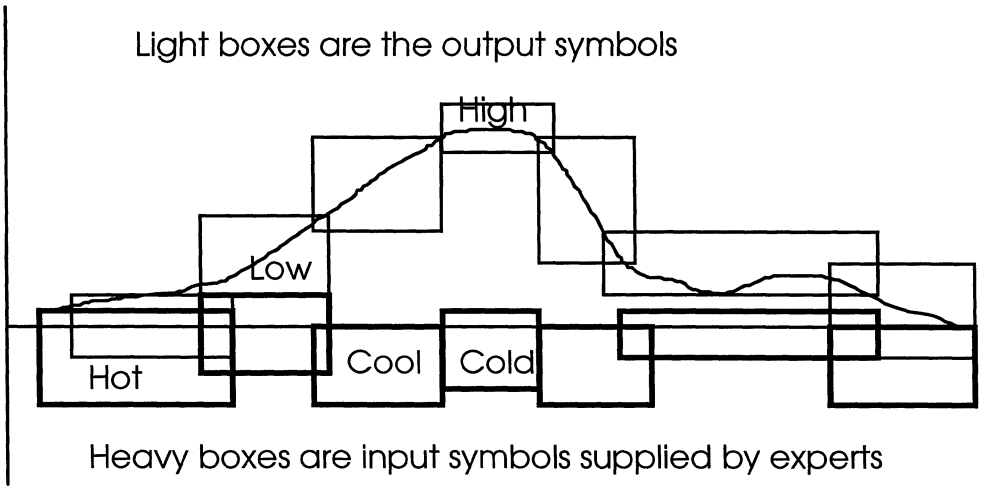


Figure 1 Interpreted symbolic graph

Example 1A. Let a set of if-then rules using linguistic variables be as follows

IF temperature IS cold, THEN fan_speed IS high
IF temperature IS cool, THEN fan_speed IS medium
IF temperate IS warm, THEN fan_speed IS low
IF temperature IS hot, THEN fan_speed IS zero

Then the rules can be expressed by the symbolic graph:
$SymIn = cold, cool, warm, hot,$
$SymOut = high, medium, low, zero,$

v	$\eta = \kappa(v)$
cold	high
cool	medium
warm	low
hot	zero

3.2 Step 2. Fuzzy graphs – fuzzification

To get the desirable control function, experts interpret each input symbol $Sin_i \in SymIn$ by a membership function f_i, in notation, the interpretations are the assignments:

$$Sin_i := f_i, \qquad i = 1, 2, \ldots h; \tag{1.4}$$

Each assignment is called a fuzzy interpretation of the linguistic constant. For the output symbols, in TVFI case, one uses supporting values. We will base our exposition on this approach.

$$Sout_i := g_j, \qquad j = 1, 2, \ldots k \tag{1.5}$$

where g_j, $j = 1, 2, \ldots k$ are supporting values (constants)

Example 1B Suppose experts suggest the following interpretations:

$$Cold = Sin1 := f1,$$
$$Cool = Sin2 := f2,$$
$$Warm = Sin3 := f3,$$
$$Hot = Sin4 := f4,$$
$$High = Sout1 := g1,$$
$$Medium = Sout2 := g2,$$
$$Low = Sout3 := g3,$$
$$Zero = Sout4 := g4$$

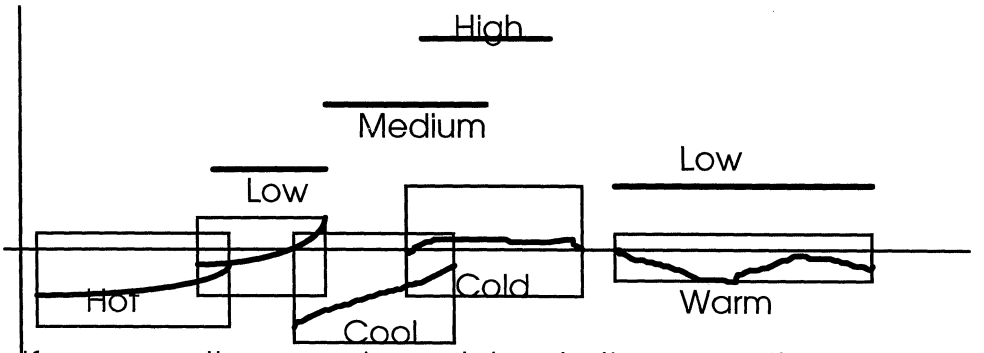


Figure 2 Fuzzy interpretation of the linguistic constant

(e.g, the support values are $g1 = 800rpm$, $g2 = 500rpm$, $g3 = 200rmp$, $g4 = 0rpm$) Then the symbolic graph is transform to a fuzzy graph $(u, k(u))$ – a fuzzy approximation.

v	$k(u)$
f_1	g_1
f_2	g_2
f_3	g_3
f_4	g_4

3.3 Step 3. Candidate graph – defuzzification

Based on TVFI or Mamdani inference methods [1], the fuzzy graph $(u, t(u))$ can be transformed into a vector graph $(u, k(u))$ which is a candidate graph of the unknown control function. To be more precise, the vector graph $(u, k(u))$ is the graph of a function which is a un-verified and un-validated candidate of an approximation of the unknown control function. The graph is called candidate graph and the function is called candidate function – a proposed unverfied approximation.

u	$(u, k(u))$
	5 RPM
78 degrees	...
...	350 RPM
...	...

3.4 Step 4. Verification and validation – Tuning

In CC, the verification and validation of a control function is a relatively easy task, because the work is embedded in the system modeling. In FLC, system modeling is skipped, so the control functions can not simply be verified/validated as a solution of the system model. The verification and validation of the fuzzy graph (that is to show that the candidate graph is indeed the graph of the real world control function) has to be carried out directly by experiments. Of course experts' interpretation (assignment of a membership function to each symbol) can not so precise that they can pinpoint exactly one membership fucntion for each symbol. In practices, there are many candidates for interpretations. For each symbol $Sin_i, i = 1, 2, \ldots, h$, let

$$Fin_i = \{f_i, f_i', f_i'', \ldots, f_i^k \ldots\}, \qquad i = 1, 2, \ldots, h \tag{1.6}$$

be a family of membership functions that interpret the symbol Sin_i. The so-called turning is to conduct experiments by varying f_i^k through Fin_i until a correct one is found. We propose to use evolutionary computing to breed better candidates, and to use rough sets to organize candidates. Roughly, FLC design is a grand scale extrapolation based on human intuition.

Example 1D

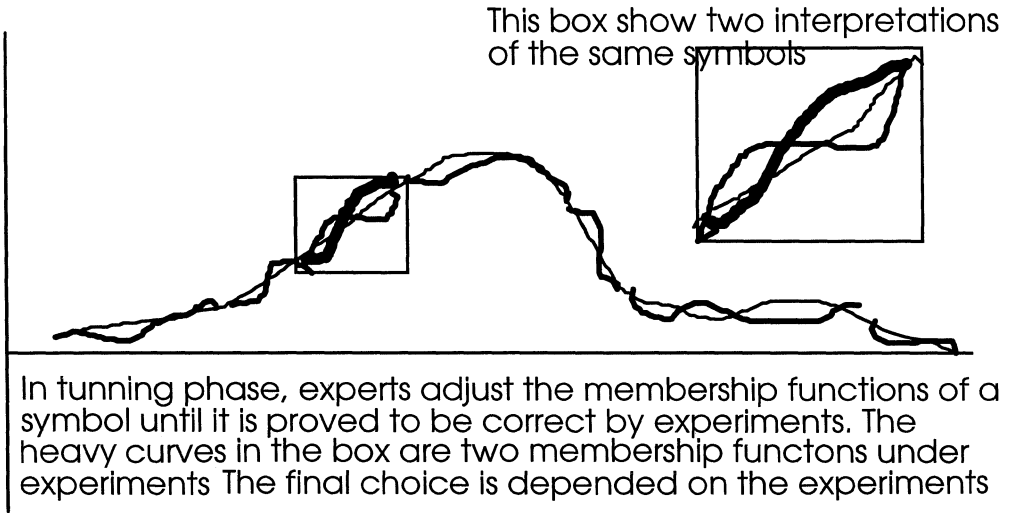


Figure 3 Interpretation of symbols

4 POSSIBLE WORLDS MODEL THEORY FOR ROUGH LOGIC

Expanding their axiomatic characterization of rough sets, Lin and Liu developed the first order rough logic [10]. It can be shown that it is equivalent to first order modal logic S5 [12, 11]. The Possible worlds are called observable worlds [10]. We will call them possible worlds here, since it is equivalent and better known. In this paper, we do not need the full theory, so we will illustrate its world model by an example.

Let $E = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ be the universe of discourse. Let R be an equivalence relation which partitions E into three equivalence classes:

$$H^1 = \{3, 6, 9\},$$

$$H^2 = \{2,5,8\},$$
$$H^3 = \{1,4,7\}.$$

Intuitively, H^i represents an equivalence class of elements which are indistinguishable from each other. The collection of one representative from each equivalence class is the possible worlds of modal logic and we have called them observable worlds [10]. For this example, all possible worlds are:

$W^1 = \{1,2,3\}$, $W^2 = \{1,2,6\}$, $W^3 = \{1,2,9\}$, $W^4 = \{1,5,3\}$,
$W^3 = \{1,5,6\}$, $W^6 = \{1,5,9\}$, $W^7 = \{1,8,3\}$, $W^8 = \{1,8,6\}$,
$W^9 = \{1,8,9\}$, $W^{10} = \{4,2,3\}$, $W^{11} = \{4,2,6\}$, $W^{12} = \{4,2,9\}$,
$W^{13} = \{4,5,3\}$, $W^{14} = \{4,5,6\}$, $W^{15} = \{4,5,9\}$, $W^{16} = \{4,8,3\}$,
$W^{17} = \{4,8,6\}$, $W^{18} = \{4,8,9\}$, $W^{19} = \{7,2,3\}$, $W^{20} = \{7,2,6\}$,
$W^{21} = \{7,2,9\}$, $W^{22} = \{7,5,3\}$, $W^{23} = \{7,5,6\}$, $W^{24} = \{7,5,9\}$,
$W^{25} = \{7,8,3\}$, $W^{26} = \{7,8,6\}$, $W^{27} = \{7,8,9\}$.

Our notion of possible worlds is different from standard modal logic. To stress this fact we will say that the possible world Wi is derived from E. The collection of these possible worlds or observable worlds $\{W^j | j = 1,\ldots,27\}$ is called the World Model W on E [10]. We should point out again, this world model is equivalent to the world model of the first order S5 logic formulated in [12].

5 TUNING BY ROUGH LOGIC

The tuning of FLCs will be organized by the World Model of rough logic theory. We will us illustrate the idea by examples.

Let $Fin_1 = \{f_1, f_1', f_1'', ...\}$ be the collection of candidates for interpreting the linguistic constant $Sin_1 = Cold$ (suggested by experts):

Cold: (heavy line is common to three membership functions)

Similarly, let $Fin_2 = \{f_2, f_2', f_2'', ...\}$ be the possible interpretations of the linguistic constant $Sin_2 = Cool$,

For clarity we have drawn separate figures for *Cool* and *Cold*. As usual, these two classes of functions have overlapping supports (support= $\{x \mid f(x) > 0\}$). A symbolic graph can be interpreted into several fuzzy graphs by varying the membership functions for each symbols. Here we illustrate three choices. Let the interpretations of linguistic constant Cold vary through $Fin_1 = \{f_1, f_1',$

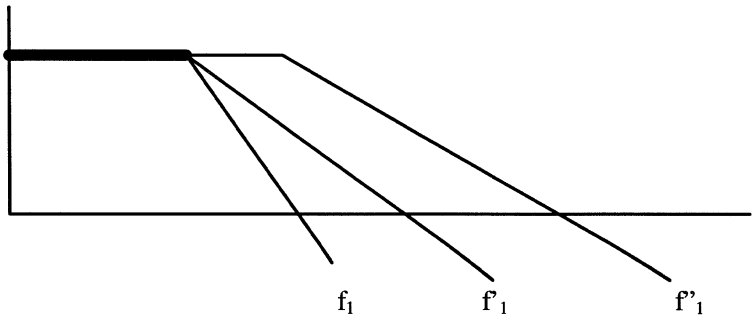


Figure 4 Cold: heavy line is common to three membership functions

$f_1'', \ldots\}$, and Cool vary through $Fin_2 = \{f_2, f_2', f_2'', \ldots\}$, then we have following fuzzy graphs:

cold	f_1
cold	f_2
warm	f_3
hot	f_4

f_1'	g_1
f_2	g_2
f_3	g_3
f_4	g_4

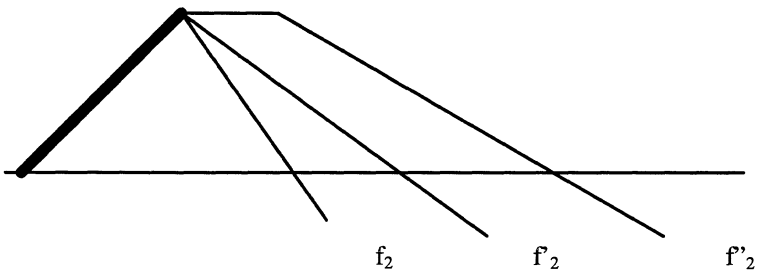


Figure 5 Possible interpretations of "Cool"

cold	f_1'
cool	f_2
warm	f_3
hot	f_4

f_1'	g_1
f_2	g_2
f_3	g_3
f_4	g_4

cold	f_1''
cool	f_2
warm	f_3
hot	f_4

f_1''	g_1
f_2	g_2
f_3	g_3
f_4	g_4

cold	f_1^4
cool	f_2
warm	f_3
hot	f_4

f_1^4	g_1
f_2	g_2
f_3	g_3
f_4	g_4

6 ROUGH GOVERNMENT OF FUZZY CONTROLLERS

6.1 The Structure of the Linguistic Rules

A symbolic graph (a set of linguistic rules) is a qualitative approximation to the unknown control function, it represents an insight of experts. If the control function is known, we can create a symbolic graph, called the ideal symbolic graph, which will perform the task exactly as the control function. The symbolic graph is a set of proper axioms in a Rough Logic Theory RLT. The ideal symbolic graph is the underlying first order logic. For a given control problem, experts may propose several symbolic graphs $SG(1), SG(2), \ldots$. These SG represent experts very very well educated guess. So we should refine them using the genetic algorithm during the tuning. Let $SG(H), H = 1, 2, \ldots, s$, be these symbolic graphs, and let $RLT(H)$ be the corresponding rough logic theories. In principle, if we have adequate family of $RLT(H)$, we have

$$\text{Underlying First Order Logic} = \lim_{\leftarrow H} RLT(H) \tag{1.7}$$

Example 2. We have a set the linguistic rules:

IF temperature IS very cold	THEN fan_speed IS very high
IF temperature IS moderate cold	THEN fan_speed IS moderate high
IF temperature IS quite cool	THEN fan_speed IS upper medium
IF temperature IS moderate cool	THEN fan_speed IS medium
IF temperate IS moderate warm	THEN fan_speed IS moderate low
IF temperate IS very warm	THEN fan_speed IS very low
IF temperature IS hot	THEN fan_speed IS near zero
IF temperature IS very hot	THEN fan_speed IS exactly zero

It is clear that this set of linguistic rules is a refinement of the set in Example 1, and hence there is a homomorphism between the corresponding two RLTs [4]. The inverse limit is taking in this sense [5].

6.2 The World Model -The Structure of Fuzzy Rules

For a fixed H, and for a linguistic variable $Sin(H)_j$, we have a set of membership functions to interpret it.

$$Fin(H)_j = \{H_j, H'_j, H''_j, ..., H^k_j\} \tag{1.8}$$

Let the union of these interpretation be

$$FuzIn(H) = \bigcup \{Fin(H)_i | i = 1, 2, ...m(i)\} \tag{1.9}$$

Note that $Fin(H)_i$'s are pairwise disjoint, they form a partition on $FuzIn(H)$. $FuzIn(H)$ together with this partition is a rough structure for the rough logic

theory $RLT(H)$. A possible world is a selection one representative from each $Fin(H)_i$, so

$$\begin{aligned} &The\ World\ Model\ W(H)\ on\ FuzIn(H) \\ &= the\ collection\ of\ possible\ worlds \\ &= the\ collection\ of\ fuzzy\ graphs. \end{aligned}$$

6.3 Building the Correct Fuzzy Rules

If we know the rough logic $RLT(H)$ is correct, then there is a correct possible world in $W(H)$. In other words, the fuzzy graph corresponding to this correct possible world will produce the desirable control functions. However, in reality we do not know which one is correct, so we have to zigzag through the tuning to find a correct $RLT(H)$. If the control function of an application were unique, then our search would be very lengthy and difficult. Fortunately, in most applications, acceptable control functions are abundant, so such search for correct $RLT(H)$ is possible. Our main strategy is to use evolutionary computing, such as genetic algorithm to breed the better fuzzy graph, and to use rough logic to govern such population of correct fuzzy graphs

Acknowledgements

This research is partially supported by Electric Power Research Institute, Palo Alto, California.

REFERENCES

[1] Aptronix, FIDE, User Manual, Aptronix, 1992.

[2] Jean-Piere Aubin, *Viability Theory*, Birkhauser, Basel, 1991.

[3] B. Chellas, *Modal Logic - An Introduction*, Cambridge University Press 1980.

[4] H. Enderton, *A Mathematical Introduction to Logic*, Academic Press, 1972.

[5] S. Eilenbergand N. Steenrod, *Foundation of Algebraic Topology*, Princeton Press, 1952.

[6] G. Franklin, J. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 2nd ed., Ch. 8, 1990.

[7] K. Kurman, *Feedback Control Theory and Design*, Elsevier, 1994.

[8] T. Y. Lin., Patterns in Data-Rough Sets and Foundation of Intrusion Detection Systems, *Journal of Computer Science and Decision Support*, Vol.18, No. 3-4, 1993, pp. 225–241.

[9] Patterns in Data—A Soft Computing Approach, in *Proceedings of the New Security Paradigms Workshop*, IEEE Computer Society Press, 1994.

[10] T. Y. Lin, Q. Liu, First Order Rough Logic I:Approximate Reasoning via Rough Sets, *Fundmentae Informaticae*, 1994.

[11] T. Y. Lin First Order Rough Logic II; Convergence and Completeness, CSC'95 Workshop on Rough sets and Database Mining, 1995

[12] W. Lukaszewicz, *Non-Monotonic Reasoning*, Institute of Information, Univeristy of Warsaw, Poland, 1990, Section 1.6.

[13] T. Munakata, Fuzzy Systems: An Overview. Communication of the ACM **37,3** 1994, pp. 69–76.

[14] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*, 1991.

[15] D. Stanat and D. McAllister, *Discrete Mathematics in Computer Science*, Prentice Hall, Englewood Cliffs, N.J., 1972.

[16] A. Tettamanzi, *Proceedings of the 6th IEEE Conference on Tools in AI*, New Orleans, Nov. 6-9, 1994.

[17] L. Zadeh, Fuzzy logic, Neural networks, and Soft Computing, *Communications of the ACM* **37,3), 1994, pp. 77–84.**

[18] H. Zimmeman, *Fuzzy Set Theory and Its Applications*, **Klumer Academic Press, 1991**

8

ROUGH REAL FUNCTIONS AND ROUGH CONTROLLERS

Zdzisław Pawlak

Institute of Computer Science
University of Technology
ul. Nowowiejska 15/19, 00 665 Warsaw, Poland

and

Institute of Theoretical and Applied Informatics
Polish Academy of Sciences
ul. Baltycka 5, 44 000 Gliwice, Poland
e-mail: zpw@ii.pw.edu.pl

1 INTRODUCTION

This paper is an extension of articles Pawlak (1987), where some ideas concerning rough functions were outlined. The concept of the rough function is based on the rough set theory (Pawlak, 1991) and is needed in many applications, where experimental data are processes, in particular as a theoretical basis for rough controllers (Czogala et al., 1994, Mrozek and Plonka, 1994).

The presented approach is somehow related to nonstandard analysis (Robinson, 1970), measurement theory (Orlowska and Pawlak, 1984) and cell-to-cell mapping (Hsu, 1980) but these aspects of rough functions will be not considered here.

In recent years we witness rapid grow of development and applications of fuzzy controllers. The philosophy behind fuzzy control is that instead of describing, as in the case of classical control theory, the process being controlled in terms of mathematical equations - we describe the behavior of human controller in terms of fuzzy decision rules, i.e. rules that involve rather qualitative then quantitative variables and can be seen as a common-sense model of the controlled process, similarly as in qualitative physics physical phenomena are described in terms of qualitative variables instead of mathematical equations.

The idea of rough (approximate) control steams yet from another philosophical background. It is based on the assumption that the controlled process is observed and data about the process are registered. The data are then used to generate the control algorithms, which can be afterwards optimized. Both, the generation of the control algorithm from observation, as well the optimization of the algorithm can be based on the rough set theory, which seems to be very well suited for this kind of tasks. The control algorithms obtained in this way are objective and can be viewed as an intermediate approach between classical and fuzzy approach to control systems.

In some cases the observation can be postponed and control algorithm can be obtained directly from the knowledgeable expert, similarly as in the fuzzy set approach. In this case the control algorithm can be also simplified using the rough set theory approach.

In general we assume that a rough controller can be seen as an implementation of rough (approximate) function, i.e. function obtained as a result of physical measurements with predetermined accuracy, depending on assumed scale.

The aim of this paper is to give basic ideas concerning rough functions, which are meant to be used as a theoretical basis for rough controllers synthesis and analysis. The presented ideas can be also applied to other problems – in general to discrete dynamic systems, and will be discussed in further papers.

2 BASIC OF THE ROUGH SET CONCEPT

Basic ideas of the rough set theory can be found in Pawlak (1991). In this section we will give only those notions which are necessary to define concepts used in this paper.

Let U be a finite, nonempty set called the *universe*, and let I be an equivalence relation on U, called an *indiscernibility relation.* By $I(x)$ we mean the set of all y such that xIy, i.e. $I(x) = [x]_I$, i.e.- is an equivalence class of the relation I containing element x. The indiscernibility relation is meant to capture the fact that often we have limited information about elements of the universe and consequently we are unable to discern them in view of the available information. Thus I represents our lack of knowledge about U.

We will define now two basic operations on sets in the rough set theory, called the *I-lower* and the *I-upper approximation*, and defined respectively as follows:

$$I_*(X) = \{x \in U : I(x) \subseteq X\},$$

$$I^*(X) = \{x \in U : I(x) \cap X \neq \emptyset\}.$$

The difference between the upper and the lower approximation will be called the *I-boundary* of X and will be denoted by $BN_I(X)$, i.e.

$$BN_I(X) = I^*(X) - I_*(X).$$

If $I^*(X) = I_*(X)$ we say the the set is *I-exact* otherwise the set X is *I-rough*. Thus rough sets are sets with unsharp boundaries.

Usually in order to define a set we use the membership function. The membership function for rough sets is defined by employing the equivalence relation I as follows:

$$\mu_X^I(x) = \frac{card(X \cap I(x))}{card(I(x))}.$$

Obviously

$$\mu_X^I(x) \in [0, 1].$$

The value of the membership function expresses the degree to which the element x belongs to the set X in view of the indiscernibility relation I.

The above assumed membership function, can be used to define the two previously defined approximations of sets, as shown below:

$$I_*(X) = \{x \in U : \mu_X^I(x) = 1\},$$

$$I^*(X) = \{x \in U : \mu_X^I(x) > 0\}.$$

3 ROUGH SETS ON THE REAL LINE

In this section we reformulate the concepts of approximations and the rough membership function referring to the set of reals, which will be needed to formulate basic properties of rough real functions.

Let $\mathbf{R}$ be the set of reals and let (a,b) be an open interval. By a *discretization* of the interval (a,b) we mean a sequence $S = \{x_0, x_1, \ldots, x_n\}$ of reals such that $a = x_0 < x_1 < \ldots < x_n = b$. Besides, we assume that $0 \in S$. The ordered pair

$A = (\mathbf{R}, S)$ will be referred to as the *approximation space* generated by S or simple as *S-approximation space.* Every discretization S induces the partition $\pi(S) = \{\{x_0\}, (x_0x_1), \{x_1\}, (x_1, x_2), \{x_2\}, (x_2, x_3),$
$\{x_3\}, \ldots, \{x_{n-1}\}, (x_{n-1}, x_n), \{x_n\}\}$ on (a, b). By $S(x)$ (or $[x]_S$) we will denote block of the partition $\pi(S)$ containing x. In particular, if $x \in S$ then $S(x) = \{x\}$. If $S(x) = (x_i, x_{i+1})$, then by $S_*(x)$ and $S^*(x)$ we will denote the left and the right ends of the interval $S(x)$ respectively, i.e. $S_*(x) = x_i$ and $S^*(x) = x_{i+1}$. The *closure* of $S(x)$ will be denoted by $S'(x)$.

In what follows we will be interested in approximating intervals $(0, x) = Q(x)$ for any $x \in [a, b]$.

Suppose we are given an approximation space $A = (\mathbf{R}, S)$. By the *S-lower* and the *S-upper* approximation of $Q(x)$, denoted by $Q_S(x)$ and $Q^S(x)$ respectively, we mean sets defined below:

$$Q_S(x) = \{y \in \mathbf{R} : S(y) \subseteq Q(x)\} = Q(S_*(x))$$

$$Q^S(x) = \{y \in \mathbf{R} : S(y) \cap Q(x) \neq \emptyset\} = Q(S^*(x)).$$

The above definitions of approximations of the interval $(0, x)$ can be also understood as approximations of the real number x which are simple the ends of the interval $S(x)$.

In other words given any real number x and a discretization S, by the S-lower and the S-upper approximation of x we mean the numbers $S_*(x)$ and $S^*(x)$ respectively.

We will say that the number x is exact in $A = (\mathbf{R}, S)$ if $S_*(x) = S^*(x)$, otherwise the number x is *inexact* (*rough*) in $A = (\mathbf{R}, S)$. Of course x is exact iff $x \in S$.

Any discretization S can be interpreted as a scale (e.g. km, in, etc.), by means of which reals from $\mathbf{R}$ are measured with some approximation due to the scale S.

The introduced idea of the rough set on the real line corresponds exactly to those defined for arbitrary sets and can be seen as a special case of the general definition.

Now we give the definition of the next basic notion in the rough set approach - the rough membership function – referring to the real line (Pawlak and Skowron, 1993).

The rough membership function for set on the real line has the form

$$\mu_{Q(x)}(y) = \frac{\Delta(Q(x) \cap S(y))}{\Delta(S(y))},$$

where $\Delta(X) = Sup|x - y|, x, y \in X$.

Assuming that $x = y$, we get

$$\mu_{Q(x)}(y) = \mu(y),$$

which can be understood as an error of measurement of x in the scale S.

Remark

We can also assume that the discretizatin S induces partition $\pi(S) = \{(-\infty, x_0), \{x_0\}, (x_0x_1), \{x_1\}, (x_1, x_2), \{x_2\}, (x_2, x_3), \{x_3\}, \ldots, \{x_{n-1}\}, (x_{n-1}, x_n), \{x_n\}, (x_n, +\infty)\}$ on $\mathbf{R}$. In this case for $x > b$ the upper approximation of x is $S^*(x) = +\infty$, and similarly for $x < a$, we have $S^*(x) = -\infty$. However for the sake of simplicity we will not consider this case here. □

4 ROUGH SEQUENCIES AND ROUGH FUNCTIONS

Let $A = (\mathbf{R}, S)$ be an approximation space and let $\{a_n\}$ be an infinite sequence of real numbers.

A sequence $\{a_n\}$ is *roughly convergent* in $A = (\mathbf{R}, S)$, (*S-convergent*), if there exists i such that for every $j > i$ $S(a_j) = S(a_i)$; $S_*(a_i)$ and $S^*(a_i)$ are referred to as the *rough lower* and the *rough upper limit* (*S-upper, S-lower limit*) of the sequence $\{a_n\}$. Any roughly convergent sequence will be called *rough Cauchy sequence.*

A sequence $\{a_n\}$ is *roughly monotonically increasing* (*decreasing*) in $A = (\mathbf{R}, S)$, (*S-increasing* (*S-decreasing*)), if $S(a_n) = S(a_{n+1})$ or $a_n < a_{n+1} (a_n > a_{n+1})$ and $S(a_n) \neq S(a_{n+1})$.

A sequence $\{a_n\}$ is *roughly periodic* in $A = (\mathbf{R}, S)$ (*S-periodic*), if there exists k such that $S(a_n) = S(a_{n+k})$. The number k is called the period of $\{a_n\}$.

A sequence $\{a_n\}$ is *roughly constant* in $A = (\mathbf{R}, S)$ (*S-constant*), if $S(a_n) = S(a_{n+1})$.

Suppose we are given a real function $f : X \to Y$ and discretizations $S = \{x_0, x_1, \ldots, x_n\}$ and $P = \{y_0, y_1, \ldots, y_m\}$ on X and Y respectively. If f is continuous in every $x \in S$, we with say that f is *S-continuous*. Let f be a S-continuous function, and let $N(x_i) = i$.

Function $F_f : \{n\} \to \{n\}$, such that $F_f(N(x_i)) = N(P_* f(x_i))$ will be called *rough representation* of f (or (S, P)-*representation* of f).

The function F_f can be used to define some properties of real functions.

A function f is *roughly monotonically increasing* (*decreasing*) if $F_f(i + 1) = f(i) + \alpha$, where α is a non-negative integer, (α is non-positive integer), for every $i = 0, 1, 2, \ldots n - 1$.

A function f is *roughly periodic* if there exist k such that $F_f(i) = F_f(i + k)$ for every $i = 0, 1, \ldots, n - 1$.

A function f is *roughly constant* if $F_f(i) = F_f(i+1)$, for every $i = 0, 1, \ldots, n-1$.

Many other basic concepts concerning functions can be expressed also in the rough function setting.

By the *P-lower approximation* of f we understand the function $f_* : X \to Y$ such that

$$f_*(x) = P_*(f(x)), \text{for every} x \in X.$$

Similarly the *P-upper approximation* of f is defined as

$$f^*(x) = P^*(f(x)), \text{for every} x \in X.$$

We say that a function f is *exact* in x iff $f_*(x) = f^*(x)$; otherwise the function f is *inexact* (*rough*) in x. The number $f^*(x) - f_*(x)$ is the *error of approximation* of f in x.

Finally in many applications we need the fix-point properties of functions.

We say that $x \in S$ is a *rough fix-point* (*rough equilibrium point*) of a real function f if $F_f(N(x)) = N(P_*(f(x))$.

Now we give a definition of a very important concept, the rough continuity of real function.

Suppose we are given a real function $f : X \to Y$, where both X and Y are sets of reals and S, P are discretizations of X and Y respectively.

A function f is (*roughly continuous*) (*S,P*)-*continuous* in x if

$$S(x) \subseteq P(f(x)).$$

In other words a function f is roughly continuous in x iff for every $y \in S(x)$, $f(y) \in P(f(x))$.

The intuitive meaning of this definition is obvious. Whether the function is roughly continuous or not depends on the information we have about the function, i.e. it depends on how exactly we "see" the function through the discretization of X and Y.

Obviously a function f is roughly continuous iff $F_f(i+1) = F_f(i) + \alpha$, where $\alpha \in \{-1, 0, +1\}$ for every $i = 0, 1, \ldots, n-1\}$.

Remark

Particularly interesting is the relationship between dependency of attributes in information systems and the rough continuity of functions

Let $\mathbf{S} = (U, A)$, be an *information system*, (Pawlak, 1991), where U is a finite set of *objects*, called the *universe* and A is a finite set of attributes. With every attribute $a \in A$ a set of *values* of attribute a, called *domain* of a, is associated and is denoted by V_a . Every attribute $a \in A$ can be seen as a function $a : U \to V$, which to every object $x \in U$ assigns a value of the attribute a. Any subset of attributes $B \subseteq A$ determines the equivalence relation $IND(B) = \{x, y \in U : a(x) = a(y) \text{for every} a \in A\}$. Let $B, C \subseteq A$. We will say that the set of attributes C *depends* on the set of attributes B, in symbols $B \to C$, iff $IND(B) \subseteq IND(C)$. If $B \to C$ then there exists a *dependency function* $f_{B,C} : V_{b_1} \times V_{b_2} \times \ldots \times V_{b_n} \to V_{c_1} \times V_{c_2} \times \ldots \times V_{c_m}$, such that $f_{B,C}(v_1, v_2, \ldots, v_n) = (w_1, w_2, \ldots, w_m)$, iff $\sigma(v_1) \cap \sigma(v_2) \cap, \ldots, \cap \sigma(v_n) \subseteq$

$\sigma(w_1) \cap \sigma(w_2) \cap, \ldots, \cap \sigma(w_m)$, where $v_1 \in V_b, w_j \in V_{c_j}, \sigma(v) = \{x \in U : a(v) = x\}$ and $v \in V_a$. The dependency function $B \to C$, where $B = \{b_1, b_2, \ldots, b_n\}$ and $C = \{c_1, c_2, \ldots, c_m\}$ assigns uniquely to every n-tuple of values of attributes from B the m-tuple of values of attributes from C.

There exists the following important relationship. $B \to C$ iff $f_{B,C}$ is (B,C)-roughly continuous. □

5 CONLUSIONS

Rough function concept is meant to be used as a theoretical basis for rough controllers. Basic definitions concerning rough functions were given and some basic properties of these functions investigated.

Applications of the above discussed ideas will be presented in the forthcoming papers.

Acknowledgements

This work was supported by grant No. 8 S503 021 06 from State Committee for Scientific Research

REFERENCES

[1] **Czogala, E., Mrozek, A. and Pawlak, Z. (1994).** Rough-Fuzzy Controllers. *ICS WUT Reports*, 32/94.

[2] **Hsu, C.S. (1980).** A Theory of Cell-to-Cell Mapping Dynamical Systems.*ASME Journal of Applied Mechanics,* Vol. 47, pp. 931-939.

[3] **Mrozek, A. and Plonka, L. (1994).** Rough Sets for Controller Synthesis. *ICS WUT Report* 51/94.

[4] **Orlowska, E., and Pawlak, Z. (1984)**. Measurement and Indiscernibility. *Bull. PAS, Math. Ser.* Vol. 32, No. 9-10, pp. 617-624.

[5] **Pawlak, Z. (1991).** Rough Sets - TheoreticaL Aspects of Reasoning about Data. KLUWER ACADEMIC PUBLISHERS.

[6] **Pawlak, Z. and Skowron, A. (1993).** Rough Membership Functions. In: Yaeger, R.R., Fedrizzi, M., and Kacprzyk, J. (Eds.), *Advances in the Dempster Shafer Theory of Evidence,* John Wiley and Sons, pp. 251-271.

[7] **Pawlak, Z. (1987).** Rough Functions. *Bull. PAS, Tech. Ser. Vol.* 35, No.5-6, pp. 249-251.

[8] **Robinson, A. (1970).** Non-Standard Analysis. NORTH-HOLLNAD PUBLISHING COMPANY.

9

A FUSION OF ROUGH SETS, MODIFIED ROUGH SETS, AND GENETIC ALGORITHMS FOR HYBRID DIAGNOSTIC SYSTEMS

Ray R. Hashemi* **, Bruce A. Pearce*, Ramin B. Arani*, Willam G. Hinson*, and Merle G. Paule*

** National Center for Toxicological Research, Jefferson AR. 75079*
*** Department of Computer and Information Science*
University of Arkansas at Little Rock, Little Rock, AR. 72204
USA

ABSTRACT

A hybrid classification system is a system composed of several intelligent techniques such that the inherent limitations of one individual technique be compensated for by the strengths of another technique. In this paper, we investigate the outline of a hybrid diagnostic system for Attention Deficit Disorder (ADD) in children. This system uses Rough Sets (RS) and Modified Rough Sets (MRS) to induce rules from examples and then uses our modified genetic algorithms to globalize the rules. Also, the classification capability of this hybrid system was compared with the behavior of (a) another hybrid classification system using RS, MRS, and the "dropping condition" approach, (b) the Interactive Dichotomizer 3 (ID3) approach, and (c) a basic genetic algorithm.

The results revealed that the global rules generated by the hybrid system are more effective in classification of the testing dataset than the rules generated by the above approaches.

1 INTRODUCTION

A diagnostic system is a classification system that is trained to classify a given record. The intelligence of a trained system may materialize in the form of weights (neural networks and statistical models) or a set of rules (Rule-based

and Fuzzy rule-based systems). The neural net and statistical classifiers do not provide a back tracking facility for a given classification and they fail to provide a set of rules that is meaningful to the user [8, 22]. In rule-based systems, rules have to be manually specified by a domain expert [6]. This process is expensive and lengthy. Several classification systems have been built based on Rough Sets (RS) [7, 19, 20], Modified Rough Sets (MRS) [9, 10], and ID3 [21] approaches that do not need a domain expert for rule specification. These systems learn from examples by induction [21, 12] and then use their intelligence to classify a new record.

We have shown previously that classification systems based on Rough Sets (RS) and Modified Rough Sets (MRS) are effective tools [9 10]. One of the limitations of such systems is that RS and MRS deliver only induced "local" rules. This is a limitation inherent to RS and MRS techniques. To overcome this limitation of RS and MRS a hybrid classification system is needed.

A hybrid classification system is a system composed of several intelligent techniques such that the inherent limitations of one individual technique be compensated for by the strengths of another technique. In this paper, we investigate the outline of a hybrid diagnostic system for Attention Deficit Disorder (ADD) in children. This system uses RS and MRS to induce rules from examples and then uses genetic algorithms to globalize the rules. Also, the classification capability of this hybrid system will be compared with the behavior of (a) another hybrid classification system using RS, MRS, and the "dropping condition" approach [10] and (b) the ID3 approach.

2 METHODS

In this section, the Rough sets (RS), Modified Rough Sets (MRS), Modified Dropping Conditions, and Modified Genetic Algorithms are explained in detail.

2.1 Rough Sets (RS).

The mathematical foundation of RS will be discussed briefly and then its application will be discussed in detail by providing an example. For a comprehensive discussion of RS refer to (19).

Mathematical Foundation of RS.

A rough set is a mathematical model used to deal with an approximate classification of objects.

Definition 1 *An approximate space P is an ordered pair $P(U, R)$ where U is a set called universe and R is a binary equivalence relationship over U. Relation R is called an indiscernibility relation. Through the rest of these definitions P, U, and R notations have the same meanings.*

Definition 2 *If $a \in U$, then $[a]_R$ is an equivalence class of R.*

Definition 3 *An elementary set (E) in P is an equivalence class of R or the empty set $\emptyset$.*

Definition 4 *Any finite union of elementary sets is called a definable set in P.*

Definition 5 *Let $A \subseteq U$. The set $(A, R) = \{(a, r) | a \in A, r \in R\}$ where R is a binary equivalence relationship over U that is called a rough set inP.*

Definition 6 *If $A \subseteq U$, then $Upper(A)$, called upper approximation of A in P, is defined by $Upper(A) = \{a \in U | [a]_R \cap A \not\subseteq \emptyset\}$ and $Lower(A)$, called lower approximation of A in P, is defined by $Lower(A) = \{a \in U | [a]_R \subseteq A\}$.*

Definition 7 *The set $M_p(A) = Upper(A) - Lower(A)$ is called a boundary of A in P. This is the space between the lower approximation of A in P, and the upper approximation of A in P. The boundary of A in P refers to those elements of the universe that are only partially in the space A.*

Definition 8 *An information system, S, as defined by Pawlak (18), is a quadruple (U, Q, V, δ) in which U is a non-empty finite set of objects, b; Q is a finite set of attributes, q; $V = \cup_{q \in Q} V_q$, and V_q is the domain of attribute q. δ is a mapping function such that $\delta(b, q) \in V_q$ for every $q \in Q$ and $b \in U$. Q is composed of two parts: a set of condition attributes (C) and a decision attribute (D).*

Application of RS.

In reference to the definition 8 (above), the attraction of using rough sets lies in their ability to define and manipulate an information system for the purpose of establishing the relationship between condition values (C) and the decision value (D) in the form of rules. These rules may be used to predict a decision value for a new object for which only the condition values are known. The following example will describe the four components of an information system and it will explain the entire process of extracting rules from an information system using RS. This example will then be used to explain MRS and its application.

example 1 Table 1 is an information system. The universe U is composed of nine objects, $U = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9\}$. Each object in U represents a child (test subject, and ai may be considered as a subject identification number). In this system, condition attributes are $C = \{M_1, M_2\}$ (M_i is a measure of the subject's operant behavior). The decision attribute is the test subject's IQ (i.e. $D = \{IQ\}$). Here, the domain of attributes, $V = \{1, 2, 3\}$. The mapping schema (δ) is defined as follows. For the M_1 attribute, 1 = "pass" and 2 = "fail"; for the M2 attribute, 1 = "poor", 2 = "average", and 3 = "good"; for the Decision attribute IQ, 1 = "above average", 2 = "average or below". To use RS and MRS for the purpose of extracting rules from the information system of Table 1, we use algorithm ONE below (a formal discussion of the process, may be found in [6, 9, 12]).

ALGORITHM ONE.

Step 1- Reduce the information system vertically and horizontally (system reduction).

Step 2- Generate partitionings and classifications.

Step 3- Generate lower and upper approximation spaces.

Step 4- Extract local rules (certain, possible, and approximate).

Step 5- End.

System Reduction. The information system is reduced by removing all redundant objects (rows) and condition attributes (columns) from it (i.e. the

U	Q		
	C		D
	M_1	M_2	IQ
a_1	1	3	1
a_2	2	1	2
a_3	1	1	2
a_4	2	2	1
a_5	1	3	1
a_6	2	2	2
a_7	2	1	2
a_8	2	1	1
a_9	1	2	2

Table 1 An Information System

information system is reduced vertically and horizontally). In *vertical reduction*, all the rows that are identical, considering their values in the Q attributes, are collapsed into one row. Here, rows 1 and 5 are identical in their values for the Q attributes and they are collapsed into one row. Also, rows 2 and 7 are identical in their values for Q attributes and are collapsed into one row, Table 2. The vertical reduction has generated seven sets out of the initial universe of nine objects. These sets are $z_1 = \{a_1, a_5\}$, $z_2 = \{a_2, a_7\}$, $z_3 = \{a_3\}$, $z_4 = \{a_4\}$, $z_5 = \{a_6\}$, $z_6 = \{a_8\}$, and $z_7 = \{a_9\}$ and they are called *Q-elementary* sets. This means that rows for members of any given Q-elementary set are identical and rows for all sets in reference to their values in Q attributes are unique.

In *horizontal reduction*, we determine whether the decision (D) depends on all the condition attributes (C) or a subset of C, like C'. If C' exists, then C' replaces C and it is called the *reduct* of C. A given information system, may have more than one reduct.

To find the reducts of C, if any exist, for the information system of Table 2 we use the following steps.

a: All the possible subsets of the condition attributes are built. These subsets are $C_1 = \{M_1\}$, and $C_2 = \{M_2\}$.

U	Q		
	C		D
	M_1	M_2	IQ
$z_1 = \{a_1, a_5\}$	1	3	1
$z_2 = \{a_2, a_7\}$	2	1	2
$z_3 = \{a_3\}$	1	1	2
$z_4 = \{a_4\}$	2	2	1
$z_5 = \{a_6\}$	2	2	2
$z_6 = \{a_8\}$	2	1	1
$z_7 = \{a_9\}$	1	2	2

Table 2 Vertical Reduction (Q-elementary sets) of the Information System described by Table 1.

b: Considering only the condition attributes in C_1, the objects of Table 2 are organized into two sets called C1-elementary sets (Table 3-a). Likewise, three sets called C_2-elementary sets (Table 3-b) for the universe of objects in Table 2.

c: If the C_1-elementary sets or C_2-elementary sets are the same as the Q-elementary sets, then the condition attributes $C = \{M_1, M_2\}$ are replaced by subset $C_1 = \{M_1\}$ or $C_2 = \{M_2\}$, respectively. If both elementary sets are the same as the Q-elementary sets then there are two reducts for the condition attributes C and one is chosen either arbitrarily or by expert consultation.

Tables 3-a and 3-b, reveal that neither C_1-elementary sets nor C_2-elementary sets are the same as the Q-elementary sets. Thus, none of the condition attributes in C are redundant and Table 2 is the final outcome of the reduction step of Algorithm One. The rest of the steps of Algorithm One will be applied to the information system of Table 2.

Partitioning. All objects in a reduced information system that have the same decision value (d_i) make one *partitioning* (L_i). Therefore, partitionings L_1 and L_2 include all the objects for which the decision values are 1 and 2, respectively: $L_1 = \{z_1, z_4, z_6\}$ and $L_2 = \{z_2, z_3, z_5, z_7\}$.

U	C_1 / M_1
$\{z_1, z_3, z_7\}$	1
$\{z_2, z_4, z_5, z_6\}$	2

(a)

U	C_2 / M_2
$\{z_2, z_3, z_6\}$	1
$\{z_4, z_5, z_7\}$	2
$\{z_1\}$	3

(b)

Table 3 Testing the Horizontal reduction of the information system of Table 2. (a) the C_1-elementary sets. (b) the C_2-elementary sets.

Classification. All objects in a reduced information system that have identical values for their condition attributes make one *classification* (R_j). The objects z_2 and z_6 have the identical values for their condition attributes of $C = \{M_1, M_2\}$. Thus z_2 and z_6 makes one classification. For the information system of Table 2 we have five classifications of $R_1 = \{z_1\}$, $R_2 = \{z_2, z_6\}$, $R_3 = \{z_3\}$, $R_4 = \{z_4, z_5\}$, and $R_5 = \{z_7\}$.

Lower approximation. If the objects of a given classification, R_i, are totally contained in a given partitioning, L_j, then the objects of R_i are included in the *lower approximation* of L_j. The classification R_1 has only one object, z_1. This object is in the partitioning L_1. Thus, z_1 is included in the lower approximation of L_1. Also, the objects of classifications R_3 and R_5 are totally contained in the partitioning L_2. As a result, the lower approximation for L_1 and L_2 are: $Lower(L_1) = \{z_1\}$ and $Lower(L_2) = \{z_3, z_7\}$.

Upper approximation. If objects in classification R_i are contained partially in L_j, then the objects of R_i are in the *upper approximation* of L_j. The objects in classification R_2 are z_2 and z_6. Only z_6 is contained in L_1. Thus, the objects of the classification R_2 are included in the upper approximation of L_1. The upper approximation for L_1 and L_2 are: $Upper(L_1) = \{z_4, z_5, z_2, z_6\}$ and $Upper(L_2) = \{z_2, z_6, z_4, z_5\}$.

Rule Extraction. *Local certain rules* for decisions 1 and 2 are extracted from the objects in $Lower(L_1)$ and $Lower(L_2)$ respectively. $Lower(L_1)$ contains

only one object, z_1. The condition for z_1 are $M_1 = 1$ and $M_2 = 3$ (Table 2). The local certain rules extracted from z_1 is:

(i): IF $(M_1 = 1 \wedge M_2 = 3) \rightarrow IQ = 1$.

$Lower(L_2)$ contains two objects, z_3 and z_7. The local certain rules generated from these objects are:

(ii): IF $(M_1 = 1 \wedge M_2 = 1) \rightarrow IQ = 2$.

(iii): IF $(M_1 = 1 \wedge M_2 = 2) \rightarrow IQ = 2$.

Rules (ii) and (iii) may be combined as follow:

$$\text{IF } (M_1 = 1 \wedge M_2 = 1) \vee (M_1 = 1 \wedge M_2 = 2) \rightarrow IQ = 2.$$

Local possible rules for decisions 1 and 2 are extracted from objects in $Upper(L_1)$ and $Upper(L_2)$, respectively. The same process that was described for extracting local certain rules is used and the local possible rules are:

(i): IF $(M_1 = 2 \wedge M_2 = 2) \vee (M_1 = 2 \wedge M_2 = 1) \rightarrow IQ = 1$.

(ii): IF $(M_1 = 2 \wedge M_2 = 1) \vee (M_1 = 2 \wedge M_2 = 2) \rightarrow IQ = 2$.

Thus, if an object satisfies local possible rule (i), then the IQ for the object is possibly 1.

2.2 Modified Rough Sets (MRS).

Objects in the lower approximation spaces contribute to the local certain and global certain rules. In contrast, the objects in the upper approximation spaces

only contribute to the local and global possible rules. As a result, the global certain rules are extracted from a subset of the objects of the universe and, based on our total number of observations, this subset is not large. It may be argued that the objects which are not contributing to the local and global certain rules still must not be ignored. The following two facts strongly support this argument:

1- Often the number of data points collected for a test is so small that we do not have the luxury of ignoring any of them.

2- The objects within a given upper approximation space (e.g. β) have a decision conflict which is the reason that they are not included in a lower approximation space. Often in β, only a few objects have a decision that is different from that of the majority of objects. In such a case we cannot ignore all of the objects in β because of a few records that exhibit a decision conflict. The best solution to such a situation might be to change the decisions for those few records to the dominant decision in β. Doing so makes the dominant decision an it approximate decision. The details of conflict resolution may be found in [12].

The Modified Rough Sets approach is a version of the Rough Sets approach in which the approximate decision concept has been exercised. Such exercise eliminates all the upper approximation spaces. In fact, the upper and lower approximations of a given decision realm are mapped on the same boundary.

To explain the Modified Rough Sets further, the concepts of *approximate decision* and *approximate rele*is introduced and applied in the example of the information system shown in Table 2.

Approximate decision. An information system, S, is given. The object space in S is divided into n classifications of $R_1, \cdots, R_n$ based on the condition values and m partitionings of $L_1, \cdots, L_m$ based on decision values of $d_1, \cdots, d_m$. Let the objects in classification R_i have decision conflicts (i.e., have different decision values). As a result of this conflict, the objects in R_i will not be included

in the lower approximation space of any decision realm. A goal of the approximate decision is to resolve the conflict of decisions in R_i. To meet this goal, let $P[R_i|j]$ be the probability of observing classification R_i, given the partitioning L_j; and let Q_j be the probability of observing the partitioning L_j in S. The probability of observing the partitioning L_j, given the classification R_i, based on Bayes theorem (5), is then:

$$P[j|R_i] = \frac{P[R_i|j]\ Q_j}{P[R_i|1]\ Q_1 + ... + P[R_i|m]\ Q_m} \tag{1.1}$$

We refer to $P[R_i|j]$ as the weight for decision j in R_i and it is shown as ${W_j}^i$. The approximate decision for the objects in R_i is defined as $d_j = \max\{P[j|R_i], j = 1, \cdots, m\}$. If there is not a unique maximum value then selecting the approximate decision among the eligible decisions is left to the user (expert).

For our example, the classifications $R_2 = \{z_2, z_6\}$ and $R_4 = \{z_4, z_5\}$ have a decision conflict among their objects(e.g. in R_2, the decision value of z_2 and z_6 are 2 and 1, respectively). The above approach for determining the approximate decisions for R_2 and R_4 is now applied. For easy reference, the two partitionings of our example are repeated here: $L_1 = \{z_1, z_4, z_6\}$ and $L_2 = \{z_2, z_3, z_5, z_7\}$ (the decisions for all the objects of L_1 and L_2 are equal to 1 and 2, respectively). For the calculation of the dominant decision in R_2 and R_4, remember that the z objects must be replaced by the actual a objects. This means that R_2, R_4, L_1 and L_2 are as follows: $R_2 = \{a_2, a_7, a_8\}$, $R_4 = \{a_4, a_6\}$, $L_1 = \{a_1, a_5, a_4, a_8\}$, and $L_2 = \{a_2, a_7, a_3, a_6, a_9\}$,

The probability of observing the partitioning L_1 and L_2 in Table 1 (the original information system) are $Q_1 = 4/9$, $Q_2 = 5/9$ (in Table 1, four out of 9 objects have the decision value of 1 and five objects have the decision value of 2). For R_2, $P[R_2|1] = 1/4$ (only one object, a_8, of classification R_2 is among the four objects of partitioning L_1), and $P[R_2|2] = 2/5$ (only two objects, a_2 and a_7, of classification R_2 is among the five objects of partitioning L_2). Based on the formula (1, above), $P[1|R_2] = (1/4 * 4/9)/(1/4 * 4/9 + 2/5 * 5/9) = 1/3$ and $P[2|R_2] = (2/5 * 5/9)/(1/4 * 4/9 + 2/5 * 5/9) = 2/3$. The decision weights in R_2 are $1/3$ and $2/3$ for decisions 1 and 2, respectively. Therefore, the approximate decision in R_2 is 2.

For R_4, $Q_1 = 4/9$, $Q_2 = 5/9$, $P[R_4|1] = 1/4$, and $P[R_4|2] = 1/5$. The $P[1|R_4] = 1/2$ and $P[2|R_4] = 1/2$. Since both decisions 1 and 2 have the same weight, the approximate decision in R_4 is 2. This means that the IQ value (decision value) for objects in R_2 are changed to 2 and the objects in R_4 are changed to 2 in Table 2-b. As a result, we have a new information system, Table 4-a.

Approximate rules. The local "certain" rules extracted from Table 4-a are no longer referred to as local certain rules but instead are called *local approximate rules* , because in their extraction approximate decisions are involved. To each approximate rule a weight *rule weight* is associated which is equal to the decision weight of the rule's decision. To extract these rules, the algorithm ONE is applied on the Table 4-a. Table 4-b shows the vertical and horizontal reduction of the new system. The set of approximate rules along with their rule weights are as follows:

(i): IF $(M_1 = 1 \wedge M_2 = 3) \rightarrow IQ = 1$ *and* $w_1 = 1$.

(ii): IF $(M_1 = 2 \wedge M_2 = 1) \rightarrow IQ = 2$ *and* $w_2 = 2/3$.

(iii): IF $(M_1 = 1 \wedge M_2 = 1) \rightarrow IQ = 2$ *and* $w_2 = 1$.

(iv): IF $(M_1 = 2 \wedge M_2 = 2) \rightarrow IQ = 2$ *and* $w_2 = 0.5$.

(v): IF $(M_1 = 1 \wedge M_2 = 2) \rightarrow IQ = 2$ *and* $w_2 = 1$.

Mapping the lower and upper approximations of decision realms on the same boundary serves another purpose. It must be remembered that when using a statistical model in prediction it is assumed that a decision realm has a distinct boundary. However, within the decision realm one can find objects which have been imposed on the realm because of some threshold satisfaction. In contrast, the RS approach separates those objects which are totally and partially in a decision realm by the use of lower and upper approximations of the decision. Because of this difference, the comparison of predictions resulting from a statistical model with those obtained from a RS approach for the same set of objects are not comparable unless for RS we map both the boundaries of lower and upper approximations of each decision realm over a common boundary as is done using Modified Rough Sets. In other words, the results obtained from

U	Q		
	C		D
	M_1	M_2	IQ
$z_1 = \{a_1, a_5\}$	1	3	1
$z_2 = \{a_2, a_7\}$	2	1	2
$z_3 = \{a_3\}$	1	1	2
$z_4 = \{a_4\}$	2	2	2
$z_5 = \{a_6\}$	2	2	2
$z_6 = \{a_8\}$	2	1	2
$z_7 = \{a_9\}$	1	2	2

(a)

U	Q		
	C		D
	M_1	M_2	IQ
$w_1 = \{a_1, a_5\}$	1	3	1
$w_2 = \{a_2, a_7, a_8\}$	2	1	2
$w_3 = \{a_3\}$	1	1	2
$w_4 = \{a_4, a_6\}$	2	2	2
$w_5 = \{a_9\}$	1	2	2

(b)

Table 4 The modified information system for Table 2. (a) the modified information system. (b) the result of verticaland horizontal reduction

Modified Rough Sets are comparable to the results obtained from statistical models.

As it was explained, Rs and MRS deliver only the local rules. Often the local rules are not practical and they need to be globalized. we discuss two globalization techniqes in the following subsection.

2.3 Globalization of Local Rules.

Local rules may be globalized using a "modified dropping conditions" technique [14] or a "modified genetic algorithm" proper for use with RS and MRS [11].

A Modified Dropping Condition Approach.

In this approach, we keep a subset of conditions in a given rule that preserves the uniqness of that rule among the set of local rules and the rest of conditions of the given rule are dropped. The global certain and possible rules are generated as follows:

A- Make a table out of the condition values of the above rules (local certain and local possible rules), Table 5-a. The rows in this table come from objects z_1, z_3, z_7, z_4, z_6, z_2, and z_5 of Table 2. Objects z_1, z_3, and z_7 belong to, at most, one lower approximation space and the rest of the objects belong to at least one upper approximation space.

B- All the duplicated rows among those that belong to either local certain or local possible rules are collapsed into one row, Table 5.b.

C- For each row of Table 5.b, keep a subset of the condition values that are unique throughout the entire table and drop the rest of the condition values for the record (the dropped values are replaced by an asterisk). The value 3 in the

	M_1	M_2
Condition values for local certain rules	1	3
	1	1
	1	2
	2	2
Condition values for local possible rules	2	1
	2	2
	2	1

(a)

	M_1	M_2
Condition values for local certain rules	1	3
	1	1
	1	2
Condition values for local possible rules	2	2
	2	1

(b)

	M_1	M_2
Condition values for local certain rules	*	3
	1	1
	1	2
Condition values for local possible rules	2	2
	2	1

(c)

	M_1	M_2
Condition values for local certain rules	*	3
	1	1
	1	2

(d)

	M_1	M_2
Condition values for local possible rules	2	2
	2	1

(e)

Table 5 Application of modefied Dropping Conditions approach. (a) Initial Table, (b) Duplicate rows removed, (c) intermediate Table, the dropped condition from Table 5-a shown by (*), (d) the Final Table for extarcting the global certain rules, and (e) the Final table for extracting the global possible rules.

condition attribute of M_2 for the first record of Table 5-b, record (r_1), is able to uniquely identify this record throughout the entire table. In other words, there is no value of 3 in any of the records of this table, except r_1. Therefore, we keep this condition value and drop the rest of the condition values in r1, as shown in Table 5-c.

D- Split Table 5-c into two tables (Table 5-d and Table 5-e). Table 5-d and Table 5-e contain all the condition values for local certain and local possible rules, respectively. This means that the rows in tables 5-d and 5-e belong to objects of lower approximation and upper approximation spaces, respectively. The extracted rules from the rows of Table 5-d are:

(i): IF $(M_2 = 3) \rightarrow IQ = 1$.

(ii): IF $(M_1 = 1 \wedge M_2 = 1) \vee (M_1 = 1 \wedge M_2 = 2) \rightarrow IQ = 2$.

The above list defines the global certain rules.

The extracted rules from the rows of Table 5-e define the global possible rules and they are:

(iii): IF $(M_1 = 2 \wedge M_2 = 2) \vee (M_1 = 2 \wedge M_2 = 1) \rightarrow IQ = 2$.

(iv): IF $(M_1 = 2 \wedge M_2 = 2) \vee (M_1 = 2 \wedge M_2 = 1) \rightarrow IQ = 1$.

The local and global possible rules are not of interest and will not be pursued further in this study. The reason for such a decision stems from the fact that for rules (iii) and (iv) the conditions are the same but the decisions are different. That is, IQ for a new object for which $M_1 = 2$ AND $M_2 = 2$ will be possibly 2 and possibly 1 (not a very definitive decision). The local approximate rules may be globalized as it was described above.

Each global certain rule has three properties:

Prop. 1: A global rule correctly classifies at least one object of the lower approximation space.

Prop. 2: Each object of the lower approximation space is correctly classified by at least one global rule.

Prop. 3: A global rule correctly classifies at zero or more objects of the upper approximation space.

The problem with the "dropping condition" globalizing technique is that it only provides for horizontal reduction of the set of local rules. We will remove this shortcoming by using a modified Genetic Algorithm.

2.4 A Basic Genetic Algorithm

A genetic algorithm simulates the evolutionary process of a set of "genomes" over time. Genome is a biological term that refers to a set of "genes" and gene is the basic building block of any living entity. For our use here, "genome" is represented by a rule and "gene" is represented by a condition within a rule. A genetic algorithm starts with a set of genomes created randomly (a generation) and then the evolutionary process of the "survival of the fittest" genomes takes place. The un-fit genomes are removed and the remaining genomes reproduce a set of new genomes. Reproduction of the genomes is accomplished by applying the simulation of the two well known genetic processes: mutation and crossover. The new genomes created by mutation and crossover along with genomes that survived from the previous generation constitute a new generation of genomes. This process is repeated and in each repetition a fitter generation is created. Because each generation is built with information derived from the previous generation of genomes, it is said that each generation evolves with time.

Our goal in developing a genetic algorithm was to meet two objectives. First, the algorithm should be appropriate for use in conjunction with RS and MRS. Second, the algorithm should be able to deliver a set of global rules that result from vertical and horizontal reductions of a given local rule set.

To explain the details of the algorithm used in this study we need to define the fitness function, mutation process, and crossover process. To do so, Let S be a universe of objects. An object of this universe may be denoted as a pair (C_i, d_i) in which C_i is a set of conditions with K members (K independent variables, $c_1^{(i)}, \cdots, c_k^{(i)}$) and d_i is a decision (one dependent variable). It is assumed that

the conditions set will determine the value of the decision ($c_1^{(i)}, \cdots, c_k^{(i)} \rightarrow d_1^{(i)}$). The domain of the condition $c_j (1 \leq j \leq k)$ in the universe S is the set $\{a_{j1}, \cdots, a_{jq}\}$. This means that, a_{jm} is a possible discrete value for the condition c_j in a given object.

Fitness Function. At this stage we need to address the problem of assessing the fitness of each generation based on a set of conditions and their corresponding decisions. First, consider the following notations:

$$\begin{aligned} A_S(C_i, d_i) &= \{(C_i', d_i') \in S : C_i' = C_i\}, \\ B_S(C_i, d_i) &= \{(C_i', d_i') \in S : (C_i', d_i') = (C_i, d_i)\}. \end{aligned}$$

The empirical estimate of the conditional probability of decision d, given the condition set C, is given by,

$$P_c(d) = \frac{card\,(B_S(C,d))}{card\,(A_S(C,d))} = \frac{N_d}{N_c} \tag{1.2}$$

One might consider $P_c(d)$ to assess the usefulness of conditions in determining the decision, d. Also, based on the above probability, different so-called fitness functions can be proposed. Two such functions were discussed by Packard (14):

$$F_1(c) = \sum_d P_c(d) \log \frac{P_c(d)}{P_m(d)} - \frac{\alpha}{N_c} \tag{1.3}$$

$$F_2(c) = \sum_d P_c(d) \log \frac{P_c(d)}{P_0(d)} - \frac{\alpha}{N_c} \tag{1.4}$$

Where $P_m(d)$, $P_0(d)$ and α/N_c refer to the maximum entropy distribution , empirical distribution of d, and a bias correcting factor, respectively. Note that α/N_c is used to reduce the bias introduced by sampling from a finite population. In practice, α is adjusted until an acceptable, i.e. shorter, confidence interval for the fitness function is obtained. In the information theory nomenclature, F_1 and F_2 refer to "relative entropy" or "Kullback Leibler distance". The details may be found in [1, 2].

In order to measure the fitness at each (C_i, d_i), we propose the conditional probability (1.1) corrected for bias.

$$F(C,d) = P_c(d) - \frac{\alpha}{N_c} \tag{1.5}$$

It is observed that the application of the RS approach divides the universe of objects S into mutually exclusive sets of lower, L, and upper, U, approximate spaces. Also, it is true that the data for the objects of L space is less noisy than the data for the objects of U space. This makes the objects of the lower approximation more valuable. Thus, we need to modify the fitness function to be responsive to this fact. To do so, the conditional probability (1.1) needs to be modified to reflect the extra confidence placed on the elements of L, thus

$$P_c{}'(d) = P(L)P_c^{(L)}(d) + \theta * \left[P(U)P_c^{(U)}(d)\right]. \tag{1.6}$$

where $0 \leq \theta \geq 1$ and it is called U-confidence factor. The $P_c{}'(d)$ will be substituted in the above fitness function (1.4).

Application of the MRS approach transforms the universe S into one conflict-free space, H. Thus the genetic algorithm used in conjunction with MRS uses the conditional probability (1.1) in fitness function (1.4).

Mutation Process. Based on Packard (14), a single genome may be mutated according to following mutation rules.

1) Eliminate the condition c_i from the genome $[c_i \rightarrow *]$.

2) Change the value of condition c_i to the new value of a' $[c_i \rightarrow a']$.

3) Restate an already mutated condition using a new value of a' $[* \rightarrow a']$.

If the value for condition c_i is made of the disjunction of j values, then:

4) Expand the value list for c_i $[(a_1\ or, \cdots,\ or\ a_j) \rightarrow (a_1\ or, \cdots,\ or\ a_j, or\ a_{j+1})]$ or

5) Shorten the value list for c_i $[(a_1\ or, \cdots,\ or\ a_j) \rightarrow (a_1\ or, \cdots,\ or\ a_{j-1})]$.

To adjust the mutation process for use in RS and MRS, mutation rule 3 is partially applicable and mutation rules 4 and 5 are not applicable at all. Mutation rule 3 is partially applicable because the use of RS or MRS generates a reduct of the universe and restating those conditions removed by RS or MRS defeats the purpose of using such approaches. However, restating those condition values removed in the mutation process is acceptable.

Neither mutation rule 4 nor mutation rule 5 is applicable to the rules generated by the RS or MRS approaches because when c_i in a rule has a list of j values, it will be delivered by RS or MRS as a list of j rules which are only different in the value of condition c_i.

Crossover Process. In this process two parent genomes (two selected rules) mate and reproduce two offspring by swapping a part of their genes. For example, the two rules:

$$c_1^{(i)} = a_{11} \;\&\; c_2^{(i)} = a_{21} \;\&\; \cdots \;\&\; c_k^{(i)} = a_{k1} \rightarrow d_1 \quad \text{and}$$

$$c_1^{(j)} = a_{12} \;\&\; c_2^{(j)} = a_{22} \;\&\; \cdots \;\&\; c_k^{(j)} = a_{k2} \rightarrow d_2$$

may swap the values for $c_1^{(i)}$ and $c_1^{(j)}$ and the values for $c_2^{(i)}$ and $c_2^{(j)}$ to generate two new genomes:

$$c_1^{(i')} = a_{12} \;\&\; c_2^{(i')} = a_{22} \;\&\; \cdots \;\&\; c_k^{(i')} = a_{k1} \rightarrow d_1 \quad \text{and}$$

$$c_1^{(j')} = a_{11} \;\&\; c_2^{(j')} = a_{21} \;\&\; \cdots \;\&\; c_k^{(j')} = a_{k2} \rightarrow d_2$$

In this study we select the parents randomly from a subset of fitter genomes. Also, the number of conditions selected for swapping were chosen randomly.

The crossover process has a conflict with the RS approach: when a subset of the conditions' values of two rules are swapped, two new rules (offspring) are created such that they may not be able to correctly classify any of the objects in the lower approximation space. To relax this conflict, one tries to operate in favor of objects in space L by controlling the value of θ in equation 1.5. Also, there is a chance that resulting offspring mutate such that they become valid global rules. If a rule is among the M fitter genomes in the final generation (we refer to these rules as *genetic-fit* rules) but does not satisfy Property 1 of the global rules, then the rule will be eliminated from the set.

Note that each genetic-fit rule may classify correctly at least one object of space L, but each object of space L may not be classified correctly by any of the genetic-fit rules (violation of Property 2 of the global rules). In fact, we try to relax Property 2 for the genetic-fit rules in order to generate vertically reduced global rules. For example, if one can select a rule that classifies correctly a large number of objects in space U by sacrificing the ability to classify one rule in space L, then one would like to exercise this selection. To enforce property 2, one insures that the number of objects in space L that cannot be correctly classified by any of the genetic-fit rules does not go beyond a threshold defined by the user.

The crossover process has less conflict with the MRS approach because in MRS the L and U spaces combine into one homogeneous space which is free of conflicted objects.

Development of the Algorithm. In general, a genetic algorithm is composed of the following steps:

Step 1. Create a set of N random rules (first generation of genomes).

The following steps will create a new generation of genomes:

Step 2. Calculate the fitness of each rule.

Step 3. Sort the rules based on their fitness value in descending order.

Step 4. Keep M fitter rules ($M < N$) and eliminate the rest of the rules.

Step 5. Create the next generation by making $N - M$ rules out of M rules using Crossover and Mutation operators.

Step 6. Go to step 2.

The basic genetic algorithm is not fit for use with RS and MRS approaches. Thus, the basic genetic algorithm is modified to remove this shortcoming.

Algorithm RS-GEN. This algorithm is a modification of the basic genetic algorithm for use with RS approach.

Step 1. Apply Rough sets to divide the universe of objects S into L and U spaces(N, N_1, and N_2 are the number of objects in the universe, lower approximation space, and upper approximation space, respectively).

Step 2. Extract the Local Certain (LC) Rules from L.

Step 3. Calculate the fitness of each rule against the universe S using formula $F(C, d)$ and conditional probabilty $P_c(d)$.

Step 4. Sort the rules based on their fitness value in descending order.

Step 5. Keep M fitter rules ($M < N$) and eliminate the rest of the rules.

Step 6. Create the next generation by making $N - M$ rules out of M rules using modified Crossover and Mutation operators proper for use with RS and MRS.

Step 7. Go to step 3 (the number of repetitions is the same as the number of generations and it is determined by the user).

Instead of creating n random rules in step 1, we use the set of LC rules extracted from the lower approximation space of the universe. To justify this action, it can be argued that the set of LC rules is more coherent than a set of random rules. Also, since RS and MRS approaches are capable of generating a reduct of the universe, those conditions that are not correlated with the decisions were already removed. In other words, the mutation process is already partially completed by the application of RS and MRS approaches which in turn shortens the time needed to reach an acceptable generation of genomes. This is considered as one of the advantages of synthesizing RS or MRS with the genetic algorithm.

Note that if there is a genetic-fit rule which does not satisfy any object of space L (violation of Property 1), the rule will be removed from the set of genetic-fit rules. By doing so, one tries not to defeat the purpose of using the RS approach.

To make the RS-GEN algorithm applicable to the MRS approach, we make changes only in steps 1 , 2, and 3 and the rest of the steps remain the same.

Algorithm MRS-GEN. This algorithm is a modification of the RS-GEN algorithm for use with MRS.

Step 1. Apply Modified Rough sets to transform the universe of objects into a conflict free space H.

Step 2. Extract the Local approximate (LA) Rules from H.

Step 3. Calculate the fitness of each rule against space H using formula $F(C, d)$ and conditional probability $P_c(d)$.

Step 4. Sort the rules based on their fitness value in descending order.

Step 5. Keep M fitter rules ($M < N$) and eliminate the rest of the rules.

Step 6. Create the next generation by making $N - M$ rules out of M rules using modified Crossover and Mutation operators proper for use with RS and MRS.

Step 7. Go to step 3 (the number of repetitions is the same as the number of generations and it is determined by the user).

3 DATA ANALYSIS

A set of five behavioral tasks, as part of an Operant Test Battery (OTB) [15, 16, 17, 18], was performed by two groups of children at the Arkansas Children's hospital. There were 80 children in each group. The children in the first group had Attention Deficit Disorder (ADD) and the children in the second group had no known problem with brain function and were used as a control group. Briefly, the OTB tasks were identified as Conditioned Position Responding (CPR), Progressive Ratio (PR), Incremental Repeated Acquisition (IRA), Delayed Matching-to-Sample (DMTS), and Temporal Response Differentiation (TRD). We measured Accuracy (ACC), Response Rate (RR) and Percentage of Task Completed task (PTC) for CPR, IRA, DMTS, and TRD. Also, we measured the Break Point (BREAK), RR, and PTC for PR.

In addition, we measured Choice Response Latency (CRL) and Observing Response Latency (ORL) for CPR, Average Hold (AVG-HLD) for TRD, CRL for IRA, and Observing Response Latency (ORL) and Increment Choice Response Latency (ICRL) for DMTS.

Moreover, for each child we collected AGE, SEX, and IQ. Collectively, each child had 24 conditions (independent variables) and one decision (dependent variable). The decision field was the actual diagnosis of the child (ADD or not ADD).

The values for all conditions except AGE, SEX, and IQ were discretized into three categories of "1" (low), "2" (average), and "3" (high) using the Equal

Interval Width method [5]. A preliminary study of the data revealed that for the older ADD children, measures of OTB task performance are very close to those of their counterparts in the control group. Thus, data for 14 and 15 year old children were removed from the dataset (a total of 3 records). Since we had only 4 children age 5, we also removed their records from the dataset. For the rest of the children, the condition AGE was discretized as follows: "1" (AGE range $6-7$), "2" (AGE range $8-9$), "3" (AGE range $10-11$), and "4" (AGE range $12-13$). The condition SEX had two categories of "1" (male) and "2" (female). The condition IQ was discretized into four categories of "1" (very low; IQ range $50-70$), "2" (low; IQ range $71-90$), "3" (average; IQ range $91-110$), and "4" (above average; IQ range $(111-130)$. Since we had only 3 children with IQ = "1" (very low), we also removed their records from the dataset.

The resulting dataset was composed of 150 records and each record had 24 condition fields and one decision field. For the rest of the paper we refer to this dataset as the original dataset.

4 RESULTS

We applied the RS-GEN and MRS-GEN algorithms separately to the original dataset for 300 repetitions. In each repetition we kept 15 fitter rules. Two sets of genetic-fit rules were created in this process and they were called GLOBAL-RS-GEN and GLOBAL-MRS-GEN.

To check the validity of the sets of genetic-fit rules, we created a testing dataset from the original dataset using the statistical approach of *random resampling* [3, 4]. Our testing dataset had 48 records. We applied GLOBAL-RS-GEN and GLOBAL-MRS- GEN genetic-fit rules on the testing dataset and the results are shown in Table 1.

Separately, for the same original dataset, we applied the "dropping condition" approach to globalize the local rules extracted from the same original dataset by RS and MRS. The two new rule sets were called GLOBAL-RS-DROP and GLOBAL-MRS- DROP. We applied these two new rule sets to the same testing dataset and the results of classifying the testing set are illustrated in Table 1.

RULE SETS	# of Correctly Classified Records of the Testing Dataset
GLOBAL-RS-GEN	40 (84 %)
GLOBAL-MRS-GEN	45 (94 %)
GLOBAL-RS-DROP	34 (71 %)
GLOBAL-MRS-DROP	37 (77 %)
ID3	33 (69 %)
BASIC-GEN	28 (54 %)

Table 6 Number of correctly classified records of the testing dataset using different techniques.

We also trained an ID3 classification system with the original dataset and tested the system against the same testing dataset. The findings are also shown in Table 1.

In addition, to compare the quality of the rule sets generated by RS-GEN and MRS-GEN algorithms, we applied the basic genetic algorithm on a set of randomly generated rules. The rules were evolved for 300 generations and the end result set of rules, called BASIC-GEN, were tested against the same training set. These findings are also shown in Table 1.

5 CONCLUSION

The results presented in Table 1 reveal that the synthesis of the genetic algorithm and RS or genetic algorithm with MRS is an effective tool in classification of data. Based on this study, it seems that the global rules generated by the hybrid system are more effective in classification than the rules generated by the basic genetic algorithm, dropping condition, and ID3 approaches. We feel this hybrid system has a great potential to be used in many types of classifications.

Acknowledgements

The first author (RRH) and the third author (RBA) had appointments to the Oak Ridge Institute for Science and Education (ORISE) at the National Center for Toxicological Research at the time that this paper was prepared.

REFERENCES

[1] Catlin D. E., "Estimation, Control, and the Discrete Kalman Filter, " Springer-Verlog, 1989.

[2] Cover T. M. and Thomas J. A., "Elements of Information Theory," John Wily and Sons, 1991.

[3] Efron B., "Bootstrap Methods: Another look at the Jacknife," Ann Statistics, 7: 1-26, 1979.

[4] Efron B., Tibshirani R., "Bootstrap Methods for Standard Errors, Confidence Intervals, and Other measures of Statistical Accuracy," Statistical Science, pp 54-77, 1985.

[5] Fayyad M., Irani B., "On the Handling of Continuous-Valued Attributes in Decision Tree Generation," Intl. J. of Machine Learning, 8: 87-102, 1992.

[6] Goonatilake S. and Khebbal S. (ed), "Intelligent Hybrid Systems," John Wily and Sons, 1995.

[7] Grzymala-Busse J. W., "The Rule Induction System LERS Q: A Version for Personal Computers," Proc. of The Intl. Workshop on Rough Sets and Knowledge Discovery, Baniff, Alberta, Canada, p 509, 1993.

[8] Hashemi R. R., Le Blanc L., Rucks C., Sheary A., "A Neural Network for Transportation Safety Modeling," Intl. J. of Expert Systems With Applications 9(3): 247-256, 1995.

[9] Hashemi R. R., Pearce B. A., Hinson W. G., Paule M. G., and Young J. F., "IQ Estimation of Monkeys Based on Human Data Using Rough Sets," Proc. of The Intl. Workshop on Rough Sets and Soft Computing, San Jose, California, pp 400-407, 1994.

[10] Hashemi R. R., Jelovsek F. R., Razzaghi M., "Developmental Toxicity Risk Assessment: A Rough Sets Approach," Intl. J. of Methods of Information in Medicine, 32: 47-54, 1993.

[11] Hashemi R. R., Pearce B. A., Arani R. B., Hinson W. G., and Paule M. G., "A Rough-Genetic Approach for classification of Complex data" Proc. of The Symposium on Applied Computing, Philadelphia, PA, pp 124-130, 1996.

[12] Hashemi R. R., Jelovsek F. R., "Inductive Learning From Examples: A Rough Sets Approach," Proc. of The 1991 ACM/IEEE Intl. Symposium on Applied Computing, Kansas City, Missouri, pp 346-349, 1991.

[13] Hashemi R. R., Jelovsek F. R., Razzaghi M., Talburt J. R., "Conflict Resolution in Rule Learning From Examples," Proc. of The 1992 ACM Conference on Applied Computing, Kansas City, Missouri, pp 598-602, 1992.

[14] Packard N. H., "A Genetic Learning Algorithm for the Analysis of Complex Data," Complex Systems, 4:543-572, 1990.

[15] Paule, M.G., "Analysis of Brain Function Using a Battery of Schedule-Controlled Operant Behaviors," Neurobehavioral Toxicity: Analysis and Interpretation, B. Weiss and J. O'Donoghue, Eds., Raven Press, New York, pp. 331-338, 1994.

[16] Paule, M.G., "Approaches to Utilizing Aspects of Cognitive Function as Indicators of Neurotoxicity," Neurotoxicology: Approaches and Methods, L. Chang and W. Slikker Jr., Eds., Academic Press, Orlando, FL., pp. 301-308, 1995.

[17] Paule, M.G., Cranmer, J.M., Wilkins, J.D., Stern, H.P., and Hoffman, E.L., "Quantitation of Complex Brain Function in Children: Preliminary Evaluation Using a Nonhuman Primate Behavioral Test Battery," Neurotoxicology 9(3): 367-378, 1988.

[18] Paule, M.G., Forrester, T.M., Maher, M.A., Cranmer, J.M., and Allen, R.R., "Monkey Versus Human Performance in the NCTR Operant Test Battery," Neurotoxicol. Teratol. 12(5): 503-507, 1990

[19] Pawlak Z., "Rough Classification," Intl. J. of Man-Machine Studies, 20:469-483, 1984.

[20] Pawlak Z., Slowinski K., and Slowinski R., "Rough Classification of Patients After Highly Selective Vagotomy for Duodenal Ulcer," Intl. J. of Man- Machine Studies, 24:413-433, 1986.

[21] Quinlan R. J., "Discovering Rules by Induction From Large Collections of Examples," Expert Systems in the Micro-Electronic Age, Edinburg University Press, Edinburg, pp 168-201, 1979.

[22] Zurada J. M., "Introduction to Artificial Neural Systems, " West Publishing Co., 1992.

10

ROUGH SETS AS A TOOL FOR STUDYING ATTRIBUTE DEPENDENCIES IN THE URINARY STONES TREATMENT DATA SET

Jerzy Stefanowski*, Krzysztof Słowiński**

** Institute of Computing Science, Poznań University of Technology, 3A Piotrowo Street, 60-965 Poznań, Poland, e-mail : stefanj@pozn1v.put.poznan.pl*

*** Clinic of Traumatology, K.Marcinkowski University of Medical Sciences in Poznań, 1/2 Długa Street, 61-848 Poznań, Poland, e-mail : slowik@pozn1v.put.poznan.pl*

ABSTRACT

The medical experience with urolithiasis patients treated by the extracorporeal shock wave lithotripsy (ESWL) is analysed using the rough set approach. The evaluation of the significance of attributes for qualifying patients to the ESWL treatment is the most important problem for the clinical practice. The use of a simple rough set model gives a high number of possible reducts which are difficult to interpret. So, the heuristic strategies based on the rough set theory are proposed to select the most significant attributes. All these strategies lead to similar results having a good clinical interpretation.

1 INTRODUCTION

Medicine in last decades of this century is characterized by an enormous development and expansion of measurement and laboratory techniques. It creates an increasing stream of data which must be analyzed by the physicians. These data contain usually different information about patients: e.g. information coming from interviewing and investigating patients by specialists, measure-

ment parameters characterizing the patients' condition, and data describing the course of the treatment.

A large part of this data is now being stored in databases. Usually such records may have different practical importance for the physicians. So, they want to analyse collected data sets concerning, e.g. the problems of diagnosis and/or treatment of a given disease in order to find and select the most important part from the diagnostic point of view. It should be stressed that a selection of *attributes* (i.e. characteristic features) describing patients to the diagnostic procedure is often based on an intuitive determination of their diagnostic and prognostic significance. So, it is possible, that among the attributes chosen to characterize patients there are ones which may be less important than others or even unimportant for predicting the results of the treatment. The evaluation of usefulness of attributes is particularly important for these medical problems which are quite new and where the diagnostic knowledge is still imprecisely defined.

An example of such a medical problem is the analysis of the clinical experience with urolithiasis patients treated by extracorporeal shock wave lithotripsy (ESWL) at the Urology Clinic of the Medical Academy in Poznań [9]. Urolithiasis is one of the most common diseases of urinary tract. The current progress in the urinary stones treatment is based on a development of a non - invasive method of disintegration of calculi by extracorporeally induced shock waves, i.e. the extracorporeal shock waves lithotripsy (**ESWL**) (cf. [1], [2], [6]). In order to qualify patients for the ESWL treatment different data, i.e. attributes, characterizing patients, are taken into account. The source of these data is usually: anamnesis (i.e. information coming from investigating patients by the physician), laboratory and imaging tests.

In should be stressed, however, that the ESWL treatment of patients is a new urology techniques, applied for a relatively short period (e.g. in Urology Clinic of the Medical Academy in Poznań since 1990). Moreover, there is still lack of convincing recommendations for it in a medical literature. So, all attempts to study recommendations for the ESWL treatment are very interesting for the practitioners.

The main aims of performed analysis from the medical point of view are:

- an evaluation of usefulness of particular attributes for qualifying patients to the ESWL treatment,

- an identification of the most significant attributes for the qualification of patients.

The additional aim is to discover relationships between values of attributes and a prediction of the course of the ESWL treatment as well as its final result for the patients.

The representation of the ESWL experience (for a smaller number of patients) has been preliminary examined in the study [14]. In this study a group of the most interesting attributes was found out using the approach based on the *rough set theory* [10]. This study was mainly focused on performing the evaluation of the attribute ability to approximate patients classification (connected with qualifying patients to the ESWL treatment), and looking for, so called, reducts of attributes (i.e. subset of attributes ensuring the same approximation of patients' classification as a set of all attributes). However, the authors and medical experts met difficulties with interpretation of obtained reducts because of their too high number.

As we want to avoid the above ambiguity in interpreting results, it seems to be necessary to consider introducing a modified approach allowing to evaluate the attribute significance in a more convincing way.

In the following paper, we are going to use three different and independent heuristic approaches which should identify the most significant attributes in the case of a *'difficult' data sets*. These are multistrategic approaches which combine elements of the rough set theory with techniques of dividing the information system into subsystems and algorithms of the rule discovery.

The usefulness of these strategies is verified in the analysis of the redefined ESWL data set. The number of patients in the currently considered ESWL data set has been extended over 27% in comparison to [14].

The paper is organised as follows. Section 2 gives the brief description of the data set. Then, basic information about the chosen methodology are given in section 3. Section 4 contains a detailed description of the performed analysis. Discussion of obtained results and final remarks are given in Section 5.

2 DATA DESCRIPTION

The ESWL treatment of urinary stones has been performed at the Urology Clinic of the Medical Academy in Poznań since 1990. We chose to the analysis only data about patients with known and confirmed (i.e. not leading to ambiguity in the interpretation) long term results of the treatment. The patient's condition before the ESWL treatment was described by attributes which are currently considered in the urological clinical practice. These are 33 preoperation attributes including
- anamnestic attributes (investigation patients by the physician),
- attributes presenting results of laboratory as well as x-ray and ultrasound imaging tests.
The definition of attributes and values belonging to their domains is presented in Table 2. Let us notice all of these attributes (except attribute 1 and 3) have a qualitative character. Their domains usually consist of a limited number of values which are qualitative and linguistic terms. In addition, the domains of many attributes cannot be ordered.

The postoperation conditions of the patients were described by two attributes having the following clinical meaning:
1. A patient's physical condition after the lithotripsy, i.e.:
- treatment without complications,
- treatment with complications.
2. Long term results of the treatment:
- recovery (good results),
- no recovery,
- lack of effects.

Classif-ication	Meaning of the classification	Decision classes	clinical meaning of the original value
$\mathcal{Y}_1$	Patient's condition after the ESWL lithotripshy	1 2	good - without complications with complications
$\mathcal{Y}_2$	Long-term result of the treatment	1 2 3	recovery no recovery lack of effects

Table 1 The definition of the classification of patients in the ESWL information system

Table 2 The definition of attributes creating the ESWL information system

No.	Attribute name	no. of values of attribute
1	Age	3
2	Sex	2
3	Duration of disease	2
4	Type of urolithiasis	2
5	Lithuresis	2
6	Operations in the past	4
7	Nephrectomy	2
8	PNCL	2
9	Number of the ESWL treatment previously done	4
10	Evacuation of calculi by zeiss catheter	3
11	Lumbar region pains	4
12	Dyspectic symptoms	3
13	Basic dysuric symptoms	3
14	Other dysuric symptoms	4
15	Temperature	3
16	General uriscopy	2
17	Urine reaction	3
18	Erythrocyturia	2
19	Leucocyturia	2
20	Bacteriuria	2
21	Crystaluria	8
22	Proteinuria	1
23	Urea	2
24	Creatinine	2
25	Bacteriological test	2
26	Kidney location	4
27	Kidney size	3
28	Kidney defect	4
29	Status of urinary system	6
30	Secretion of urinary contrast	4
31	Location of the concrement	8
32	Calixcalculus	7
33	Stone size	4

The two postoperation attributes define two classifications of patients, denoted as $\mathcal{Y}_1$ and $\mathcal{Y}_2$ respectively. These definitions are presented in Table 1. The values of these classifications are called *decision classes.* The both classifications are typical standards used to evaluate the medical treatment.

Although the current experience at the Urology Clinic of the Medical Academy in Poznań includes approximately over 1000 patients per year, only part of it can be taken into account. In study [14] data about 343 patients were available. Nowadays, we have extended this number up to 435 patients.

3 BRIEF INFORMATION ABOUT THE METHOD

From the medical point of view the performed analysis should verify the currently used intuitive indications for qualifying patients to the ESWL treatment and help the urologists to predict the result of the treatment. In other words, these aims lead to the evaluation of usefulness of particular preoperation attributes for two patients' classifications and to the identification of the most significant attributes for these classifications.

The rough sets theory [10] is applied to achieve these aims. We have decided to choose it because of at least two reasons. The first one is connected with a qualitative character of analysed data what makes them difficult for standard statistical techniques. The second reason is an inspiration of previous successful applications of the rough sets theory in the analysis of a surgical experience [11], [12], [13].

From the rough set theory point of view the analysis is connected with examining *dependencies between attributes* in the defined data set (called further an *ESWL information system*). More precisely, similarly to previous medical applications [11], [12],[13], the following elements of rough set theory are used:

- creating classes of *indiscernibility relation* (atoms) and building *approximations* of the objects' classification,
- evaluating the ability of attributes to approximate the objects' classification; the measure of the *quality of approximation of the classification* is used to for this aim; it is defined as the ratio of the number of objects in

the lower approximations to the total number of objects in the information system,

- discovering *cores* and *reducts* of attributes (the reduct is the minimal subset of attributes which ensures the same quality of classification as the entire set of attributes; the core is an intersection of all reducts),
- examining the *significance* of attributes by observing changes in the quality of approximation of the classification caused by removing or adding given attributes.

All necessary definitions could be found in [10], [15] or [18].

Results obtained in [14] show that using these elements to identify the most significant attributes for the two patients' classifications may be insufficient. So, we propose to use independently three additional heuristic approaches directly oriented to determine the most significant attributes. The heuristics are the following:

- The strategy based on adding to the core the attributes of the highest discriminatory power,
- The strategy based on dividing the set of attributes into disjoint subsets and analysing the significance of attributes inside subsets,
- The strategy oriented into the analysis of the condition parts of decision rules induced from the information system.

In the first strategy, the core of attributes is chosen as a starting reduced subset of attributes. It usually ensures lower quality of approximation of the objects' classification than all attributes. A single remaining attribute is temporarily added to the core and the influence of this adding on the quality is examined. Such an examination is repeated for all remaining attributes. The attribute with the highest increase of the quality of classification is chosen to add to the reduced subset of attributes. Then, the procedure is repeated for remaining attributes. It is finished when an acceptable quality of the classification is obtained. If there are ties in choosing attributes several possible ways of adding are checked. This strategy has been previously used by authors in [11] giving results having the good medical interpretation.

The aim of the second strategy is to reduce the number of interchangeable and independent attributes in the considered information system. If the system

contains too many of such attributes, one usually gets as a result an empty core, high number of equivalent reducts and atoms supported by single objects. We suggest to divide the set of all attributes into disjoint subsets. Each subset should contain attributes which are dependent each other in a certain degree and have a common characteristic for a domain expert. Such a division could be done either nearly automatically as in [20] or depending on the background domain knowledge (as it was done in [13]).

In two previous strategies elements of the rough set theory were used to determine the significance of attributes. On the other hand, the objects represented in the information system are also treated as *learning examples* and *decision rules* can be induced from them. In this paper, we assume that decision rules are represented in the following form:

$$IF\ (a_1,v_1)\&(a_2,v_2)\&\ \ldots\ \&(a_n,v_n)\ \ THEN\ class_j$$

where a_i is the ith attribute, v_i is its value and $class_j$ is one of the decision classes in the objects' classification.

The decision rules reflect the *important* and *hidden* relationships between values of condition attributes and a classification decision [19]. So, it is also possible to examine the syntax of condition parts of these rules and to identify the condition attributes occurring the most often in the rules. This concept is similar to the heuristic used in INLEN system [7] and has been also studied by Stefanowski in [8].

In this study for the rule discovery we have mainly used our implementation of LEM2 algorithm introduced by Grzymala (see [3], [4]. In this algorithm inconsistencies in data sets are handled by means of the rough set theory. So, the approximations of decision classes are treated as the target concepts. The algorithm LEM2 induces from lower approximations of decision classes, so called, *discriminating rules* (also called certain rules). These rules distinguish positive examples, i.e. objects belonging to the lower approximation of the decision class, from other objects (cf. introductory sections in [4], [5],[17]).

To help the physicians in evaluating the induced rules we use two measures characterizing the rules: *strength* of the rule, and *length* of the rule. The strength of the rule is a number of learning examples (i.e. here patients in the *ESWL information system*) satisfying the condition part of the rule. The length of the rule is the number of elementary conditions (i.e. attribute value pairs)

being used in the rule. Generally, we interesting in discovering the shortest and strongest (i.e. the most general) rules.

4 ANALYSIS OF THE ESWL INFORMATION SYSTEM

4.1 Looking for reducts

Let us consider two classifications: $\mathcal{Y}_1$ and $\mathcal{Y}_2$. The first one determines the patient's condition after the performing lithotripsy treatment and the second classification expresses the final ESWL treatment result.

Decision class	Number of patients	Lower approx.	Upper approx.	Accur-accy
	$\text{card}(Y_1)$	$\text{card}(\underline{Q}Y_1)$	$\text{card}(QY_1)$	$\mu_Q(Y_1)$
1	296	296	296	1.00
2	139	139	139	1.00

Table 3 First classification ($\mathcal{Y}_1$) - accuracy of approximation of each class by all attributes Q

To analyse the dependency between condition attributes and the classification of patients the rough set theory was used. The results are presented in Table 3 and 4. It can be seen that the quality of approximations of the patients' classification, by the set of all 33 attributes (denoted by Q) in both cases was equal to 1.00. However, the number of atoms was the nearly same as the number of objects (patients), and was equal to 434 for both classifications. Although, the quality of the approximation of the classification was the maximal one, this number of atoms is too high. Nearly all of these atoms were represented by single patients. So, they could not be treated as a good basis for expressing strong classification patterns. One can check, that similar results were also obtained for the smaller number patients in the previous study [14].

Then, we looked for cores and reducts of attributes. Using the microcomputer program RoughDAS [16] we were able to conclude, that the core of the first classification $\mathcal{Y}_1$ is empty and the core of classification $\mathcal{Y}_2$ consisted of two attributes only (i.e. 20 and 21). For both classifications, we found out that

Decision class	Number of patients	Lower approx.	Upper approx.	Accur-accy
	card(Y_2)	card($\underline{Q}Y_2$)	card($\overline{Q}Y_2$)	$\mu_Q(Y_2)$
1	270	270	270	1.00
2	143	143	143	1.00
3	22	22	22	1.00

Table 4 Second classification ($\mathcal{Y}_2$) - accuracy of approximation of each class by all attributes Q

the number of the reducts was very high. We could not precisely determine it within reasonable time because of the limited capacity of the used computer equipment.

Let us, notice, that identical results (i.e. getting many reducts) were obtained previously for the smaller number of analysed patients. We could suspect, that the attributes used to construct the data set are independent each other. So, taking into account their ability to approximate the patients' classifications we must say that they are interchangeable. One can remove few of them and others will take their role and still give the highest classification ability. As a result, the number of reducts is very high and even finding all of them would not lead to any good solutions because experts cannot analyse the reducts and indicate the most acceptable one.

Therefore, to help the urologists in determining the significance of attributes, we decided to use three heuristic approaches to select the most significant attribute.

Before performing these experiments, we additionally checked the contents of the information system. We calculated the distribution of values for each attributes over all objects. It was found out that the distributions of possible values for attributes 7, 8, 10, 17, 26, 27, and 28 are characterized by occurring mainly the one single value (i.e. the same value for 405 - 430 objects). So, one could suspect that these attributes may have weak discriminatory power.

4.2 Adding attributes to the core

Proceeding in the way described in section 3, for both classifications we obtained the most acceptable reduced subset of attributes. They are presented in Table

Classification	Attributes
$\mathcal{Y}_1$	1 3 6 11 14 21 22 25 28 29 30 31 33
$\mathcal{Y}_2$	1 2 6 11 14 16 20 21 31 33

Table 5 Acceptable subsets of attributes for both classifications obtained as a result of adding the most discriminatory attributes to a core

The current reduced subset of attributes	Added attribute	Quality of approx. of classification after adding the attribute
20,21	11	0.08
	31	0.09
	32	0.07
20,21,31	6	0.26
	11	0.30
	33	0.25
11,20,21,31	6	0.50
	33	0.48
6,11,20,21,31	1	0.69
	33	0.66
1,6,11,20,21,31	33	0.82
	14	0.80
	3	0.80
1,6,11,20,21,31,33	2	0.90
	3	0.82
1,2,6,11,20,21,31,33	14	0.94
	16	0.94
1,2,6,11,14,20,21,31,33	16	0.99
	18	0.98

Table 6 Partial listing of steps in adding attributes to the core for classification $\mathcal{Y}_2$

5. We noticed that starting subsets of attributes gave a very low quality of approximation of the patients' classification (i.e. around 0.01).

The partial listing of the steps of adding attributes in this strategy (for classification $\mathcal{Y}_2$) is presented in Table 6. Due to the large number of analysed possible

adding, we give information about choosing between the most discriminatory attributes only. The other remaining attributes gave the smaller increase of the quality.

The obtained reduced subsets of attributes are nearly the same as the ones found in the study [14]. On the other hand, one can notice that this strategy started from adding attributes to the core characterized by a very low quality of approximation of the classification and first additions did not lead to the fast increase of this quality. The final results partly depends on the first choices which in the ESWL case are not fully reliable. This is the additional motivation to check other strategies.

4.3 Dividing the set of attributes

The aim of this strategy is to solve the difficulties with existing in the ESWL information system too many interchangeable and independent attributes. We suggest to divide the set of all attributes into disjoint subsets. According to the medical experts' background knowledge it is possible to divide attributes into two disjoint subsets which have a different medical source and interpretation:

- attributes coming from the physician's investigation of the patient - anamnesis; i.e. these are attributes 1 - 14 and they create *information system A*,
- attributes obtained as a result of laboratory tests and examinations; i.e. these are attributes 15 - 33 and they create *information system B*.

Then, for both classifications $\mathcal{Y}_1$ and $\mathcal{Y}_2$ and each information system A and B we examined the significance of attributes using "traditional rough set approach" [12].

For classification $\mathcal{Y}_1$ and information system A we calculated the quality of classification. It was equal to 0.8. The information system contained one core and reduct, i.e. { 1,2,3,4,6,8,9,10,11,12,13,14 }. Then, we checked the influence of particular core attribute on the ability of approximation of the objects classification. We removed temporarily single attributes and observed the value of the quality of the classification for the reduced set of attributes. Results of this experiment are presented in Table 7. The attributes in Table 7 are ordered according to their influence on the quality of classification. So,

information system A		information system B	
The removed attribute	Quality of classification	The removed attribute	Quality of classification
11	0.63	21	0.81
1	0.67	25	0.82
6	0.71	29	0.83
3	0.72	30	0.83
12	0.75	32	0.83
2	0.75	20	0.85
14	0.76	31	0.855
4	0.77	15	0.867
5	0.77	22	0.867
9	0.78	33	0.867
13	0.78	24	0.869
10	0.79	16	0.871
8	0.8	18	0.874
		19	0.874
		17	0.883
		26	0.883
		28	0.883

Table 7 Analysis of the significance of attributes for classification $\mathcal{Y}_1$ and information systems A and B

information system A		information system B	
The removed attribute	Quality of classification	The removed attribute	Quality of classification
11	0.618	33	0.810
6	0.660	30	0.837
1	0.670	21	0.84
2	0.706	25	0.846
3	0.713	32	0.848
5	0.722	20	0.850
12	0.736	31	0.860
9	0.749	22	0.869
14	0.756	29	0.871
4	0.761	18	0.874
10	0.775	19	0.878
13	0.763	24	0.878
		15	0.885
		16	0.885
		26	0.885
		27	0.885
		28	0.885

Table 8 Analysis of the significance of attributes for classification $\mathcal{Y}_2$ and information systems A and B

we can say that attributes 11, 1, 6, 3 seem to be the most significant while attributes 7, 8, 9, 10, 13 have the smallest discriminatory power.

The similar analysis was performed for classification $\mathcal{Y}_1$ and information system B. The quality of classification was equal to 0.89. The information system contained one core and reduct, i.e. {15,16,17, 18, 19, 20, 21, 22, 24, 25, 26, 28, 29, 30, 31, 32,33 }. Results of checking the influence of particular core attribute on the ability of approximation of the objects classification are presented in Table 7. If we analyse the attribute influence on the quality of classification, we can say that attributes 21, 25, 29, 30, 32 seem to be the most significant while attributes 17, 23, 26, 27 , 28 have the smallest discriminatory power.

For classification $\mathcal{Y}_2$ and information system A the quality of classification was equal to 0.78. The information system contained one core and reduct, i.e. {1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 14}. Results of the significance analysis are

presented in Table 8. One can say that attributes 11, 1, 6 seem to be the most significant while attributes 7, 8, 10 , 13 may be treated as unimportant.

The similar analysis was performed for classification $\mathcal{Y}_2$ and information system B. The quality of classification was equal to 0.89. The information system contained one core and reduct, i.e. { 15, 16, 18, 19, 20, 21, 22, 24, 25, 26, 28, 29, 30, 31, 32, 33 }. Then, we checked the significance of particular core attribute. Results are presented in Table 8. One can notice that attributes 33, 30, 21, 25 may be the most significant while attributes 17, 23, 15, 16, 26, 27 , 28 are non significant.

To sum up, one can say that the performed analysis has led us to the following selection of the most significant attributes:

- For classification $\mathcal{Y}_1$ attributes 1,3,6,11,21,25,29,30,32.
- For classification $\mathcal{Y}_2$ attributes 1,6,11,21,25,30,32,33.

4.4 Analysis of condition parts of the decision rules

For both classifications $\mathcal{Y}_1$ and $\mathcal{Y}_2$, we induced decision rules taking into account *all attributes* describing patients. In this strategy, first we restricted the set of induced rules to the "strong" ones, i.e. satisfied by large enough number of learning examples. So, from the set of induced rules we removed the "weak" rules, i.e. supported by few examples only (in the ESWL case - rules satisfied by 1-2 patients for decision classes of low cardinality and rules satisfied by 1-5 patients for larger decision classes). Then, we created the histogram of occurrence of particular attributes in condition parts of "strong" rules. Finally, we chose attributes occurring in the decision rules more often than the defined threshold τ (here 25%). The selected attributes are presented in Table 9.

4.5 Determining the most significant attributes

One can notice that results obtained using three above strategies are similar, i.e. the similar attributes are selected to the final subset. Moreover, similar

Classification	Attributes
$\mathcal{Y}_1$	2 3 5 14 15 25 29
$\mathcal{Y}_2$	1 4 6 11 21 31 32 33

Table 9 The most frequent attributes in decision rules for both classifications, discovered by the analysis of decision rules

Class-fication	no. of subset	Selected attributes	Quality
$\mathcal{Y}_1$	1	1 3 6 11 21 25 29 30 31 32	0.95
$\mathcal{Y}_1$	2	1 2 3 6 11 12 14 20 21 25 29 30 31 32	0.98
$\mathcal{Y}_2$	3	1 6 11 21 25 30 31 32 33	0.89
$\mathcal{Y}_2$	4	1 2 3 4 5 6 11 20 21 22 25 29 30 31 32 33	1.0

Table 10 The classification ability of subsets of the most significant attributes

attributes were identified as non - significant ones. In fact these non - significant attributes contain the ones observed in section 4.1 as 'badly' defined.

Taking into account results obtained by all strategies we performed an additional experiment. We took into account the subsets of the most discriminatory attributes identified by the second strategy. Then, we extended these subsets by adding the most significant attributes chosen by two other strategies. This operation led us to first reduced subsets of the most significant attributes for both patients classifications (subsets no. 1 and 3 in the Table 10). Additionally, we add to these subsets other remaining attributes characterized by still satisfactory discriminatory power. In this way we created two other subsets of attributes (subsets no. 2 and 4. in Table 10). Then, we have checked the ability of these subsets to approximate the patients' classifications. The results are given in Table 12.

We analysed the ESWL information system reduced to these subsets of attributes. For system built using subset no. 1, we found out that this subset is the unique reduct. The significance analysis in this reduced system indicated attributes 6, 11 and 31 as the most important. If the subset no. 2 is used to reduce the ESWL system, one could also get one reduct. For the second classification $\mathcal{Y}_2$, and subset no. 3 we also obtained one reduct. Subset no. 4

led to four possible reducts but their core has enough attributes and ensured quality equal to 0.89.

The above results show that the selected attributes, in particular subset no. 1 for classification $\mathcal{Y}_1$ and subset no. 3 for classification $\mathcal{Y}_2$ have good ability to approximate patients classification and could be taken as the most significant ones.

5 CONCLUSIONS

In this paper the ESWL information system was analysed. The aim of this analysis was to evaluate the significance of attributes describing patients for two classifications expressing the patient's condition after the ESWL treatment and the long-term results of the treatment.

One can notice that use of the simple rough set methodology to analyse the extended ESWL information system (435 patients) has led to results which are very difficult to interpret, i.e. too many atoms supported by single patients, empty or nearly empty cores of attributes, high number of the possible reducts of attributes. To avoid this interpretation ambiguity, three additional heuristic strategies have been used to examine the significance of attributes. We noticed that all these strategies have led to very similar results. So, we could identify the most significant attributes in a more satisfactory way than in the previous study. The selected attributes are presented in Table 10.

We hope that these strategies could be useful tools for studying data relationships in the so called 'difficult' data sets which are very often met in medical applications.

In further analysis of the ESWL problem, the chosen most significant attributes will be the basis for discovering the decision rules. The decision rules will be interpreted from the point of view of the clinical practice. The most powerful decision rules (supported by the highest number of patients, with the good practical interpretation and verified in several tests) will be further used to create the methodology supporting the qualification of new coming patients to the ESWL treatment.

Acknowledgements

The work on this paper has been supported by KBN grant no. 8-S5030 2106. The authors are particularly grateful to Prof. Zbigniew Kwias and Dr Andrzej Antczak from Urology Clinic, K.Marcinkowski University of Medical Sciences in Poznań for collecting and providing the ESWL information system and for the help in the interpretation of the obtained results.

REFERENCES

[1] Chaussy C., Schmiedt E., Forssmann B. and Brendel W., "Contact-free renal stone destruction by means of shock waves". *Eur. Surg. Res.*, **11** 36, 1979.

[2] Chaussy C., Schmiedt E., Jocham D., Walter V. and Brendel W., *"Extracorporeal Shock Wave Lithotripsy. New Aspects in the Treatment of Kidney Stones"*. New York: S. Karger, 1982.

[3] Chan Ch, Grzymala-Busse J.W., "On the two local inductive algorithms PRISM and LEM2". *Foundations of Computing and Decision Sciences*, vol. 19 no 4, 1994, pp. 185–204.

[4] Grzymala-Busse J.W., "LERS - a system for learning from examples based on rough sets". In Slowinski R. (ed.): *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.* Kluwer Academic Publisher, 1992, pp. 3–18.

[5] Grzymala-Busse J.W., Stefanowski J., Ziarko W., "Rough sets: facts vs misconceptions". *ICS Research Report*, 61/95 1995.

[6] Heine G., "Physical aspects of shock-wave treatment". In: J. S. Gravenstein and K. Peter. (eds) *Extracorporeal Shock-Wave Lithotripsy for Renal Stone Disease.* Boston, Butterworths, 1986, chapt. 2, p. 9.

[7] Imam I.F., Michalski R.S., Kerschberg L., "Discovering Attribute Dependence in Databases by Integrating Symbolic and Statistical Analysis Techniques", In Piatetsky-Shapiro (ed.) *Proc. of Knowledge Discovery in Databases Workshop 1993*, AAAI Press, pp. 264–275.

[8] Krusinska E., Stefanowski J., Babic A., Wigertz O., "Rough sets and Correspondence Analysis as Tools for Knowledge Discovery in Studying Data Relationships", In *Proc. of the 2nd Int. Workshop on Rough Sets and Knowledge Discovery RSKD'93*, Banff, Canada, 1993, pp. 15–21.

[9] Kwias Z., and all, "Clinical experiences in treatment of urinary tract's stones using Dornier MPL 9000 lithotriptor", *Nowiny Lekarskie*: **2**, 1992, pp. 15-24 (in Polish).

[10] Pawlak Z., *Rough sets. Theoretical aspects of reasoning about data*, Kluwer Academic Publishers, 1991.

[11] Slowinski K., Slowinski R., Stefanowski J., "Rough sets approach to analysis of data from peritoneal lavage in acute pancreatitis". *Medical Informatics*, **13**, 1988, pp. 143–159.

[12] Slowinski K., "Rough classification of HSV patients", In Slowinski R. (ed.): *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.* Kluwer Academic Publ., 1992, pp. 363–372.

[13] Slowinski K., El. Sanossy Sharif, "Rough Sets Approach to Analysis of Data of Diagnostic Peritoneal Lavage Applied for Multiple Injuries Patients". In Ziarko W. (ed.): *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, 1994, pp. 420–425.

[14] Slowinski K., Stefanowski J, Antczak A, Kwias Z., "Rough set approach to the verification of indications for treatment of urinary stones by extracorporeal shock wave lithotrpsy (ESWL)", In Lin T.Y, Wildberg A.M. (ed.) *Soft Computing*, Simulation Council Inc., San Diego, 1995, pp. 93–96.

[15] Slowinski R. (ed.), *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer Academic Publ., 1992.

[16] Slowinski R., Stefanowski J., "RoughDas and RoughClass software implementation of the rough sets approach", In R. Slowinski (ed.), *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer Academic Publishers, 1992, pp. 445–456.

[17] Stefanowski J., Vanderpooten D. 1994, "A general two stage approach to rule induction from examples". In Ziarko W. (ed.): *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, 1994, pp. 317–325.

[18] Ziarko W. (ed.), *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, London, 1994.

[19] Ziarko W. and Shan N., "KDD-R: A comprehensive system for Knowledge Discovery in databases Using Rough Sets". In Lin T.Y, Wildberg A.M. (eds.) *Soft Computing.* Simulation Inc. Press CA., 1995, pp. 298-301.

[20] Ziarko W, Shan N, "On Discovery of Attributes Interactions and Domain Classifications", *Int. Journal of Automation and Soft Computing* 1995 (to appear).

PART III

RELATED AREAS

11

Data Mining Using Attribute-Oriented Generalization and Information Reduction

Nick J. Cercone, Howard J. Hamilton, Xiaohua Hu and Ning Shan

Department of Computer Science
University of Regina, Regina, Saskatchewan, Canada S4S 0A2
e-mail: {nick, ning, hamilton}@cs.uregina.ca

ABSTRACT

Two families of systems developed over the past several years for easy information access and analysis from relational databases, DBLEARN/DB-Discover and DBROUGH/GRG, are described. DBLEARN is a system for discovering knowledge in databases (data mining) by performing attribute-oriented induction integrating learning-from-examples with set-oriented database operations. DBLEARN extracts generalized data from databases by applying concept tree ascension generalization, which substantially reduces the computational complexity of the learning processes. Different knowledge rules, including characteristic rules, discrimination rules, quantitative rules, and data evolution regularities can be efficiently discovered. DB-Discover is a newer implementation of attribute-oriented induction, which features a graphical user interface and a significantly faster discovery process (approximately 1000-fold increase in speed). DBROUGH is a rough set based knowledge discovery system which integrates database operations, rough set theory and machine learning methods (data generalization and data reduction). In the data reduction phase, rough sets techniques are applied to the generalized relation to eliminate irrelevant or unimportant attributes to the discovery task thus reducing the generalized relation to the best minimal attribute set. GRG (Generalization, Reduction, Generation) combines the best features of DB-Discover and DBROUGH into a single system providing improved efficiency and more fine-grained results. After describing these system families, design issues are discussed.

1. INTRODUCTION AND DATA MINING

We are forever making tradeoffs in computer science. Historically we have devised algorithms and written programs in which the major tradeoff has been between computing speed (time) versus computer memory (storage). With present-day fast processors and large capacity memories and disks, we can attempt to solve problems given up a generation ago as unreasonable. Parallel processing has added another dimension of capability to our repertoire of problem solving tools.

A subtler form of the traditional space/time tradeoff is the decision of search versus inference: when does the cost of retrieving information exceed the cost of re-creating that information? It is impractical to predict all possible valid inferences which can be made from a database relational structure and the values of the attributes, and many of these inferences would be meaningless. Nevertheless determining the search/inference tradeoff is useful. This tradeoff is the crux of the knowledge discovery in databases (KDD) or data mining process.

Data mining in relational databases requires three primitives for the specification of a discovery task: *task-relevant data, background knowledge,* and the *expected representations of the learned results.* We can subsequently generalize our results from relational databases to other databases as well.

Characterizing the features of science graduate students requires only data relevant to science graduates, but this data may extend over several relations. Thus, a query can be used to collect task-relevant data from the database. Task-relevant data can be viewed as examples for learning and learning-from-examples is an important strategy for knowledge discovery in databases. Most learning-from-examples algorithms partition the set of examples into positive and negative sets and perform generalization using the positive data and specialization using the negative ones. Unfortunately, a relational database does not explicitly store negative data, and thus no explicitly specified negative examples can be used for specialization. Therefore, a database induction process relies only on generalization, which must be performed cautiously to avoid over-generalization.

Concept hierarchies represent background knowledge necessary to control the generalization process. Different levels of concepts can be organized into a taxonomy of concepts which is partially ordered according to a general-to-specific ordering. The most general concept is the null description, described by a reserved word "ANY", and the most specific concepts correspond to the specific values of attributes in the database (Han et al., 1992). Using a concept hierarchy, the rules learned can be represented in terms of generalized concepts and stated in a simple and explicit form, which is desirable to most users.

A concept hierarchy table of a typical university database for three attributes is shown in Table 1.

Attribute	concept	Values
Major	Sciences	Biology, Chemistry, Physics, ...
	Humanities	English, Philosophy, Religious Studies, ...
	Social Sciences	Political Science, Sociology, History, ...
	ANY	Science, Humanities, Social Sciences, ...
Birth-Place	British Columbia	Vancouver, Victoria, Richmond, ...
	Alberta	Edmonton, Calgary, Red Deer, ...
	Saskatchewan	Regina, Saskatoon, Moose Jaw, ...
	ANY	British Columbia, Alberta, Saskatchewan, ...
GPA	Excellent	80, 81, ..., 100
	Above Average	70, 71, ..., 79
	Average	60, 61, ..., 69
	ANY	Excellent, Above Average, Average, ...
...	...	...

Table 1: Example Concept Hierarchy Tables.

Concept hierarchies can be provided by knowledge engineers or domain experts. This is realistic even for large databases since a concept tree registers only the distinct discrete attribute values or ranges of numerical values for an attribute which is, in general, not very large and can be input by domain experts. Moreover, many conceptual hierarchies are actually stored in the database implicitly. For example, the information that "Vancouver is a city of British Columbia, which, in turn, is a province of Canada", is usually stored in the database if there are "city", "province" and "country" attributes. Such hierarchical relationships can be made explicit at the schema level by indicating "city province country". Then, the taxonomy of all the cities stored in the database can be retrieved and used in the learning process.

Some concept hierarchies can be discovered automatically or semi-automatically. Numerical attributes can be organized as discrete hierarchical concepts, and the hierarchies can be constructed automatically based on database statistics. Such automatic construction can be performed by first obtaining the distribution of attribute values in the database, then setting the range of the values and performing refined classifications in tightly clustered subranges. For example, for an attribute "CGPA", an examination of the values in the database discloses that cumulative grade point averages (CGPAs) fall between 0 to 4, and most CGPA's for graduates are clustered between 3 and 4. One may classify 0 to 1.99 into one class, and 2 to 2.99 into another but give finer classifications for those between 3 and 4. Even for attributes with discrete values, statistical techniques can be used under certain circumstances. For example, if the birth-places of most employees

are clustered in Canada and scattered in many different countries, the highest level concepts of the attribute can be categorized as "Canada" and "foreign". Thus, the available concept hierarchies can be modified based on database statistics. Moreover, the concept hierarchy of an attribute can also be automatically discovered or refined based on its relationship with other attributes (Cai et al., 1991).

Different concept hierarchies can be constructed on the same attribute based on different viewpoints or preferences. For example, the birthplace could be organized according to administrative regions such as provinces, countries, etc., geographic regions such as east-coast, west-coast, etc., or the sizes of the city, such as, metropolis, small-city, town, countryside, etc. Usually, a commonly referenced concept hierarchy is associated with an attribute as the default concept hierarchy for the attribute. Other hierarchies can be selected explicitly by users.

Rules are one of the expected forms of the learning results. Different rules, such as characteristic rules, discrimination rules, data evolution regularities, etc. can be discovered by the generalization processes. A *characteristic rule* is an assertion which characterizes a concept satisfied by all or a majority of the examples in the class undergoing learning (the target class). For example, the symptoms of a specific disease can be summarized by a characteristic rule. A *discrimination rule* is an assertion which discriminates a concept of the target class from other (contrasting) classes. For example, to distinguish one disease from others, a discrimination rule should summarize the symptoms that discriminate this disease from others. *Data evolution regularities* represent the characteristics of the changed data if it is a characteristic rule, or the changed features which discriminate the current data instances from the previous ones if it is a discrimination rule. If quantitative measurement is associated with a learned rule, the rule is called a *quantitative rule.*

In learning a characteristic rule, relevant data are collected into one class, the target class, for generalization. In learning a discrimination rule, it is necessary to collect data into two classes, the target class and the contrasting class(es). The data in the contrasting class(es) imply that such data cannot be used to distinguish the target class from the contrasting ones, that is, they are used to exclude the properties shared by both classes.

Each tuple in a relation represents a logic formula in conjunctive normal form, and a data relation is characterized by a large set of disjunctions of such

conjunctive forms. Thus, both the data for learning and the rules discovered can be represented in either relational form or first-order predicate calculus.

A relation which represents intermediate (or final) learning results is called an intermediate (or a final) generalized relation. In a generalized relation, some or all of its attribute values are generalized data, i.e., nonleaf nodes in the concept hierarchies. Some learning-from-examples algorithms require the final learned rule to be in conjunctive normal form (Dietterich et al., 1983). This requirement is usually unrealistic for large databases since the generalized data often contain different cases. However, a rule containing a large number of disjuncts indicates that it is in a complex form and further generalization should be performed. Therefore, the final generalized relation should be represented by either one tuple (a conjunctive rule) or a small number (usually 2 to 8) of tuples corresponding to a disjunctive rule with a small number of disjuncts. A system may allow a user to specify the preferred generalization threshold, a maximum number of disjuncts of the resulting formula.

Exceptional data often occur in a large relation. The use of statistical information can help learning-from-examples handle exceptions and/or noisy data. A special attribute, *vote*, can be added to each generalized relation to register the number of tuples in the original relation which are generalized to the current tuple in the generalized relation. The attribute vote carries database statistics and supports the pruning of scattered data and the generalization of the concepts which take a majority of votes. The final generalized rule will be the rule which either represents the characteristics of a majority number of facts in the database (called an approximate rule), or in a quantitative form (called a quantitative rule) indicating the quantitative measurement of each conjunct or disjunct in the rule.

2. Attribute-Oriented Generalization

Attribute-oriented induction in which generalization is performed attribute by attribute using attribute removal and concept tree ascension is summarized below.[1] As a result, different tuples may be generalized to identical ones, and the final generalized relation may consist of a small number of distinct tuples, which

---

[1] In fact we utilize seven strategies when performing attribute-oriented induction: (1) generalization on the smallest decomposable components; (2) attribute removal; (3) concept tree ascension; (4) "vote" propagation; (5) attribute threshold control; (6) generalization threshold control; and (7) rule transformation. See Cai, Cercone, & Han (1991) for details.

can be transformed into a simple logical rule. We presented the general idea of basic attribute-oriented induction in detail elsewhere (Cai et al., 1991). Basic attribute-oriented induction is specified in Algorithm 1.

This basic attributed-oriented induction algorithm extracts a characteristic rule from an initial data relation. Since the generalized rule covers all of the positive examples in the database, it forms the necessary condition of the learning concept, that is, the rule is in the form: learning_class(x) $\rightarrow$ condition(x), where "condition(x)" is a formula containing "x". However, since data in other classes are not taken into consideration in the learning process, there could be data in other classes which also meet the specified condition. Therefore, "condition(x)" is necessary but may not be sufficient for "x" to be in the learning class.

Algorithm 1. Attribute-oriented induction in relational databases.

Input: (i) A relational database, (ii) a concept hierarchy table, and (iii) the learning task, and optionally, (iv) the preferred concept hierarchies, and (v) the preferred form to express learning results.

Output. A {characteristic, discrimination, ...} rule learned from the database.

Method. Attribute-oriented induction consists of the following 4 steps:

Step 1. Collection of the task-relevant data.
Step 2. Basic attribute-oriented induction.
Step 3. Simplification of the generalized relation, and
Step 4. Transformation of the final relation into a logical rule.

Notice that the basic attribute-oriented induction (Step 2) is performed as follows.

begin for each attribute A_i (1<i<n, # of attributes) in the generalized relation **do**
while number_of_distinct_values_in_A_i > generalization_threshold **do**
begin
if no higher level concept in the concept hierarchy table for A_i
then remove A_i
else substitute for the values of A_i's by its corresponding minimal generalized concept;
merge identical tuples
end
while number_of_tuples_in_generalized_relation > generalization_threshold **do**
selectively generalize some attributes and merge identical tuples
end. {Attribute-oriented induction}

Attribute-oriented generalization can also be applied to learning other knowledge rules, such as discrimination rules, data evolution regularities, etc. Since a discrimination rule distinguishes the concepts of the target class from those of contrasting classes, the generalized condition in the target class that overlaps the condition in contrasting classes should be detected and removed from the description of discrimination rules. Therefore, a discrimination rule can be extracted by generalizing the data in both the target class and the contrasting class synchronously and by excluding the properties that overlap in both classes in the final generalized rule.

3. DBLEARN/DB-Discover Family

DBLEARN is our initial version of the machine learning program which implements attribute oriented generalization using concept hierarchies (Han et al., 1992; Han et al., 1993). DB-Discover consists of five components: a user-interface, a command module, a database access module, a concept hierarchy, and a learning module. DB-Discover is illustrated structurally in Figure 1.

The user-interface of DBLEARN consisted of an interactive command line interface which implemented a superset of SQL (structured query language). Subsequently, DB-Discover incorporated a graphical user interface which made the diwcovery program accessible by unskilled data miners via knowledge of the concept hierarchies rather than of the database schema.

The command module is the primary controller of communication between the DB-Discover modules. It provides one or two relations to be generalized to the learning module and provides the functions necessary to do so. The command handler guides the construction of the necessary query to extract desired relations and connects to the database access module to initialize the query and retrieve tuples for the learning module. The command module also directs loading of the concept hierarchies from the concept hierarchy module, provides access to them so the interface can display them, and then performs the translation from high level concepts to low level database attribute values.

We illustrate DBLEARN using the NSERC (Natural Sciences & Engineering Research Council) Grants Information System (NGIS). NGIS is intended to be used by individuals in "universities, government and industry ... to search for grants that are of particular interest". Together with details about individual grants, the system provides summary statistics and reports.

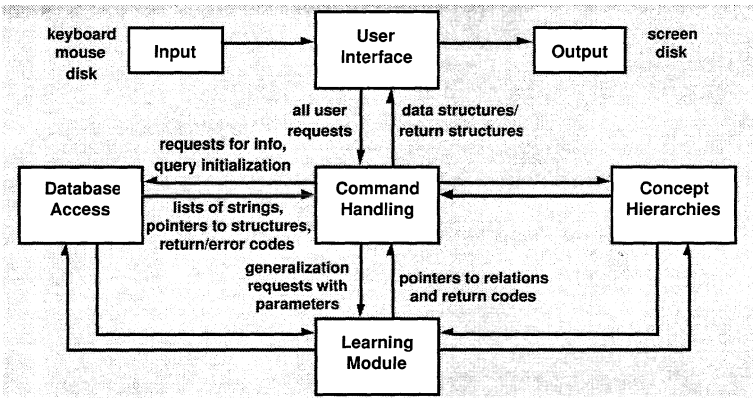


Figure 1. The architecture of DB-Discover.

The central database table consists of rows, each of which describes an award (grant) by NSERC to a researcher. Values constituting each row (that is, the columns constituting the table) specify the different properties of the award, including the name of the recipient, the amount of the award and so on. In the schema diagram in Figure 2, nodes representing the properties of awards are represented by nodes linked to the "Award" node.

Several subsidiary tables record other properties of awards such as, the province of the organization where the work will be performed. Most subsidiary tables simply associate English (and French) phrases describing the entity to a code denoting it. Tables are specified by rectangular nodes and attributes are represented by ovals.

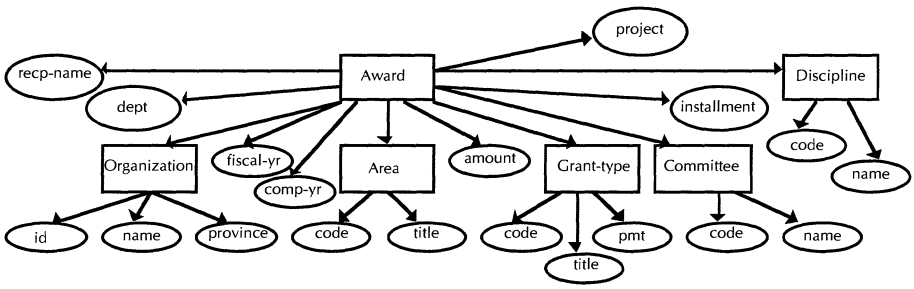


Figure 2. Schema diagram for NSERC Grants Information System.

A concept hierarchy table for the attribute "province" of the NGIS database is shown in Table 2.

attribute	concept	values
province	British Columbia	British Columbia
	Praries	Alberta, Saskatchewan, Manitoba
	Ontario	Ontario
	Quebec	Quebec
	Maritime	New Brunswick, Nova Scotia, Newfoundland, Prince Edward Island
	Canada	British Columbia, Praries, Ontario, Quebec, Maritime,
	ANY	Canada, Outside Canada

Table 2: A concept hierarchy table for the attribute province.

A concept hierarchy table for the attribute "disc_code" of the NGIS database is shown in Table 3.

attribute	concept	values
disc_code	hardware	23000 ~ 23499
	system organization	23500 ~ 23999
	software	24000 ~ 24499
	theory	24500 ~ 24999
	database systems	25500 ~ 25999
	artificial intelligence	26000 ~ 26499
	computing methods	26500 ~ 26999
	other disciplines	0 ~ 22999, 27000
	computing science	hardware, system organization, software, theory, database systems, artificial intelligence, computing methods
	ANY	computing science, other disciplines

Table 3: A concept hierarchy table for the attribute disc_code

DBLEARN's database learning language can be viewed as an extension to SQL. Suppose that the learning task is the discovery of the characteristics of computing science operating grants by amounts, provinces, and the percentages of grants awarded in a given discovered category and the percentage of funds awarded for the discovered category.[2] The learning task is presented to DBLEARN as

```
learn characteristic rule for "CS_Op_Grants"
from Award A, Organization O, grant_type G
where O.org_code = A.org_code and G.Grant_order =
"Operating Grants"
and A.grant_code = G.grant_code and A.disc_code =
"Computer"
in relevance to amount, province, prop(votes)*,
prop(amount)[3]
using table threshold 18
```

The results returned from DBLEARN:

Amount	Geographic Area	# of Grants	Prop. of amount
0-20Ks	B.C.	7.4%	4.7%
0-20Ks	Prairies	8.3%	5.4%
0-20Ks	Quebec	13.8%	8.7%
0-20Ks	Ontario	24.5%	15.7%
0-20Ks	Maritime		
20Ks-40Ks	B.C.	5.3%	7%
20Ks-40Ks	Prairies	5.3%	6.6%
20Ks-40Ks	Quebec	5.1%	7%
20Ks-40ks	Ontario	12.9%	16%
20Ks-40ks	Maritime	1%	1.3%
40Ks-60Ks	B.C.	1.2%	3.1%
40Ks-60Ks	Prairies	0.2%	0.4%
40Ks-60Ks	Quebec	1%	2.5%
40Ks-60Ks	Ontario	5.1%	11.5%
60Ks-	B.C.	0.2%	0.6%
60Ks-	Prairies	0.4%	1.6%
60Ks-	Quebec	0.2%	0.6%
60Ks-	Ontario	1.2%	4.5%
Total:	$10,196,692	100%	100%

Discussion

The original DBLEARN prototype suffered from relatively poor performance, albeit serving well as an adequate proof of concept for knowledge discovery in databases. The primary causes of the performance difficulties are detailed in Carter & Hamilton (1994) and include: excessive storage requirements, inefficient data representations, and inefficient data retrieval.

All initial data in DBLEARN are read into internal data structures before gen-

---

[2] For another example, suppose that the learning task is to learn characteristic rules for graduate students relevant to the attributes Major, Birth-Place, and GPA in a student database using conceptual hierarchies such as the one shown earlier (Table 1) and a threshold value of 3. The learning task is presented to DBLEARN as

```
in relation    Student
learn characteristic rule for    Status = "graduate"
from Student
in relevance to    Name, Major, Birth_Place, GPA
```

After applying the appropriate strategies (generalization on the smallest decomposable components, attribute removal, concept tree ascension, etc.), we could learn the logical formula:

```
∀x graduate(x) →
     {Birth_Place(x) ∈ Canada & GPA(x) ∈ excellent} [75%] |
     {Major(x) ∈ science & Birth_Place(x) ∈ foreign &
          GPA(x) ∈ good} [25%].
```

[3] prop() is a built-in function which returns the number of original tuples covered by a generalized tuple in the final result and the proportion of the specified attribute value.

eralization. Thus DBLEARN defined a maximum number of possible tuples in the initial retrieved relation. A similar strategy was carried out for attributes. These and other assumptions led to an internal relation structure in the multiple megabyte range, enough to cause severe disk swapping on a central server (client-server model). DB-Discover implemented a dynamically allocated minimal storage scheme.

DBLEARN stored all values as strings in (maximum-predicted-length) fixed length tables. DB-Discover developed a compact attribute-value representation in which range[4] and non range values (leaves of concept hierarchies) can be stored compactly.

A key factor in DBLEARN's overall program efficiency is the speed of matching a concept with an attribute value. DBLEARN performed a linear search of unordered concepts. DB-Discover addressed these problems resulting in a 1000-fold speedup of process and additionally added a graphical user interface which permitted users access to discovered data via concept hierarchies.

4. Information Reduction with Rough Sets

Throughout this section we will make use of the information presented in Table 4 by way of illustration. Table 4 illustrates a collection of Japanese and American cars and our objective is to discover knowledge which can tell us factors that affect the gasoline mileage of a car. We partition the table into two disjoint subsets, the *condition attributes* C ("make_model", type of fuel system "fuel", engine displacement "disp", "weight", number of cylinders "cyl", "power", presence of turbocharge "turbo", compression ratio "comp", and transmission "trans") and the *decision attribute* D ("mileage").

An attribute-oriented generalization algorithm similar to DBLEARN and DB-Discover is first applied constrained by two thresholds: the *attribute* threshold and the *proportion* threshold, using the concept hierarchy shown in Table 5.

If the attribute is generalizable, it should be generalized to a higher level concept.[5] The generalized car information system illustrated in Table 6 is the result

---

[4] The notation x~y denotes a range value with a lower bound of x and an upper bound of y, such that for any $z \in x \sim y$, $x \leq z < y$.

[5] An attribute is *generalizable* if there exists a concept hierarchy for the attribute, otherwise it is *nongeneralizable*. Furthermore an attribute is *generalizable to threshold* $t>1$ if it is generalizable and the number of distinct values for the attribute in relation R exceeds t.

Make_Model	fuel	disp	weight	cyl	power	turbo	comp	trans	mileage
Ford Escort	EFI	medium	876	6	high	yes	high	auto	medium
Dodge Shadow	EFI	medium	1100	6	high	no	medium	manu	medium
Ford Festiva	EFI	medium	1589	6	high	no	high	manu	medium
Chevrolet Corvette	EFI	medium	987	6	high	no	medium	manu	medium
Dodge Stealth	EFI	medium	1096	6	high	no	high	manu	medium
Ford Probe	EFI	medium	867	6	high	no	medium	manu	medium
Ford Mustang	EFI	medium	1197	6	high	no	high	manu	medium
Dodge Daytona	EFI	medium	798	6	high	yes	high	manu	high
Chrysler LeBaron	EFI	medium	1056	4	medium	no	medium	manu	medium
Dodge Sprite	EFI	medium	1557	6	high	no	medium	manu	low
Honda Civic	2-BBL	small	786	4	low	no	high	manu	high
Ford Escort	2-BBL	small	1098	4	low	no	high	manu	medium
Ford Tempo	2-BBL	small	1187	4	medium	no	high	auto	medium
Toyoto Corolla	EFI	small	1023	4	low	no	high	manu	high
Mazda 323	EFI	medium	698	4	medium	no	medium	manu	high
Dodge Daytona	EFI	medium	1123	4	medium	no	medium	manu	medium
Honda Prelude	EFI	small	1094	4	high	yes	high	manu	high
Toyoto Paseo	2-BBL	small	1023	4	low	no	medium	manu	high
Chevrolet Corsica	EFI	medium	980	4	high	yes	medium	manu	medium
Chevrolet Beretta	EFI	medium	1600	6	high	no	medium	auto	low
Chevrolet Cavalier	EFI	medium	1002	6	high	no	medium	auto	medium
Chrysler LeBaron	EFI	medium	1098	4	high	no	medium	auto	medium
Masda 626	EFI	small	1039	4	medium	no	high	manu	high
Chevrolet Corsica	EFI	small	980	4	medium	no	high	manu	high
Chevrolet Lumina	EFI	small	1000	4	medium	no	high	manu	high

Table 4: A collection of "cars" information.

attribute	concept	values
make_model	honda	civic, acura, ..., accord
	toyota	tercel, ..., camry
	mazda	mazda_323, mazda_626, ..., mazda 939
	japan (car)	honda, toyoto, ..., mazda
	ford	escort, probe, ..., taurus
	chevrolet	corvette, camaro, ..., corsica
	dodge	stealth, daytona, ..., dynasty
	usa (car)	ford, dodge, ..., chevrolet
	any (make-model)	japan (car), ..., usa (car)
	light	0, ..., 800
	heavy	801, ..., 1200
	medium	1201, ..., 1600
	any (weight)	light, medium, heavy

Table 5: A concept hierarchy for Table 4.

of applying algorithm 2 to Table 4 with all thresholds set to 2 and p = 0.84.

Algorithm 2 differs from previous attribute oriented algorithms (algorithm 1) in the use of the ratio d_i/t_i to choose the next attribute for generalization. Rather than selecting the attributes in arbitrary order, we select the attribute which has the most values in proportion to threshold, thus directing the algorithm to areas where the most improvement is possible.

If the attribute is generalizable, it should be generalized to a higher level concept.[6] The generalized car information system illustrated in Table 6 is the result of applying algorithm 2 to Table 4 with all thresholds set to 2 and p = 0.84. Algorithm 2 differs from previous attribute oriented algorithms (algorithm 1) in the use of the ratio d_i/t_i to choose the next attribute for generalization. We select the next attribute for processing which has the most values in proportion to threshold rather than in arbitrary order, thus directing the algorithm to areas where the most improvement is possible.

Make_Model	fuel	disp	weight	cyl	power	turbo	comp	trans	mileage
USA	EFI	medium	medium	6	high	yes	high	auto	medium
USA	EFI	medium	medium	6	high	no	medium	manu	medium
USA	EFI	medium	heavy	6	high	no	high	manu	medium
USA	EFI	medium	medium	6	high	no	high	manu	medium
USA	EFI	medium	light	6	high	yes	high	manu	high
USA	EFI	medium	medium	4	medium	no	medium	manu	medium
USA	EFI	medium	heavy	6	high	no	medium	manu	low
Japan	2-BBL	small	light	4	low	no	high	manu	high
USA	2-BBL	small	medium	4	low	no	high	manu	medium
USA	2-BBL	small	medium	4	medium	no	high	auto	medium
Japan	EFI	small	medium	4	low	no	high	manu	high
Japan	EFI	medium	light	4	medium	no	medium	manu	high
Japan	EFI	small	medium	4	high	yes	high	manu	high
Japan	2-BBL	small	medium	4	low	no	medium	manu	high
USA	EFI	medium	medium	4	high	yes	medium	manu	medium
USA	EFI	medium	heavy	6	high	no	medium	auto	low
USA	EFI	medium	medium	6	high	no	medium	auto	medium
USA	EFI	medium	medium	4	high	no	medium	auto	medium
Japan	EFI	small	medium	4	medium	no	high	manu	high
USA	EFI	small	medium	4	medium	no	high	manu	high

Table 6: A generalized cars information system.

---

[6] An attribute is *generalizable* if there exists a concept hierarchy for the attribute, otherwise it is *nongeneralizable*. Furthermore an attribute is *generalizable to threshold* t>1 if it is generalizable and the number of distinct values for the attribute in relation R exceeds t.

Algorithm 2. Extracts a generalized information system from a relation (EGIS).

Input: (i) A set of task-relevant data R, a relation or arity n with a set of attributes A_i $(1 \leq i \leq n)$; (ii) a set H of concept hierarchies where each $H_i \in H$ is a hierarchy on the generalized attribute A_i, if available; (iii) t_i is a threshold for attribute A_i, and d_i is the number of distinct values of attribute A_i; and (iv) p defined by user is a proportional value $(0 < p \leq 1)$.

Output. The generalized information system R'.

MAXTUPLES $\leftarrow p \times |R|$; $R' \leftarrow R$;

while $|R'| \geq$ MAXTUPLES and $\exists d_i > t_i$ **do**

select an attribute $A_i \in A$ such that d_i/t_i is maximal

if A_i is generalizable

then ascend tree H_i 1 level & make appropriate substitutions in R'

else remove attribute A_i from R'

endif

remove duplicates from R'; recalculate d_i for each attribute

endwhile

Often it is difficult to know exactly which features are relevant and/or important for the learning task. Usually all features believed to be useful are collected into the database; hence databases normally contain some attributes that are unimportant, irrelevant, or even undesirable for a given learning task. The need to focus attention on a subset of relevant attributes is now receiving a great deal of attention in the data mining community (Matheus et al., 1993; Kira & Rendell, 1992). Pawlak (1982) introduced rough sets theory which provides the necessary tools to analyze a set of attributes globally. Using rough set theory, the minimal attribute set or *reduct* of the attribute in the generalized relation can be computed and each reduct can be used instead of the entire attribute set without losing any essential information. By removing these attributes which are not in the reduct, the generalized relation can be further reduced. To reduce the generalized relation further, two fundamental concepts play an important role - the *reduct* and the *core*. Intuitively, a reduct of the generalized relation is its essential part, that part which is sufficient to define all basic concepts in the class under consideration. The core is, in a certain sense, the reduct's most important part. Reducing the generalized relation entails removal of irrelevant or superfluous attributes in such a way that the set of elementary categories in the generalized relation are preserved. This procedure enables us to eliminate all unnecessary data from the generalized relation, preserving only that part of the data which is most useful for decision making.

Objects can be grouped to represent a certain relationship among a set of attributes *C*, in a generalized information system. Each relationship among the set of attributes *C* correspond to a classification of objects on the generalized information system into disjoint *equivalence* classes, where objects belonging to the same classification class have the same attribute values for every attribute in *C*. An equivalence relation $U \times U \supseteq R(C)$ represents the classification corresponding to the set of attributes in *C*. Pawlak (1991) calls the pair $AS = (U, R(C))$ an *approximation space*.

Information and Attribute Reduction

Before discussing attribute reduction, it is instructive to perform a dependency analysis of attributes first. Let $R^*(C) = \{X_1, X_2, ..., X_n\}$ be the collection of equivalence classes of the relation $R(C)$, where an element X_i is a group of objects having the same values for all attributes in C, and let $R^*(D) = \{Y_1, Y_2, ..., Y_m\}$ be a collection of equivalence classes of the relation $R(D)$, where each element is a group of objects having the same values for all attributes in D and creates a concept class on the universe *U*. The lower approximation in the approximation space *AS*, denoted as *LOW(C, D)* is defined as the union of those equivalent classes of the relation *R(C)* which are completely contained by one of the equivalence classes of relation *R(D)*, that is

$$LOW(C, D) = \cup_{Yi\, R^*(D)} \{X \in R^*(C): Y_i \supseteq X\}$$

The upper approximation in the approximation space *AS*, denoted as *UPP(C, D)*, is defined as the union of those equivalence classes of *R(C)* which are partially contained by one of the equivalence classes of *R(D)*, that is

$$UPP(C, D) = \cup_{Yi\, R^*(D)} \{X \in R^*(C): Y_i \cap X \neq 0\}$$

The lower approximation *LOW(C, D)* characterizes objects which can be classified into one of the concepts without any *uncertainty* based only on the classification information. The upper approximation *UPP(C, D)* is a set of objects which can *possibly* be classified into one of the concepts with some ambiguous measurements. By definition

$$U \supseteq UPP(C, D) \supseteq LOW(C, D)$$

The degree of dependency K(C, D) in the relationship between the groups of attributes C and D can be defined as

$$K(C, D) = card\,(LOW(C, D))/card(U)$$

where *card* yields set cardinality. The dependency between two sets of attributes C and D indicates the extent to which values of attributes in D depend on values of attributes in C. By definition, $0 \leq K(C, D) \leq 1$ because $U \supseteq LOW(C, D)$. If $K(C, D)$ is equal to 1, the dependency is considered to be fully functional. $K(C, D)$ is equal to 0 when none of the values of attributes in D can be uniquely determined from the values of attributes in C.

In actual applications, databases usually contain incomplete and ambiguous information. The original rough sets technique does not use information in the boundary area $UPP(C, D) - LOW(C, D)$ of an approximation space AS. In some situations, this leads to information loss and the inability to take advantage of statistical information. Extensions to rough sets theory to rectify this situation can be found in Shan et al. (1994) and Ziarko (1993). Essentially these extensions draw some elementary sets belonging to the boundary area into the lower approximation; we can easily modify our approach by changing slightly the computation of the degree of dependency. The decision rules obtained in this fashion are characterized by an uncertainty factor which is, in fact, probabilistic that an object matching the condition part of the rule belongs to the concept.

We say that an attribute $a \in C$ is *superfluous* in C with respect to D if $K(C, D) = K(C-\{a\}, D)$; otherwise a is *indispensable* in C with respect to D. If we remove an indispensable attribute, we decrease the degree of dependency, that is, $K(C-\{a\}, D) < K(C, D)$, if a is indispensable. Furthermore, we call a subset B of a set of attributes C a *reduct* of C with respect to D if and only if: (1) $K(B, C) = K(C, D)$; and (2) $K(B, D) \neq K(B,-\{a\}, D)$, for any $a \in B$.[7] The first condition ensures that the reduct preserves the degree of dependency with respect to D and the second condition ensures that the reduct is a minimal subset and that any further removal will change the degree of dependency.

A given information system can have more than one reduct and each reduct can be used to represent the original information system. Hu et al. (1993) computed all reducts for small information systems and then chose one to use. Unfortunately, finding all reducts of an information system is NP-hard (Wong & Ziarko, 1985) and, for many applications such as ours, is also unnecessary. We are interested in finding one "good" reduct.[8] Table 7 illustrates the signifi-

[7] A reduct is a minimal sufficient subset of a set of attributes which preserves the degree of dependency with respect to another set and which has the same ability to discern concepts as when the full set of attributes is used, (Pawlak, 1991).

cance values for the attributes in Table 6. Higher significance value for an attribute indicates greater interaction with decision attributes in D.

attribute name	$\chi 2$
weight	17.54
make_model	12.86
disp	7.08
cyl	5.94
power	5.68
tran	4.53
comp	3.84
fuel	0.63
turbo	0.63

Table 7:

The following greedy algorithm, algorithm 3, constructs a reduct for a generalized information system U.

Algorithm 3. Computes a reduct (GENRED).

Input: (i) A generalized information system U; (ii) a set of attributes C over the information system U; and (iii) the degree of dependency $K(C, D)$ in the information system U;

Output. A reduct, that is, a set of attributes SM.

Compute the significance value for each attribute $a \in C$;
Sort the set of attributes C based on significance values;
$SM \leftarrow 0$;
while $K(SM, D) \neq K(C, D)$
  do /*create subset SM of attr's C by adding attr's */
  select an attr a with the highest significance value in C; $SM \leftarrow a \cup SM$;
  compute degree of dependency $K(SM, D)$ in the information system U
endwhile
$N \leftarrow |SM|$;
for $i = 0$ to $N-1$
  do /*create a reduct of attr's SM by dropping condition attr's */
  remove the i^{th} attribute a_i from the set SM;
  compute the degree of dependency $K(SM, D)$ in the information system U
  if $K(SM, D) \neq K(C, D)$ **then** $SM \leftarrow SM \cup a_i$
  endif
endfor

---

[8] The computation of a "good" reduct depends on the optimality criterion associated with attributes. Alternatively/additionally, we can assign significance values to attributes and base the selection of those values. The chi-square statistic, traditionally used to measure the association between two attributes in a contingency table, compares the observed frequencies with the frequencies that one would expect if there were no association between the attributes (Press et al., 1988).

Algorithm 3 assigns a significance value based on an evaluation function to each attribute and sorts the attributes based on their significance values. A forward selection method is then employed to create a smaller subset of attributes with the same discriminating power as the original attributes. At the end of this phase, the attribute set *SM* contains the "good" performing attribute subset found thus far. Finally, to compute a reduct, a backward elimination method removes attributes, one by one, from the set *SM*. The lower the significance value is, the earlier the attribute is processed. The degree of dependency is calculated at each step based on the remaining attributes in *SM*; if the degree of dependency is changed the attribute is restored to the set *SM*, otherwise it is permanently removed. Attributes remaining in the set *SM* for the reduct, other attributes may be removed. Table 8 illustrates a reduct for the generalized car information system presented in Table 6. The forward selection process collects the attributes with higher significance values one by one. For Table 6 this process stops with the collected set *SM* = {weight, make_model, disp, cyl, power, tran, comp} which has the same degree of dependency as the original set. The backward elimination step deletes redundant attributes from *SM* resulting in the set *SM* = {weight, make_model, power, tran, comp} as a reduct from which further deletion would reduce the degree of dependency.

make_model	weight	power	comp	tran	mileage
USA	medium	high	high	auto	medium
USA	medium	high	medium	manu	medium
USA	heavy	high	high	manu	medium
USA	medium	high	high	manu	medium
USA	light	high	high	manu	high
USA	medium	medium	medium	manu	medium
USA	heavy	high	medium	manu	low
Japan	light	low	high	manu	high
USA	medium	low	high	manu	medium
USA	medium	medium	high	auto	medium
Japan	medium	low	high	manu	high
Japan	light	medium	medium	manu	high
Japan	medium	high	high	manu	high
Japan	medium	low	medium	manu	high
USA	heavy	high	medium	auto	low
USA	medium	high	medium	auto	medium
Japan	medium	medium	high	manu	high
USA	medium	medium	high	manu	high

Table 8: A reduct of the generalized car information system (Table 6).

For n objects (tuples) with a attributes, the time complexity of our algorithm is $O(an + a \log a)$ in the worst case because computing the degree of dependency using a hashing technique is $O(n)$, computing attribute significance values is $O(an)$, sorting the attributes based on significance values is $O(a \log a)$, creating the smaller subset of attributes using a hash technique is $O(an)$, and creating the reduct is $O(an)$.

Before introducing GRG, we first introduce an earlier version entitled DBROUGH, which inspired many of the ideas we wish to incorporate and improve upon.

5. DBROUGH/GRG (Generalize, Reduce, Generate) Family

DBROUGH is a direct descendant of DBLEARN; its architecture is shown in Figure 3. The system takes SQL-like database learning requests and applies different algorithms to discover rules. Again background knowledge is stored in concept hierarchies, which, in this case, can be adjusted dynamically according to database statistics and specific learning requests.

DBROUGH can execute the following procedures to produce results:

(1) **DBChar:** find the characteristic rule for the target class;
(2) **DBClass**: find the characteristic rules of the target class with other classes;
(3) **DBDeci:** find the decision rules for the decision attributes;
(4) **DBMaxi:** find all the maximal generalized rules or the best k maximal generalized rules;
(5) **DBTrend:** find the data trend regularities for the target class;
(6) **DBMkbs:** find different knowledge bases for the target class;

Perhaps the best way to illustrate DBROUGH is by example as well. Hu & Cercone (1994) provide details on system operation, including the syntax of its extended SQL language. Our example illustrates use of the procedure DBChar; specification of the learning task to DBROUGH is as follows:

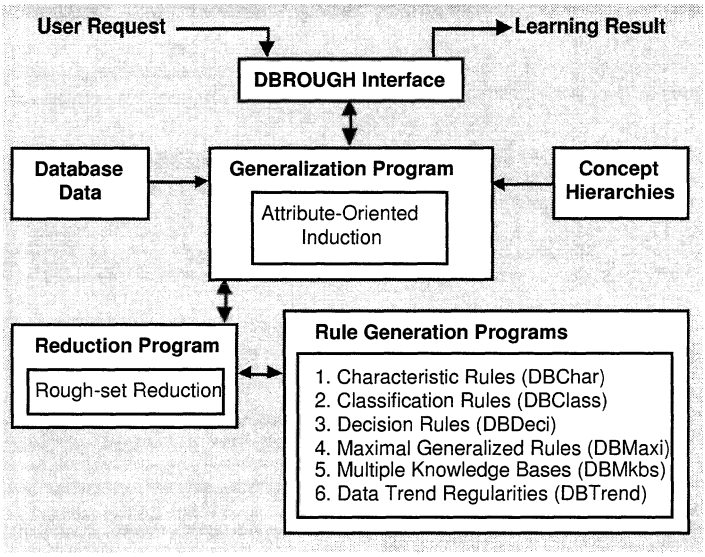


Figure 3. The Architecture of DBROUGH.

```
learn characteristic rule for "CS_Op_Grants"
from Award A, Organization O, grant_type G
where O.org_code = A.org_code and   G.Grant_order=
   "Operating Grants"
and A.grant_code=G.grant_code and A.disc_code="Computer"
in relevance to amount, province, prop(votes)*,
    prop(amount)[9]
using table threshold 18
using hierarchy disc, amount, prov, grant_type  go
```

The results returned from DBROUGH are almost identical to those shown earlier in response to a similar request of DBLEARN, as expected. Another example illustrates the diversity of DBROUGH:

```
learn discrimination rule for "Ontario_CS_Grants"
where O.province= "Ontario"
in constrast to  "Newfoundland_CS_Grants"
where O.province= "Newfoundland"
from award A, organization O, grant_type G
where A.grant_code=G.grant_code and
   A.org_code=O.org_code
and A.disc_code = "Computer"
in relevance to disc_code, amount, grant_order  go
```

[9] prop() is a built-in function which returns the number of original tuples covered by a generalized tuple in the final result and the proportion of the specified attribute respectively.

Notice that both attribute and table threshold values are defaulted. All of the concept hierarchy information required is stored in a default file *concept*. The classification rule for "Ont_Grants" versus "Newfoundland_Grants" is:

```
∀x Ont_Grants(x) ←
      {disc_code="Computer" &
        grant_order="Operating_grants"  &
        amount=("20-40K,40-60K")} [34.387%] |
     {disc_code="Computer" & grant_order="Other"
        & amount=("40K- ,40-60K")} [4.743%] |
      {disc_code="Computer" &
        grant_order="Strategic_grants,
        Operating_grants"  & amount=("40K- ")}
        [5.534%] |
      {disc_code="Computer" &
        grant_order="Strategic_grants"  &
        amount=("40-60K")} [0.004%]
```

The final reduced relation is illustrated in Table 9.

disc_code	grant_order	amount	votes
Computer	Operating_Grants	20-40K	62
Computer	Operating_Grants	40-60K	25
Computer	Other	60K-	7
Computer	Other	40-60K	5
Computer	Strategic_Grants	60K-	8
Computer	Operating_Grants	60K-	6
Computer	Strategic_Grants	40-60K	1

Table 9: The final reduced relation.

DBROUGH is the first system to apply attribute oriented generalization to remove undesirable attributes and generalize the primitive data to a desirable level, much like the DBLEARN family, and then perform a data reduction process based on rough set theory to compute the minimal attribute set (reduct) for use is further reducing the generalized relation. Although the realization of a general purpose, fully automated knowledge discovery system is still in the future, DBROUGH and its successor, GRG (still under development) are promising to lead us to such a realization.

Induction of Decision Rules

Decision rules preserve logical properties of data. They are easy to understand. Decision rules are a common way to represent knowledge in rue-based

expert systems and have become popular in inductive learning systems. A *rule* is a combination of values of some condition attributes such that the set of all objects matching it is contained in the set of objects labeled with the same class (and such that there exists at least one such object). A rule r is denoted as an implication

$$r: (a_{i1} = V_{i1}) \,\&\, (a_{i2} = V_{i2}) \,\&\, \ldots \,\&\, (a_{in} = V_{in}) \rightarrow (d = V_d),$$

where a_{i1}, a_{i2}, ..., and a_{in} are the condition attributes and d is the decision attribute. The set of attribute-value pairs occurring on the left hand side of the rule r is referred to as the *condition part*, denoted *cond(r)*, and the right hand side is the *decision part*, *dec(r)*, so that the rule can be expressed as $cond(r) \rightarrow dec(r)$. Including more condition attributes in *cond(r)* makes the rule more specific. Decision rules obtained directly from the reduced relation (information system) are the specific rules which only match one equivalence class. These rules can be generalized by removing one or several conditions from the condition part.

Our aim is to produce rules in the learning process which are maximally general rules by removing the maximum number of condition attributes values without decreasing classification accuracy of the rule. Computing such rules in especially important in data mining applications since they represent the most general patterns existing in the data. A reduced information system can be considered as a set of specific decision rules, each rule of which corresponds to an equivalence class of $R^*(RED)$ which is the set of equivalence classes generated by the subset of $C \supseteq RED$ of condition attributes C, where the subset *RED* is a reduct of C. Before describing our rule generation algorithm, algorithm 4, we introduce two propositions: *rule redundancy* and *rule inconsistency.*

Rule redundancy:

(1) If r_i and r_j are valid rules where $cond(r_i) = cond(r_j)$ and $dec(r_i) = dec(r_j)$, then r_i and r_j are *logically equivalent rules.*

(2) If r_i and r_j are valid rules where $cond(r_j) \supset cond(r_i)$ and $dec(r_i) = dec(r_j)$, then r_j is *logically included* in r_i.

Rule inconsistency:

(1) If r_i and r_j are valid rules where $cond(r_j) \supseteq cond(r_i)$ and $dec(r_i) \neq dec(r_j)$, then r_i and r_j are *decision inconsistent.*

Algorithm 4 computes a set of maximally generalized rules.

Algorithm 4. Computes a set of maximally generalized rules (GENRULES).

Input: A non-empty set of specific decision rules *RULE*
Output. A non-empty set of maximally general rules *MRULE*
MRULE ← 0; *N* ← |*RULE*| /* *N* is the number of rules in *RULE* */
for *i* = *0* to *N-1* **do**
 r ← r_i
 M ← |*r*| /* *M* is the number of condition attribures in rule *r* */
 compute the significance value SIG for each condition of the rule *r*
 sort the set of conditions of the rule based on the significance values
 for *j* = 0 to *M-1* **do**
  remove the j^{th} condition attribute a_j in rule *r*
  if *r* inconsistent with any rule $r_n \in RULE$ **then**
   restore the dropping condition a_j
  endif
 endfor
 remove any rule $r' \in MRULE$ that is logically included in the rule *r*
 if rule r is not logically included in a rule $r' \in MRULE$ **then**
 MRULE ← $r \cup MRULE$
 endif
endfor

To obtain a set of maximally general rules, algorithm 4 tells us to consider each rule in the set of specific decision rules for dropping conditions until we are left with a set of maximally general rules. The order in which we process the attributes determines which maximally general rule is generated. Thus a maximally general rule may not turn out to be the best with respect to the conciseness or the coverage of the rule. Given a rule with m conditions, we could evaluate all 2^m-1 possible subsets of conditions on the database and select the best rule but this is, in general, impractical.

For a near optimal solution, each condition of the rule is assigned a significance value by an evaluation function before the dropping conditions process is started. The significance value indicates the relevance of this condition for this particular case. Higher significance values indicate more relevance. The process of dropping conditions should first drop the conditions with lower significance values, as described in Ziarko and Shan (1995). Their evaluation function for a condition c_i of a rule is defined as

$$\mathrm{SIG}(c_i) = \mathrm{P}(c_i)(\mathrm{P}(D|c_i) - \mathrm{P}(D)),$$

where $\mathrm{P}(c_i)$ is the probability of occurrence of the condition c_i or the proportion of objects in the universe matching to this condition; $(\mathrm{P}(D|c_i)$ is the

conditional probability of the occurrence of the concept D conditioned on the occurrence of the condition c_i; P(D) is the proportion of the concept D in the database. For example the specific rule (the seventh entry in Table 8) can be translated as

(1) **if** (make_model=USA) & (weight=heavy) & (power=high) & (comp=medium) & (tran=manu)

(2) **then** (mileage=low)

By definition we have

(1) SIG(tran=manu) = -0.03

(2) SIG(make_model=USA) = 0.04

(3) SIG(power=high) = 0.06

(4) SIG(comp=medium) = 0.07

(5) SIG(weight=heavy) = 0.093

Thus we drop conditions of the rule in the sequence given above. No inconsistency results from dropping the first three conditions. After dropping the fourth condition "comp", the new rule "if (weight=heavy) then (mileage=low)" is inconsistent with the specific rule derived from the third entry in Table 8, thus the condition "comp" is replaced. The fifth condition "weight" also cannot be dropped because of inconsistency. Thus the maximally generalized rule for the specific rule derived from the seventh entry in Table 8 is

if (weight=heavy) & (comp=medium) then (mileage=low)

Suppose there are n' tuples (decision rules) with a' attributes in the reduced information system. The computation of significance values of one rule requires computation $O(a'n')$ and the process of dropping conditions on one rule requires $O(a'n')$. Thus finding a maximally general rule for one decision rule requires $O(2a'n')$ time and finding maximally general rules for n' decision rules requires $O(2a'n'^2)$ time. Eliminating redundant rules requires $O(n'^2)$ time and the complexity of algorithm 4 is $O((2a'+1)n'^2) = O(a'n'^2)$.

Table 10 shows the set of maximally general rules corresponding to the values in Table 8 where "-" indicates "don't care". Rules in Table 10 are more concise than the original data in Table 4 and they provide information at a more abstract level. Nevertheless they are guaranteed to give decisions about mileage consistent with the original data. The column "supp" is the number of tuples in the original database which support the generalized rule. This measure provides

confidence because if the tuples in the original database distribute evenly over all possible discrete values for an attribute then it is impossible to obtain a meaningful set of rules. Higher values for "supp" indicate greater confirmation of the rule.

make_model	weight	power	comp	tran	mileage	supp
-	heavy	-	medium	-	low	2
USA	medium	high	-	-	medium	9
USA	medium	-	medium	-	medium	8
-	medium	-	-	auto	medium	4
USA	-	light	-	-	medium	1
-	heavy	-	high	-	medium	1
-	-	medium	high	manu	high	3
Japan	-	-	-	-	high	6
-	light	-	-	-	high	3

Table 10: A set of maximally general rules.

6. Conclusions

Attribute-oriented induction provides a simple and efficient way to learn different kinds of knowledge rules in databases. As a newly emerging field, many systems reported to date are based on previously developed learning algorithms. A major difference of our approach from the others is attribute-oriented generalization, in contradistinction to the tuple-oriented generalizations of other approaches. It is instructive to compare these two approaches.

Both tuple-oriented and attribute-oriented generalization take attribute removal and concept tree ascension as their major generalization technique. However, the former technique performs generalization tuple by tuple, while the latter, attribute by attribute. The two approaches involve significantly different search spaces. Among many learning-from-examples algorithms, we use the candidate elimination algorithm as an example to demonstrate such a difference.

In the candidate elimination technique, the set of all concepts which are consistent with training examples is called the version space of the training examples. The learning process is the search in the version space to induce a generalized concept which is satisfied by all of the positive examples and none of the negative examples. Since generalization in an attribute-oriented approach is performed on individual attributes, a concept hierarchy of each attribute can be treated as a factored version space. Factoring the version space may significantly improve the computational efficiency. Suppose there are p nodes in each concept tree and there are k concept trees (attributes) in the relation, the total size of k

factorized version spaces is $p^{x\ exp\ k}$. However, the size of the unfactorized version space for the same concept tree should be p_k.

Similar arguments hold for other tuple-oriented learning algorithms. Although different algorithms may adopt different search strategies, the tuple-oriented approach examines the training examples one at a time to induce generalized concepts. To discover the most specific concept that is satisfied by all of the training examples, the algorithm must search every node in the search space which represents the possible concepts derived from the generalization on this training example. Since different attributes of a tuple may be generalized to different levels, the number of nodes to be searched for a training example may involve a huge number of possible combinations. Therefore, most learning-from-examples algorithms that adopt the tuple-oriented approach have a huge search space, which affects learning times when operating in large databases.

We have gone significantly beyond DBLEARN which was our first in a series of systems which incorporated attribute oriented generalization. Our latest version in this family, DB-Discover has achieved storage and speed efficiencies of such a magnitude that we are currently installing a version of DB-Discover at Rogers Cablesystems Ltd., Canada's largest cable television supplier, for use by their marketing division. DB-Discover will be put to the test of helping marketing personnel analyze their PPV (pay per view) database. To make DB-Discover useful to unskilled users, a graphical front-end has been designed and tested which will allow Rogers marketing personnel access to generalizable data via any concept hierarchy they design which "makes sense" to the database.

DBROUGH was our first prototype of the new generation data mining utilities. DBROUGH incorporated novel ideas as well. DBROUGH integrated a variety of knowledge discovery algorithms such as DBChar for characteristic rules, DBClass for classification rules, DBDeci for decision rules, etc. This integration permits DBROUGH to exploit the strengths of diverse discovery programs.

Just as DB-Discover incorporated many efficiencies of design and implementation when compared to the earlier DBLEARN efforts, GRG is intended to do the same to the combined efforts which resulted in both DB-Discover and DBROUGH. The importance of the information reduction phase which DBROUGH explored and upon which GRG will capitalize cannot be minimized. The potential for speeding up the learning process and the improvement in the quality of classification, the conciseness and expressiveness of the rules generated

are but a few of the advantages of this approach.

We envision a future prototype GRG working in a distributed client-server architecture like the one illustrated in Figure 4. Figure 4 depicts a distributed system of cooperating agents. Each agent has a specialized function which it provides as a service to the other agents in the system. Distinguished agents interact with human users via X-window interfaces. The agents communicate with each other via Transport Level Interface (TLI) communication channels.

The distributed architecture will allow multiple users on a network to share the CPU intensive services offered by the system, such as English to SQL translation and data mining via DBLEARN, DB-Discover, DBROUGH, and GRG. The graphical user interfaces (GUIs) will provide a wide complement of input and output modalities for effective, complex user/system interaction.

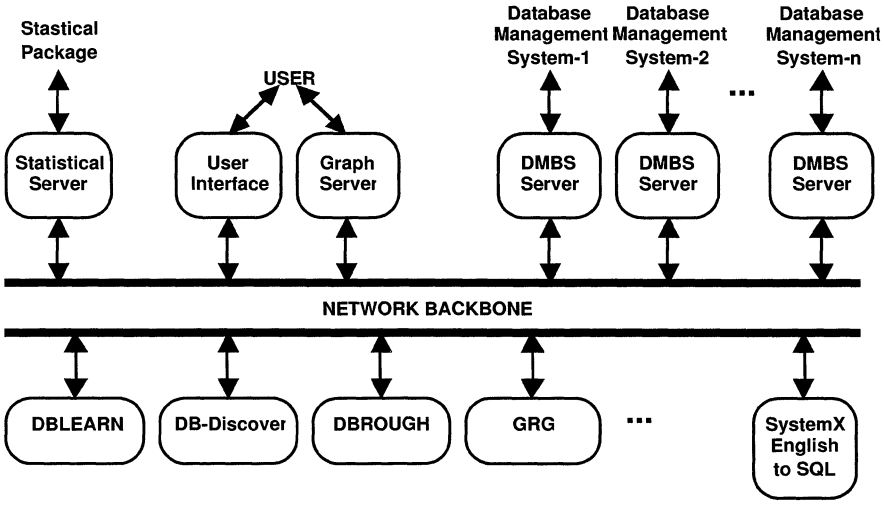


Figure 4. A distributed client-server architecture for data mining programs.

Acknowledgments

The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Centres of Excellence Program of the Government of Canada, the Natural Sciences and Engineering Research Council, and the participation of PRECARN Associates Inc. We are grateful to Canadian Cable Labs Fund and Paradigm Consultants Ltd. for financial assistance, and to the University of Regina for their facilities.

References

Cai, Y., Cercone, N., & Han, J. (1991) Attribute oriented induction in relational databases, in G. PiatetskyShapiro (ed.) **Knowledge Discovery in Databases**, AAAI Press, 213-228.

Carter, C., and Hamilton, H. (1994) Performance improvement in the implementation of DBLEARN, TR CS-94-05, University of Regina, Canada.

Dietterich, T., & Michalski, R. (1983) A comparative review of selected methods for learning from examples, in Michalski et al. (ed) **Machine Learning: An Artificial Intelligence Approach** 1, Morgan Kaufmann, 41-82.

Han, J., Cai, Y., & Cercone, N. (1992) Knowledge discovery in databases: An attribute-oriented approach, *Proc. of the 18th VLDB Conference*, Vancouver, Canada.

Han, J., Cai, Y., & Cercone, N. (1993) Data-driven discovery of quantitative rules in relational databases. **IEEE Trans. Knowledge and Data Engineering**, 5(1), 29-40.

Hu, X., Cercone, N., & Han, J. (1993) An attribute-oriented rough set approach for knowledge discovery in databases, *International Workshop On Rough Sets & Knowledge Discovery (RSKD-93)*, Banff, 79-94.

Hu, X., and Cercone, N. (1994) Learning in relational databases: a rough set approach, *3rd AI & Math.*

Kira, K., & Rendell, L. (1992) The feature selection problem: traditional methods and a new algorithm, *AAAI-92*, MIT Press, 129-134.

Matheus, C., Chan, P., & Piatetsky-Shapiro, G. (1993) Systems for knowledge discovery in databases, **IEEE Trans. Knowledge and Data Engineering**, 5(6), 903-913.

Pawlak, Z. (1982) Rough Sets, **Information and Computer Science**, 11(5), 341-356.

Pawlak, Z. (1991) Rough Sets, **Theoretical Aspects of Reasoning About Data**, Kluwer Academic.

Press, W., Flannery, S., Teukolsky, S., & Vetterling, W. (1988) **Numerical recipes in C: the art of scientific computing**, Cambridge University Press.

Shan, N., Hu, X., Ziarko, W., & Cercone, N. (1994) A generalized rough set model, *PRICAI-94*, Beijing, 437-443.

Wong, M., & Ziarko, W. (1985) On optimal decision rules in decision tables, *Bulletin of the Polish Academy of Sciences*, 33(11-12), 693-696.

Ziarko, W. (1993) Variable precision rough set model, **Computer and System Sciences**, 46(1), 39-59.

Ziarko, W. & Shan, N. (1995) Knowledge discovery as a search for classification, Workshop on Rough Sets and Database Mining, 23rd Annual Computer Science, *CSC'95*.

12

NEIGHBORHOODS, ROUGH SETS, AND QUERY RELAXATION IN COOPERATIVE ANSWERING

James B. Michael* and T. Y. Lin**

** Advanced Vehicle Control Systems Group,*
California Partners for Advanced Transit and Highways Program,
University of California at Berkeley,
1357 South 46th Street, Richmond, California 94804-4698

*** Department of Mathematics and Computer Science,*
San Jose State University, One Washington Square,
San Jose, California 95192-0103

ABSTRACT

We present a mathematical treatment of query relaxation based on the notions of neighborhoods and rough sets. In a relational database management system, for each query term, the user provides the system a list of related terms, called a *neighborhood.* A relational database management system engine then performs queries using related terms in the neighborhood. A neighborhood in this sense is a binary relation without further axioms. Since the relational database mananagement system engine can not perform transitive closure, the resulting list of relaxed queries is the closure (or upper approximation) in the theory of neighborhood systems. In contrast to relational database management systems, the underlying logic model for first-order deductive database systems provides for transitive closure and therefore guarantees the transitive closure of a *neighborhood*, the resulting list of which is precisely the upper approximation in rough set theory.

1 INTRODUCTION

Collaborative computing is characterized by interaction between two or more parties in order to perform a problem-solving task. The parties can consist of a mixture between humans and computer process; such a paradigm is commonly referred to as computer-support for cooperative work (CSCW). Alternatively, all of the parties can be computer processes, acting as proxies for humans; an

example of this type of proxy is a software intelligent agent.

In this paper we focus on a particular class of collaborative computing known as cooperative answering. We define *cooperative answering* to be two or more parties acting together to answer a query posed by one or more of these parties. One technique used in conjunction with cooperative answering is query relaxation. Query relaxation involves rewriting the query terms to form a new query. Some of the reasons for relaxing a query include the following:

- *The query is too general or too specific.* For example, the query may include the terms "adaptive," "control," and "system," resulting in the retrieval of data about many different types of adaptive control systems, when the purpose of the query was to create a view of the data limited to adaptive systems for longitudinal control of an automobile.
- *One or more of the query terms are not in the database or data dictionary.* For example, a query about "vehicles" may be unsuccessful because the terms for "vehicle" used in a particular accident reporting database always refer to a specific class of vehicle, such as "light-duty passenger vehicles," "transit buses," and "heavy articulated trucks."
- *One or more of the query terms have multiple meanings.* For example, the semantics of the concept "system lag" differ between an adaptive control system and that of a signal processing system.

Let $party_a$ be the party that poses the initial query $q_{initial}$, $party_b$ be the party that receives $q_{initial}$, $\triangle$ be the set of databases available to these parties to search for the answer to $q_{initial}$, $q_{relaxed_i}$ denote the ith relaxation of $q_{initial}$, and $q_{result_{initial}}$ and $q_{result_{relaxed_i}}$ be the results from processing $q_{initial}$ and $q_{relaxed_i}$, respectively.

We define *query relaxation* for cooperative answering as an iterative process which involves the following steps:

- $party_a$ submits $q_{initial}$ to $party_b$.
- $party_b$ processes $q_{initial}$ against $\triangle$ and returns $q_{result_{initial}}$ to $party_a$.
- $party_a$ requests assistance from $party_b$ to relax $q_{initial}$.
- $party_a$ and $party_b$ generalize, specialize, or eliminate terms in $q_{initial}$, resulting in $q_{relaxed_1}$.

- $party_a$ submits $q_{relaxed_1}$ to $party_b$.
- $party_b$ processes $q_{relaxed_1}$ against $\triangle$ and returns $q_{result_{relaxed_1}}$ to $party_a$.
- If desired by $party_a$, $q_{relaxed_1}$ is further relaxed i times, with the last result returned by $party_b$ being $q_{result_{relaxed_i}}$.

At each iteration, the set of query terms T in $q_{relaxed_i}$ is either a superset, subset, or disjoint set of $q_{relaxed_{i-1}}$.

The cooperative answering systems developed by Chu and Chen [5] and Gaasterland [6, 7] implement this type of collaboration. The two systems, however, are based on different data models—the former on the Relational Data Model and the latter on the a deductive logic model.

We now introduce the mathematical basis for implementing cooperative answering using the notions of neighborhoods and rough sets.

2 NEIGHBORHOODS AND ROUGH SETS

The concept of neighborhood systems is studied in the theory of topological spaces or more generally Frechet spaces [14]. Intuitively, neighborhood systems handle the notions of *close to, analogous to,* and *approximate to.* Such a notion does not necessitate a transitive relation for all possible cases. For example, let P be a set of places known as "East Los Angeles," "Downtown Los Angeles," and "West Los Angeles." Let R be the relation on P defined "x is close to y," denoted by $x \overset{close\ to}{\longrightarrow} y$.

$$EastLosAngeles \overset{close\ to}{\longrightarrow} DowntownLosAngeles \tag{1}$$

$$WestLosAngeles \overset{close\ to}{\longrightarrow} DowntownLosAngeles \tag{2}$$

$$EastLosAngeles \overset{close\ to}{\not\longrightarrow} WestLosAngeles \tag{3}$$

In this example *is close to* is not an equivalence relation since East Los Angeles is not close to West Los Angeles.

In contrast, in the following example the notion of neighborhood must be transitive since the terms "Greater San Jose," "South Bay," and "Silicon Valley" are assumed to be synonyms for each other:

$$GreaterSanJose = SouthBay \tag{4}$$

$$GreaterSanJose = SiliconValley \tag{5}$$

$$SouthBay \overset{analogous\ to}{\longrightarrow} GreaterSanJose \tag{6}$$

$$SiliconValley \overset{analogous\ to}{\longrightarrow} GreaterSanJose \tag{7}$$

$$SouthBay \overset{analogous\ to}{\longrightarrow} SiliconValley \tag{8}$$

Approximation in rough set theory is based on transitive neighborhoods, namely the Pawlak topology. For relational database management systems, interactive query relaxation must be based on neighborhood systems, since transitive closure is not supported by the Relational Data Model. To support a transitive neighborhood system, a "pre-compiled" or non-interactive approach has to be adopted; these are known as goal query or approximate retrieval [3, 8, 9, 11, 13, 4]. In contrast, deductive database management systems explicitly support transitive neighborhoods.

A neighborhood system is the primitive notion in topological spaces, or more generally Frechet (V) spaces, and has been formulated in logic [1]. From a computational perspective, a neighborhood system is an association that associates each datum with a list of data, of which the list data structure can be processed. The notion has been used in databases [8, 9, 11, 4, 13], studied implicitly in rough set theory, and recently defined in the context of evolutionary computing [2]. The notion of neighborhoods covers the whole spectrum of generalized rough sets that are based on various forms of modal logic or binary relations [12]. If we require the collection of all lists to be pairwise disjoint, namely, the collection is a partition of the data, then the theory of neighborhoods becomes rough set theory.

For references, we recall some basic definitions about neighborhood systems . Let U be the collection of data, and x a datum. A neighborhood of x, denoted by $\mathrm{N}(x)$, is a non-empty subset of U (a list of data) that may or may not contain x. Any subset that contains a neighborhood is a neighborhood. A neighborhood system of x, denoted by $\mathrm{NS}(x)$, is a non-empty maximal family of neighborhoods of x. A neighborhood system of U, denoted by $\mathrm{NS}(U)$, is the collection of $\mathrm{NS}(x)$ for all x in U. If a neighborhood system $\mathrm{NS}(U)$ satisfies certain axioms, U is a topological space. For a neighborhood system $\mathrm{NS}(U)$ without any extra axioms, U is a Frechet (V) space. Examples of neighborhood systems are covering and partition of an universe.

One can interpret a covering of U as a neighborhood system $\mathrm{NS}(U)$ by taking all covers of x as $\mathrm{NS}(x)$. So some x may have several neighborhoods. A partition of U is a special covering, where each datum x has only one neighborhood and all neighborhoods are pairwise disjoint. A **basic neighborhood** is the minimal neighborhood of a point. We are interested in the case that every datum has a basic neighborhood. Let R be a binary relation, and x is a datum. Define $\mathrm{N}(x) = y|yRx$, then $\mathrm{N}(x)$ is a basic neighborhood for x.

Let X be a subset of U. A lower approximation of X in a neighborhood system is defined as

$$\begin{aligned} I(X) &= \{x \mid \exists \mathrm{N}(x)[\mathrm{N}(x) \subseteq X]\} \\ &= \text{interior of } X, \end{aligned} \tag{9}$$

and a upper approximation of X is defined as

$$\begin{aligned} C(X) &= \{x \mid \forall \mathrm{N}(x)[\mathrm{N}(x) \cap X \neq \emptyset]\} \\ &= \text{closure of } X. \end{aligned} \tag{10}$$

If the neighborhood system of U forms a partition, then $I(X)$ and $C(X)$ is the upper and lower approximation of rough sets. Let U and V be spaces with neighborhood systems. There is a natural neighborhood system, called the product neighborhood, in the Cartesian product $U \times V$:

$$\begin{aligned} &\mathrm{NS}(U \times V) = \\ &\quad \{\mathrm{N}((x,y)) = \mathrm{N}(x) \times \mathrm{N}(y) \mid (x,y) \in U \times V\}. \end{aligned} \tag{11}$$

We call $U \times V$ a neighborhood product space. A neighborhood of a tuple in databases is a product neighborhood of each element.

Let R be an equivalence relation and Q/R be the quotient set. There is a natural neighborhood system on Q/R under the natural projection $P : U \longrightarrow$

Q/R. We denote such a neighborhood system on Q/R by $(Q/R, P(\mathrm{NS}(U)))$. The quotient set Q/R with such a neighborhood systems is called the space of concepts, classification space, or simply quotient space. The neighborhood is defined as follows: Let $x \in U$ such that $P(x) = y$. For $y \in Q/R$, we will take a subset $\mathrm{N}(y)$ as a neighborhood of y, if $F^{-1}(\mathrm{N}(y))$ is a neighborhood of x. The family of all such $\mathrm{N}(y)$'s forms a neighborhood system $\mathrm{NS}(Q/R)$.

Let S be a subset of U. There is a natural neighborhood system in S defined as follows: For $x \in S$, we will take $\mathrm{N}(x) \cap S$ as a neighborhood of x in S.

Using the neighborhood system of subsets and quotients, we can derive the neighborhood system of the rule base that consists of rules extracted from a Pawlak information system (e.g., a relation in a database). From such a neighborhood system of rule bases we can apply approximate reasoning [10].

3 QUERY RELAXATION EXAMPLE

Suppose there exists a deductive database containing data—stated as facts and relationships—about the location of heavy articulated trucks carrying hazardous cargo. The database contains the following data:

vehicle(t101).
vehicle(t102).
vehicle(t103).
place(Greater_San_Jose).
place(South_Bay).
place(Silicon_Valley).
located_at(truck_1, Greater_San_Jose).
located_at(truck_2, South_Bay).
located_at(truck_3, Silicon_Valley).
close_to(truck_4, Silicon_Valley).
close_to(truck_5, San_Jose).

Table 1 Database of facts and relationships.

That is, there are five trucks numbered 1 through 5. The names "Greater San Jose," "South Bay," and "Silicon Valley" are places (i.e., locations). Assume that there is an integrity constraint that every truck has a location.

Now suppose that $party_a$ poses $q_{initial}$: *Which vehicles are located in Silicon Valley?* located_at(X, Silicon_Valley).

The deductive DBMS will respond with the following answer "truck 3." This type of answer exemplifies a non-collaborative computing paradigm; that is, there are other trucks located in Silicon Valley but the database management system (i.e., $party_b$) in this case does not assist the user in modifying the query to obtain additional data.

In contrast, a collaborative system can, for instance, provide the user with information about the application domain, database schema (e.g., the predicate names and relationships), or data definition or manipulation language. Similarly, by providing feedback on search terms, for example, the user's response assists the DBMS in narrowing or broadening the search for an answer to the user's query, as demonstrated by Gaasterland [6].

Now suppose that $party_b$ employs query relaxation and that $party_b$ has access to the following relations in a thesaurus:

synonym(Greater_San_Jose, South_Bay).
synonym(South_Bay, Greater_San_Jose).
synonym(Silicon_Valley, Greater_San_Jose).
synonym(Greater_San_Jose, Silicon_Valley).

Table 2 Synonyms defined in a thesaurus.

These are read "X is a synonym for Y." Given this information, $party_b$ could rewrite $q_{initial}$ as $q_{relaxed_1}$ as

located_at(X, Silicon_Valley).
located_at(X, South_Bay).
located_at(X, Greater_San_Jose).

Table 3 The relaxed query, $q_{relaxed_1}$.

and answer the query with "trucks 1, 2, and 3." This represents a form of "horizontal" relaxation in that related terms are substituted for the original terms.

Now we introduce the query relaxation rule *The variable dependency "located at" can be rewritten as "close to:"* relax(located_at(X,Y), close_to(X,Y)).

If we apply this rule, $q_{relaxed_1}$ is generalized to the following:

close_to(X, Silicon_Valley).
close_to(X, South_Bay).
close_to(X, Greater_San_Jose).
located_at(X, Silicon_Valley).
located_at(X, South_Bay).
located_at(X, Greater_San_Jose).

Table 4 The relaxed query, $q_{relaxed_2}$.

$party_b$ will reply "trucks 1, 2, 3, 4, and 5." $party_a$ could have specialized (i.e., the opposite of generalization) the query. Both generalization and specialization are forms of vertical query relaxation.

In this example the relaxation rule represents a form of approximate reasoning and can be implemented using the notion of an indiscernibility function based on families of equivalence classes. The term rewriting based on substitution of predicates, constants, and variable dependencies can be replaced with operations based on the derivation of the lower and upper approximation of the neighborhood system or rough sets.

4 CONCLUSIONS

Neighborhoods and rough sets provide a mathematical basis for assessing the relationship between query terms and query relaxation terms. Rough sets give us a more complete relaxation of query terms than relaxation based on neighborhoods. Moreover, formal languages based on the Relational Data Model do not support transitive closure, so neighborhoods must be used to support such a model when relaxation is done on the "fly." In deductive database systems, one may want to use both.

REFERENCES

[1] Engesser. K, Some connections between topological and modal logic, *Mathematical Logic Quarterly*, **41**, pp. 49–64, 1995.

[2] Back, T., *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, 1996.

[3] Bairamian, S., *Goal Search in Relational Databases*, thesis, California State Univeristy-Northridge, 1989.

[4] Chu, W.W., Neighborhood and associative query answering, *Journal of Intelligent Information Systems*, **1**, pp. 355–382, 1992.

[5] Chu, W. W., and Q. Chen, A structured approach for cooperative query answering, *IEEE Transactions on Knowledge and Data Engineering*, **6(5)**, pp. 738–749.

[6] Gaasterland, T., *Generating Cooperative Answers in Deductive Databases*, Ph.D. dissertation, University of Maryland, College Park, Maryland, 1992.

[7] Gaasterland, T., Restricting query relaxation through user constraints, in *Proceedings of the International Conference on Intelligent and Cooperative Information Systems*. Huhns, M.; Papazoglou, M. P.; and Schlageter, G., eds. Los Alamitos, California, IEEE Computer Society Press, 1993, pp. 359–366.

[8] Lin, T. Y. and S. Bairamian, Neighborhood systems and goal queries, unpublished manuscript, California State University, Northridge, 1987.

[9] Lin, T. Y., Neighborhood systems and relational databases, abstract, in *Proceedings of Annual Computer Science Conferences*, CSC '88, February 1988.

[10] Lin, T. Y., Neighborhood systems and approximation in database and knowledge base systems. Poster session paper presented at the *Colloquium of the Fourth International Symposium on Methodologies of Intelligent Systems*, 1989.

[11] Lin, T.Y., Q. Liu, and K.J. Huang, Rough sets, neighborhood systems and approximation, in *Fifth International Symposium on Methodologies of Intelligent Systems, Selected Papers*, 1990.

[12] Yao, Y.Y. and Lin, T.Y., Generalization of rough sets using modal logic. To appear in *Intelligent Automation and Soft Computing.*

[13] Motro, A., Supporting Goal Queries in Relational Databases, in *Expert Database Systems: Proceedings of the First International Conference*, L. Kerschberg, ed., Institute of Information Management, Technology and Policy, University of South Carolina, 1986, pp. 85–96.

[14] Sierpenski, W. and Krieger, C.,*General Topology*, University of Toronto, Toronto, 1956.

13

RESOLVING QUERIES THROUGH COOPERATION IN MULTI-AGENT SYSTEMS

Zbigniew W. Ras

Univ. of North Carolina, Dept. of Comp. Sci., Charlotte, N.C. 28223, USA

Polish Academy of Sciences, Inst. of Comp. Sci., 01-237 Warsaw, Poland

ras@mosaic.uncc.edu or ras@wars.ipipan.waw.pl

ABSTRACT

Traditional query processing provides exact answers to queries. It usually requires that users fully understand the database structure and content to issue a query. Due to the complexity of the database applications, incorrect or incompletely specified queries are frequently posed and the users often receive no answers or they might need more information than they have received. In this paper a multi-agent system called a cooperative knowledge-based system (CKBS) is presented to rectify these problems.

Key Words: intelligent information system, cooperative query answering, rough sets, multi-agent system, knowledge discovery.

1 INTRODUCTION

By a multi-agent system or simply a cooperative knowledge-based system (CKBS) we mean a collection of autonomous knowledge-based systems called agents (sites) which are capable of interacting with each other. These agents work for one another in problem solving according to their respective abilities. Each agent is represented by an information system (collection of data) and a dictionary (collection of rules). In [10], we proposed a strategy for generating rules from an information system S. This set of rules is sound in S and it remains sound for certain extensions of S. In contrary, a set of rules generated from an information system under closed world assumption (see [8], [18]) is usually not sound if we extend the system. Rules can be seen as rough descriptions of

some attribute values in terms of other attribute values. These descriptions are not precise and they provide only lower and upper approximations of attribute values. We assume here that additional rules can be provided by experts and added, if necessary, to appropriate dictionaries. Clearly, a dictionary built that way have a small chance to be consistent (its rules might be contradictory). The problem of repairing such rules is investigated in this paper. CKBS is locally sound if all its dictionaries are consistent. Any site of CKBS can be a source of a local or a global query. By a local query (reachable) for a site i we mean a query entirely built from values of attributes local for i. Local queries need only to access the information system of the site where they were issued and they are completely processed on the system associated with that site. In order to resolve a global query for a site i (built from values of attributes not necessarily local for i) successfuly, we have to access local dictionaries or/and information systems at other sites of CKBS because of the need of additional data. Moreover, we assume that CKBS is locally sound and it has to remain sound if we update any of its sites.

Cooperative knowledge-based systems have been investigated by [3],[7], [5], and many others. In our paper by CKBS we mean, similarly as S.M. Deen in [6], a multi-agent system. Agents must cooperate in order to resolve locally unreachable queries. At the same time, following Gaasterland [7], we assume that cooperative answer is a correct, nonmisleading, and useful answer to a query. To be more precise, we assume that each contribution (rule created by one of the agents of CKBS) should be:

- Locally valid. Saying another words, an agent (site) at CKBS should only create rules that he believes to be true. Elements of dictionaries satisfy that criterion,
- Relevant to an agent of CKBS which has requested that rule. We mean here that rules should be described in a language which is local, if possible, for a receiving site.

In this paper, we define a global query language (language of locally unreachable queries) for CKBS, give its local interpretation (by one of the agents of CKBS) and provide a sound and complete set of axioms and rules.

2 BASIC DEFINITIONS

In this section, we introduce the notion of an information system, a distributed information system, a dictionary, and $s(i)$-terms which are called local for a site i. We introduce the notion of a rough rule and show the process of building dictionaries.

Information system is defined as a sequence (X, A, V, h), where X is a finite set of objects, A is a finite set of attributes, V is the set-theoretical union of domains of attributes from A, and h is a classification function which describes objects in terms of their attribute values. We assume that:

- $V = \bigcup\{V_a : a \in A\}$ is finite,
- $V_a \cap V_b = \emptyset$ for any $a, b \in A$ such that $a \neq b$,
- $h : X \times A \longrightarrow V$ where $h(x, a) \in V_a$ for any $x \in X$, $a \in A$.

Let $S_1 = (X_1, A_1, V_1, h_1)$, $S_2 = (X_2, A_2, V_2, h_2)$ be information systems. We say that S_2 is a subsystem of S_1 if $X_2 \subseteq X_1$, $A_2 \subseteq A_1$, $V_2 \subseteq V_1$ and $h_2 \subseteq h_1$.

We use a table-representation of a classification function h which is naturally identified with an information system $S = (X, A, V, h)$. For instance, let us assume that $S_2 = (X_2, A_2, V_2, h_2)$ is an information system where $X_2 = \{a_1, a_6, a_8, a_9, a_{10}, a_{11}, a_{12}\}$, $A_2 = \{C, D, E, F, G\}$ and $V_2 = \{e_1, e_2, e_3, f_1, f_2, g_1, g_2, g_3, c_1, c_2, d_1, d_2\}$. Additionally, we assume that $V_E = \{e_1, e_2, e_3\}$, $V_F = \{f_1, f_2\}$, $V_G = \{g_1, g_2, g_3\}$, $V_C = \{c_1, c_2\}$, and $V_D = \{d_1, d_2\}$. Then, the function h_2 defined by Table 1 is identified with an information system S_2.

X_2	F	C	D	E	G
$a1$	$f1$	$c1$	$d2$	$e1$	$g1$
$a6$	$f2$	$c1$	$d2$	$e3$	$g2$
$a8$	$f1$	$c2$	$d1$	$e3$	$g1$
$a9$	$f2$	$c1$	$d1$	$e3$	$g1$
$a10$	$f2$	$c2$	$d2$	$e3$	$g1$
$a11$	$f1$	$c2$	$d1$	$e3$	$g2$
$a12$	$f1$	$c1$	$d1$	$e3$	$g1$

Table 1 Information System S_2

By a distributed information system [16] we mean a pair $DS = (\{S_i\}_{i \in I}, L)$ where:

- $S_i = (X_i, A_i, V_i, h_i)$ is an information system for any $i \in I$,
- L is a symmetric, binary relation on the set I,
- I is a set of sites.

Systems S_{i1}, S_{i2} (or sites $i1, i2$) are called neighbors in a distributed information system DS if $(i1, i2) \in L$. The transitive closure of L in I is denoted by L^+.

A distributed information system $DS = (\{S_i\}_{i \in I}, L)$ is consistent if:

- $(\forall i)(\forall j)(\forall x \in X_i \cap X_j)(\forall a \in A_i \cap A_j)[(x, a) \in Dom(h_i) \cap Dom(h_j) \longrightarrow h_i(x, a) = h_j(x, a)]$.

We assume that all sites in DS share a factual knowledge, so our distributed system is consistent.

Now, we introduce the notion of a dictionary D_{ki}, $(k, i) \in L^+$, containing rules describing values of attributes from $A_k - A_i$ in terms of values of attributes from $A_k \cap A_i$ (see [16]). We begin with definitions of $s(i)$-terms, $s(i)$-formulas and their standard interpretation M_i in a distributed information system $DS = (\{S_j\}_{j \in I}, L)$, where $S_j = (X_j, A_j, V_j, h_j)$ and $V_j = \bigcup\{V_{ja} : a \in A_j\}$, for any $j \in I$.

By a set of $s(i)$-terms we mean a least set T_i such that:

- $\mathbf{0}, \mathbf{1} \in T_i$,
- $w \in T_i$ for any $w \in V_i$,
- if $t_1, t_2 \in T_i$, then $(t_1 + t_2), (t_1 * t_2), \sim t_1 \in T_i$.

We say that:

- $s(i)$-term t is *primitive* if it is of the form $\prod\{w : w \in U_i\}$ for any $U_i \subseteq V_i$,

- $s(i)$-term is in *disjunctive normal form* (DNF) if $t = \sum\{t_j : j \in J\}$ where each t_j is primitive.

Let $t1 = \sum\{t1_j : j \in J1\}$, $t2 = \sum\{t2_k : k \in J2\}$ be $s(i)$-terms in DNF. We say that:

- $t1$ is a subterm of $t2$ if $(\forall j)(\exists k)[Set(t1_j) \subseteq Set(t2_k)]$ where $Set(\prod\{w : w \in U_i\}) = \{w : w \in U_i\}$,
- $t1$ is a proper subterm of $t2$ if $t1$ is a subterm of $t2$ and $t1 \neq t2$.

By a set of $s(i)$-formulas we mean a least set F_i such that:

- if $t_1, t_2 \in T_i$, then $(t_1 = t_2), (t_1 \neq t_2), (t_1 \leq t_2) \in F_i$.

Elements in T_i and F_i represent queries local for a site i. Following Grzymala-Busse [8] we call them reachable in S_i or simply i-reachable.

For example, a DNF query

select $*$ from *Flights*
where *airline* = "*Delta*"
and *departure_time* = "*morning*"
and *departure_airport* = "*Charlotte*"

is reachable in a database

Flights(*airline*, *departure_time*, *arrival_time*, *departure_airport*, *arrival_airport*).

Standard interpretation M_i of $s(i)$-terms and $s(i)$-formulas in a distributed information system $DS = (\{S_j\}_{j \in I}, L)$ is defined as follows:

- $M_i(\mathbf{0}) = \emptyset$, $M_i(\mathbf{1}) = X_i$

- $M_i(w) = \{x \in X_i$: if $w \in V_{ia}$ then $w = h_i(x, a)\}$ for any $w \in V_i$,
- if t_1, t_2 are s(i)-terms, then

$$M_i(t_1 + t_2) = M_i(t_1) \cup M_i(t_2),$$
$$M_i(t_1 * t_2) = M_i(t_1) \cap M_i(t_2),$$
$$M_i(\sim t_1) = X_i - M_i(t_1).$$
$$M_i(t_1 = t_2) = (\text{ if } M_i(t_1) = M_i(t_2) \text{ then } T \text{ else } F)$$
$$M_i(t_1 \neq t_2) = (\text{ if } M_i(t_1) \neq M_i(t_2) \text{ then } T \text{ else } F)$$
$$M_i(t_1 \leq t_2) = (\text{ if } M_i(t_1) \subseteq M_i(t_2) \text{ then } T \text{ else } F)$$

where T stands for $True$ and F for $False$

Let $DS1 = (\{S1_j\}_{j \in I}, L1)$, $DS2 = (\{S2_j\}_{j \in I}, L2)$ be distributed information systems and $DS1$ is a subsystem of $DS2$. Now, assume that $S1_i = (X1_i, A_i, V_i, h1_i)$, $S2_i = (X2_i, A_i, V_i, h2_i), i \in I$ and M_i, N_i are standard interpretations of terms and formulas in $DS1$, $DS2$, respectively. If $X1_i \subset X2_i$ and $h2_i|(X_i \times A_i) = h1_i$, then N_i is called a standard extension of M_i.

By (k, i)-rule in $DS = (\{S_j\}_{j \in I}, L)$, $k, i \in I$, we mean a triple (c, t, s) such that:

- $c \in V_k - V_i$,
- t, s are $s(k)$-terms in DNF and they both belong to $T_k \cap T_i$,
- $M_k(t) \subseteq M_k(c) \subseteq M_k(t + s)$.

We say that (k, i)-rule (c, t, s) is in k-reduced form if there are no other $s(k)$-terms $t_1, s_1 \in T_k \cap T_i$, both in DNF such that:

- $M_k(t) \subset M_k(t_1) \subseteq M_k(c)$ where $M_k(t) \neq M_k(t_1)$ and $t1$ is a proper subterm of t,
- $M_k(s) = M_k(s_1)$ and $s1$ is a proper subterm of s.

We say that (k, i)-rule (c, t, s) is in k-optimal form if there are no other $s(k)$-terms $t_1, s_1 \in T_k \cap T_i$, both in DNF such that:

- $M_k(t) \subset M_k(t_1) \subseteq M_k(c)$ and $M_k(t) \neq M_k(t_1)$,

- $M_k(-(t+s)) \subset M_k(s_1) \subseteq M_k(-c)$ and $M_k(-(t+s)) \neq M_k(s_1)$.

Now, we show the relationship between rough sets introduced by Pawlak (see [12]) and (k,i)-rules.

Theorem 1 Let $S_k = (X_k, A_k, V_k, h_k)$, $S_i = (X_i, A_i, V_i, h_i)$. If (c,t,s) is a (k,i)-rule in k-optimal form and $S_k^\star = (X_k, A_i \cap A_k, V_i \cap V_k, h_k)$, then:

- $M_k(c)$ belongs to a rough set in (X_k, ρ) (see [8]), where ρ is the indiscernibility relation on X_k induced by $S_k^\star$,
- $M_k(t)$ is its lower approximation,
- $M_k(t+s)$ is its upper approximation.

Proof. It follows directly from the definition of a rough set, or more precisely from its lower and upper approximation. □

For any (k,i)-rule (c,t,s) in $DS = (\{S_j\}_{j \in I}, L)$, we say that:

- $(t \rightarrow c)$ is a k-certain rule in DS,
- $(t+s \rightarrow c)$ is a k-possible rule in DS.

Now, we introduce the notion of a strong consistency of (k,i)-rules in the interpretation M_k. So, let us assume that $r_1 = (c_1, t_1, s_1)$, $r_2 = (c_2, t_2, s_2)$ are (k,i)-rules. We say that: r_1, r_2 are strongly consistent in M_k, if either c_1, c_2 are values of two different attributes in S_k or $N_k(t_1 * t_2) = 0$ for any standard extension N_k of the interpretation M_k.

Now, we are ready to introduce the notion of a dictionary D_{ki}. Its elements can be seen as approximate descriptions of values of attributes from $V_k - V_i$ in terms of values of attributes from $V_k \cap V_i$. To be more precise, we assume that D_{ki} is a set of (k,i)-rules such that: if $(c,t,s) \in D_{ki}$ and $t_1 =\sim (t+s)$ is true in any standard extension N_i of the interpretation M_i then $(\sim c, t1, s) \in D_{ki}$.

Dictionary D_{ki} is in k-reduced form (k-optimal form) if all its (k,i)-rules are in k-reduced form (k-optimal form).

Dictionary D_{ki} is strongly consistent in M_k if any two rules in D_{ki} are strongly consistent in M_k.

Let us assume that a distributed information system $DS = (\{S_i\}_{i\in\{1,2\}}, L)$ has two sites, one represented by Table 1 and the second by Table 2. We assume here that $V_B = \{b1, b2\}$, $V_C = \{c1, c2, c3\}$, $V_D = \{d1, d2\}$, and $V_E = \{e1, e2, e3\}$.

X_2	B	C	D	E
$a1$	$b1$	$c1$	$d1$	$e1$
$a2$	$b1$	$c2$	$d1$	$e2$
$a3$	$b2$	$c3$	$d2$	$e1$
$a4$	$b1$	$c2$	$d1$	$e2$
$a5$	$b1$	$c3$	$d1$	$e3$
$a6$	$b2$	$c1$	$d2$	$e3$
$a7$	$b2$	$c2$	$d1$	$e2$

Table 2 Information System S_1

We show first how to construct a dictionary D_{12} in 1-reduced form. The following rules can be computed directly from the information system S_1: $(b1, c1 * d1 * e1 + c3 * d1 * e3, c2 * d1 * e2)$, $(b2, c3 * d2 * e1 + c1 * d2 * e3, c2 * d1 * e2)$. So, the dictionary D_{12} in 1-reduced form may contain the following $(1, 2)$-rules: $(b1, d1 * e1 + d1 * e3, e2)$, $(b2, d2, e2)$.

Now, we give a hint how to built a dictionary D_{12} which is strongly consistent in M_1. First, we start with rules computed directly from the information system S_1: $(b1, c1 * d1 * e1 + c3 * d1 * e3, c2 * d1 * e2)$, $(b2, c3 * d2 * e1 + c1 * d2 * e3, c2 * d1 * e2)$. Next, we optimize them. The optimization process for rules, built at any site i from the data in S_i, is based on the following principle:

Any two rules $(m1, t1, s1)$, $(m2, t2, s2)$ can be generalized at site i to $(m1, t1^*, s1^*)$, $(m2, t2^*, s2^*)$ if $m1, m2$ are values of the same attribute and for any information system S_j and the interpretation M_j in S_j where S_i is a subsystem of S_j we have:

- $M_j(t1^*) \cap M_j(t2^*) = \emptyset$,

- $M_j(t1) \subseteq M_j(t1^*)$ and $M_j(t2) \subseteq M_j(t2^*)$,
- $M_j(s2) \subseteq M_j(s2^*)$ and $M_j(s1) \subseteq M_j(s1^*)$,
- $M_j(t1^*) \cap M_j(s1^*) = \emptyset$ and $M_j(t2^*) \cap M_j(s2^*) = \emptyset$,
- $M_j(t1^*) \cap M_j(s2^*) = \emptyset$ and $M_j(t2^*) \cap M_j(s1^*) = \emptyset$.

It can easily be checked that a dictionary D_{12} which is strongly consistent in M_1 contains, for example, the following rules: $(b1, d1 * e1 + d1 * e3, d1 * e2)$, $(b2, d2, d1 * e2)$.

Dictionary D_{ki} is built at site k and some of its elements (rules), if needed, can be sent to the site i of a distributed information system $DS = (\{S_j\}_{j \in I}, L)$, for any $k, i \in I$) (see [10], [16]). Dictionary D_{ki} will represent beliefs of agent k at site i. Elements of dictionaries D_{kj}, $k \in J$ sent to site i can be stored there. This way site i knows beliefs of all agents from J. Clearly, all these beliefs may form a set at site i which is inconsistent.

3 COOPERATIVE KNOWLEDGE-BASED SYSTEM

In this section we define a Cooperative Knowledge Based System (CKBS) based on dictionaries and introduce the notion of its local and global consistency.

Let $\{D_{ki}\}_{k \in K_i}$, $K_i \subseteq I$, be a collection of dictionaries where D_{ki} is created at site $k \in I$ for any $k \in K_i$ and $D_i = \bigcup\{D_{ki} : k \in K_i\} \cup R_i$. By R_i we mean a set of rules (c, t, s) created by an expert and presented to the site i. Clearly, an expert believes here that $(t \rightarrow c)$ is a certain rule and $(t + s \rightarrow c)$ is a possible one. Additionally, we assume that t, s are $s(i)$-terms. System $(\{S_i, D_i\}_{i \in I}, L)$, introduced in [16], is called a cooperative knowledge based system (CKBS). We say that $(\{S_i, D_i\}_{i \in I}, L)$ is S_i-consistent if for any two rules $(w, t1, s1), (w, t2, s2) \in D_i$, $i \in I$, we have $M_i(t1) \subseteq M_i(t2 + s2)$. System $(\{S_i, D_i\}_{i \in I}, L)$ is consistent if it is S_i-consistent for any $i \in I$ and if $(\{S_i\}_{i \in I}, L)$ is consistent.

Let us clarify the notion of S_i-consistency. So, assume that $r1 = (c, t1, s1) \in D_{ki}$ and $r2 = (c, t2, s2) \in D_{mi}$. It means that $(t1 \rightarrow c)$ and $(c \rightarrow t2 + s2)$ are m-certain rules. Saying another words, site k of CKBS knows that all its objects satisfying property $t1$ have also propert c and all its objects satisfying c

have also property $t1 + s1$. Similarly, site m of CKBS knows that all its objects satisfying property $t2$ have also property c and all its objects satisfying c have also property $t2 + s2$. On this basis, site i may assume that both rules are true at i unless they are in *conflict*. What do we mean by a conflict? Clearly, the value c is not reachable (foreign) for a site i. It means that our definition of rules in a conflict has to be based entirely on the semantical relationship between terms $t1, s1, t2, s2$ at site i.

We say that rules $r1, r2$ are in a conflict at site i, if $[M_i(t1) \subset M_i(t2 + s2)$ and $M_i(t1) \neq M_i(t2 + s2)]$ or $[M_i(t2) \subset M_i(t1 + s1)$ and $M_i(t2) \neq M_i(t1 + s1)]$. The condition $[M_i(t1) \subset M_i(t2 + s2)$ and $M_i(t1) \neq M_i(t2 + s2)]$ is equivalent to $M_i(t1* \sim (t2 + s2)) \neq \emptyset$. Similarly, the condition $[M_i(t2) \subset M_i(t1 + s1)$ and $M_i(t2) \neq M_i(t1 + s1)]$ is equivalent to $M_i(t2* \sim (t1 + s1)) \neq \emptyset$. So, rules $r1, r2$ are in a conflict at site i, if the interpretation M_i is a model of the formula $(t1* \sim (t2 + s2)) + (t2* \sim (t1 + s1)) \neq 0$. Rules which are not in a conflict at site i will be called *sound* at i.

Let $(\{S_i, D_i\}_{i \in I}, L)$ be a cooperative knowledge based system, $a \in A_k \cap A_m$ and $c \in V_a$. We say that systems S_k, S_m are in a conflict (disagreement) on c if $[\exists x \in X_k \cap X_m][c = h_k(x, a) \neq h_m(x, a)]$.

Now, we are ready to state the following question: Can we repair rules which are in a conflict at one of the sites of CKBS? What we mean here is to redefine, if possible, the rule $r1$ in S_k and $r2$ in S_m in a such a way that they are no longer in a conflict. To answer this question, we propose the notion of *repairable* rules.

Let us assume that the rules $r1 = (c, t1, s1) \in D_{ki}$, $r2 = (c, t2, s2) \in D_{mi}$ are in a conflict at sie i which means that $M_i(t1* \sim (t2 + s2)) \neq \emptyset$ or $M_i(t2* \sim (t1 + s1)) \neq \emptyset$. We say that these rules are repairable at site i if systems S_k, S_m are not in a conflict on c.

The repair process for rules $r1 = (c, t1, s1) \in D_{ki}$, $r2 = (c, t2, s2) \in D_{mi}$, repairable at site i, is outlined below.

Let us assume that $M_i(t1* \sim (t2 + s2)) \neq \emptyset$ and that the term $p_1 + p_2 + ... + p_j$ is semanticaly equivalent to $t1* \sim (t2 + s2)$ where each p_i is a conjunct of values of attributes from A_i. By semanticaly equivalent, we mean that $M_i(t1* \sim (t2 + s2)) = M_i(p_1 + p_2 + ... + p_j) = M_i(p_1) + M_i(p_2) + ... + M_i(p_j)$. Let $x \in M_i(p_1)$. So, x has a property c and $\sim c$ at site i. Now, we have two options:

- $x \in X_k - X_m$ or $x \in X_m - X_k$,
- $x \notin X_k \cup X_m$.

If $x \in X_k - X_m$, then the rule $r2 = (c, t2, s2)$ is replaced by a new rule $r2^* = (c, t2, s2 + p_1)$ in D_{mi}.
If $x \in X_m - X_k$, then the rule $r1 = (c, t1, s1)$ is replaced by a new rule $r1^* = (c, t1 * (\sim p_1), s1)$ in D_{ki}.
Clearly, if $x \notin X_k \cup X_m$ then the rule $r2 = (c, t2, s2)$ in D_{mi} is replaced by a new rule $r2^* = (c, t2, s2 + p_1)$ and the rule $r1 = (c, t1, s1)$ is replaced by a new rule $r1^* = (c, t1 * (\sim p1), s1)$ in D_{ki}. We have to repeat the same process for terms $p_2, p_3, ..., p_j$. If $M_i(t2* \sim (t1 + s1)) \neq \emptyset$, the repair process is similar.

To give an example of a CKBS let us assume for simplicity reason that our system has only two sites $SITE1$ and $SITE2$ and two information systems associated with them are represented by Figure 2. Dictionaries D_{21}, D_{12} are added to the sites $SITE1$ and $SITE2$, respectively. Rules in a dictionary D_{21} are computed at $SITE2$ and rules in a dictionary D_{12} at $SITE1$. The resulting CKBS is shown in Figure 1. We assume here that $(1, 2)$-rules have been requested by $SITE2$ and added to the dictionary D_{12} if needed.

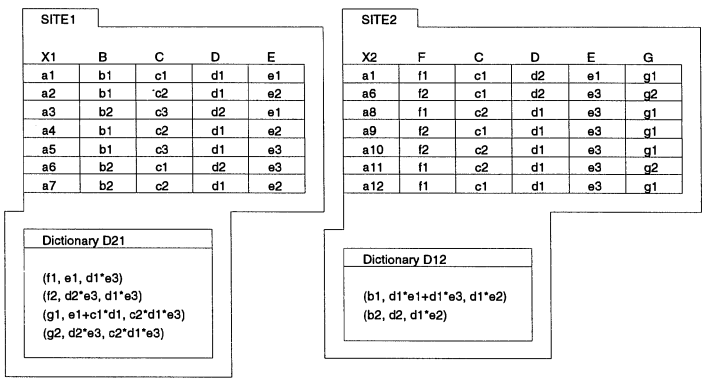


Figure 1 DKBS with two sites SITE1 and SITE2

4 QUERY LANGUAGE AND ITS INTERPRETATION.

In this section we introduce a (global) query language and propose a class of its local interpretations (called standard) at one of the sites of CKBS. Next, for this class of local interpretations we give a complete and sound set of axioms and rules.

Standard interpretation M_i, introduced in Section 3, shows how to interpret local queries in a CKBS. Now, we propose how to interpret queries (called global or locally unreachable) built from values of attributes belonging to a superset of V_i. We begin with definitions of DS-terms, DS-formulas and their standard interpretation in S_i-consistent cooperative knowledge based system $DS = (\{(S_j, \{D_{nj}\}_{n \in K_j})\}_{j \in I}, L)$, where $S_i = (X_i, A_i, V_i, f_i)$ for any $i \in I$. To simplify our notation, we will write S instead of S_i and assume that $V = V_i = \bigcup\{V_{ia} : a \in A_i\}$ and $C_S = \bigcup\{V_j : j \in I\} - V$. Elements in C_S are called concepts or unreachable values of attributes for the site i in DS.

For example, SQL query

select $*$ from *Flights*
where *airline* = "*Delta*"
and *departure_time* = "*morning*"
and *departure_airport* = "*Charlotte*"
and *cost* = "*low*"

would be unreachable in a database

Flights(*airline*, *departure_time*, *arrival_time*, *departure_airport*, *arrival_airport*)

because of the term cost = "low".

So, coming back to our formal definitions, let us assume that a query language $L(DS, C_S)$ is a sequence (A, T, F), where A is an alphabet, T is a set of terms, and F is a set of formulas.

The alphabet A of $L(DS, C_S)$ contains:

- constants: w where $w \in V_i \cup C_S$
- constants: $\mathbf{0}, \mathbf{1}$
- functors: $+$, $*$, $\sim$
- predicate: $=$
- connectives: $\vee, \wedge, \sim, \Rightarrow$
- auxiliary symbols: $(,)$.

The set of terms T is a least set such that:

- if w is a constant, then w is a term
- if t_1, t_2 are terms, then $(t_1 + t_2)$, $(t_1 * t_2)$, $(\sim t_1)$ are terms.

Parentheses are used, if necessary, in the obvious way. As will turn out later, the order of a sum or product is immaterial. So, we will abbreviate finite sums and products as $\sum\{t_j : j \in J\}$ and $\prod\{t_j : j \in J\}$, respectively. Intentionally, terms are names of certain features of parts being processed by CKBS, more complex than those expressed by constants.

The set of formulas F is a least set such that:

- if t_1, t_2 are terms, then $(t_1 = t_2)$ is a formula, and
- if α, β are formulas, then $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, $(\alpha \Rightarrow \beta)$, $(\sim \alpha)$ are formulas.

Let M_i be a standard interpretation of $s(i)$-terms and $s(i)$-formulas in $DS = (\{S_j\}_{j \in I}, L)$.

By S-standard interpretation of queries from $L(DS, C_S)$ in S-consistent cooperative knowledge based system $(\{S_j, \{D_{kj}\}_{k \in K_j})\}_{j \in I}, L)$, where $S = (X_i, A_i, V_i, h_i)$ and $V_i = \bigcup\{V_{ia} : a \in A_i\}$, we mean the interpretation M_{i,K_i} such that:

- $M_{i,K_i}(\mathbf{0}) = \emptyset$, $M_{i,K_i}(\mathbf{1}) = X_i$
- for any $w \in V_i$,

$$M_{i,K_i}(w) = M_i(w),$$
$$M_{i,K_i}(\sim w) = X_i - M_{i,K_i}(w)$$

- for any $w \in C_S$,

$$M_i(w) = \{x \in X_i : (\exists n \in K_i)(\exists t, s)([w, t, s] \in D_{ni} \wedge x \in M_i(t))\}$$
$$M_i(\sim w) = \{x \in X_i : (\exists n \in K_i)(\exists t, s)([w, t, s] \in D_{ni} \wedge x \notin M_i(s))\}$$

- for any terms $t_1, t_2 \in T$

$$M_{i,K_i}(t_1 + t_2) = M_{i,K_i}(t_1) \cup M_{i,K_i}(t_2),$$
$$M_{i,K_i}(t_1 * t_2) = M_{i,K_i}(t_1) \cap M_{i,K_i}(t_2),$$
$$M_{i,K_i}(\sim (t_1 + t_2)) = (\sim M_{i,K_i}(t_1)) \cap (\sim M_{i,K_i}(t_2)),$$
$$M_{i,K_i}(\sim (t_1 * t_2)) = (\sim M_{i,K_i}(t_1)) \cup (\sim M_{i,K_i}(t_2)),$$
$$M_{i,K_i}(\sim\sim t) = M_{i,K_i}(t).$$

- for any terms $t_1, t_2 \in T$

$$M_{i,K_i}(t_1 = t_2) = (\text{ if } M_{i,K_i}(t_1) = M_{i,K_i}(t_2) \text{ then } T \text{ else } F)$$
where T stands for $True$ and F for $False$

- for any formulas $\alpha, \beta \in F$

$$M_{i,K_i}(\alpha \vee \beta) = M_{i,K_i}(\alpha) \vee M_{i,K_i}(\beta)$$
$$M_{i,K_i}(\alpha \wedge \beta) = M_{i,K_i}(\alpha) \wedge M_{i,K_i}(\beta)$$
$$M_{i,K_i}(\alpha \Rightarrow \beta) = M_{i,K_i}(\alpha) \Rightarrow M_{i,K_i}(\beta)$$
$$M_{i,K_i}(\sim \alpha) = \sim M_{i,K_i}(\alpha)$$

From the point of view of the site i the interpretation M_{i,K_i} represents a pessimistic approach to query evaluation. It means that $M_{i,K_i}(t)$ is interpreted as a set of objects in X_i which have the property t for sure. We are not retrieving here objects which might have property t.

Let us adopt the following set A of Axiom Schemata:

A1. Substitutions of the axioms of distributive lattices for terms and the axioms of equality
A2. $\sim w * w = \mathbf{0}$ for any constant w
A3. $\sim w + w = \mathbf{1}$ for any $w \in V_i$
A4. for each $w \in V_i$ there is a subset $w_1, w_2, ..., w_n$ of V_i such that $\sim w = w_1 + w_2 + ... + w_n$
A5. $v_1 * v_2 = 0$ if $v_1, v_2 \in V_{ia}$ for some $a \in A_i$
A6. for any term t, $\sim \mathbf{0} = \mathbf{1}$, $\sim \mathbf{1} = \mathbf{0}$, $\mathbf{1} + t = \mathbf{1}$, $\mathbf{1} * t = t$, $\mathbf{0} * t = \mathbf{0}$, $\mathbf{0} + t = t$, $\sim\sim t = t$
A7. for any $w \notin V_i$ $w = \sum\{t : [w, t, s] \in D_{ki} \wedge k \in K_i\}$
A8. for any $w \notin V_i$ $\sim w = \sum\{t : [\sim w, t, s] \in D_{ki} \wedge k \in K_i\}$
A9. $\sim (t_1 + t_2) = (\sim t_1) * (\sim t_2)$
A10. $\sim (t_1 * t_2) = (\sim t_1) + (\sim t_2)$
A11. Substitutions of the propositional calculus axioms

The rules of inference for our formal system are the following:

R1. from $(\alpha \Rightarrow \beta)$ and α we can deduce β for any formulas α, β
R2. from $t_1 = t_2$ we can deduce $t(t_1) = t(t_2)$, where $t(t_1)$ is a term containing t_1 as a subterm and $t(t_2)$ comes from $t(t_1)$ by replacing some of the occurences of t_1 with t_2.

We write $A \vdash \alpha$ if there exists a derivation from a set A of formulas as premises to the formula α as the conclusion.

We write $A \models \alpha$ to denote the fact that A semantically implies α, that is, for any S-standard interpretation M_{i,K_i} of $L(DS, C_S)$ in S-consistent cooperative knowledge based system we have $M_{i,K_i}(\alpha) = T$.

Let us adopt the following definitions:

- A term t is called *simple* if it is of the form $\prod\{w_b : c \in A_i\}$, where $w_b \in V_{ib}$ for all $b \in A_i$.
- A term t is of *standard form* if $t = \sum\{t_j : j \in J\}$ where each t_j is simple and all t_j's are different.
- A formula is *elementary* if it has the form $(t = F)$ where t is a simple term.

- A formula is *primitive* if it has the form $\prod\{\alpha_j : j \in J\}$ where each α_j is either an elementary formula or the negation of an elementary formula.
- A formula is in *normal disjunctive form* if it is of the form $\sum\{\beta_j : j \in J\}$ where each β_j is primitive.
- A formula $\prod\{\alpha_j : j \in J\}$ is *basic* if it is primitive and all the elementary formulas occur in it, each one exactly once.
- A formula is in *standard form* if it has the form $\sum\{\beta_j : j \in J\}$ where each β_j is basic and all $\beta_j's$ are different.
- A formula is *positive* if connectives " $\Rightarrow, \sim$ " do not occur in it.

Theorem 2 (Soundness). For any formula α, if $A \vdash \alpha$ then $A \models \alpha$.

Proof. Our S-standard interpretation was defined in such a way as to make the axioms $A1, A3, A4, A6, A7, A8, A9, A10, A11$ true. Axiom $A2$ holds because the system is consistent. Axiom $A5$ holds because of the definition of an information system. Both rules of inference preserve validity under the standard interpretation. □

Theorem 3 For each term t there is a term s in a standard form such that $A \vdash (t = s)$.

Proof. Axioms $A1, A2, A3, A4, A6, A7, A8, A9, A10$ are applied to find term $t1 = \sum\{t1_j : j \in J\}$ in normal disjunctive form such that $\vdash (t = t1)$. Now, if two different $w1, w2 \in V_{ia}$ occur in $t1_j$ for some $a \in A_i$ and $j \in J$, we have $\vdash (t1_j = F)$ because of axioms $A5, A6$. Therefore, we can derive term $t2 = \sum\{t2_j : j \in J\}$ in normal disjunctive form such that each $t2_j$ is simple and $\vdash (t1 = t2)$. Let $a \in A_i$ be an attribute such that no $w \in V_{ia}$ occurs in $t1_j$ for soem $j \in J$. Using $\vdash (t1_j = t1_j * T)$ (axiom $A6$), we get $\vdash (t1_j = t1_j * \sum\{w : w \in V_{ia}\}$ (axioms $A3, A4$) and, by the distributive law (axiom $A1$), $\vdash (t1_j = \sum\{t1_j * w : w \in V_{ia}\}$. Thus we have diminished by one the number of attributes which were not represented in $t1_j$. A repeated application of the above procedure completes the proof. Clearly, finiteness of V_{ia} and A_i for any $i \in I$, $a \in A_i$ is essential. □

Theorem 4 For each atomic formula $(t = s)$ there is a provably equivalent positive primitive formula.

Proof. By applying Theorem 3 we transform t and s to standard form. Assume now that $t_1, t_2, t_3, ..., t_k$ are simple terms occuring in either t or s but not in both. Multiplying both sides of $(t = s)$ by $t_1, t_2, t_3, ..., t_k$ we obtain
$\vdash (s = t) \Rightarrow (t_1 = F) \wedge (t_2 = F) \wedge ... \wedge (t_k = F)$. □

Theorem 5 For each formula α there is provably equivalent formula β in standard form.

Proof. By applying Theorem 4 we replace in α each equality of terms by a positive primitive formula. Then, by using the propositional calculus axioms we obtain formula α_1 in normal disjunctive form provably equivalent to α. Now, to obtain β from α_1, it is sufficient to exploit the theorem
$(((t = F) \vee \sim (t = F)) \Rightarrow T) \wedge (T \Rightarrow ((t = F) \vee \sim (t = F)))$ where t is a term.
□

Theorem 6 (Completeness). For any formula α, if $A \models \alpha$ then $A \vdash \alpha$.

Proof. The set of formulas A is equivalent to a single formula Φ, in the sense that $A \vdash \Phi$ and for every $\beta \in A$, $\Phi \vdash \beta$. Formula Φ can be constructed by transforming each $\beta \in A$ into the standard form, then deleting all repeating formulas and taking the conjunction of the remaining finite number of formulas. We can assume (see Theorem 5) that both Φ and α are in standard form. Suppose that $A \models \alpha$ and $non(A \vdash \alpha)$. It means that there is a basic formula Φ_o occuring in Φ but not in α. Assume that $\Phi_o = \prod\{\sim (t = F) : t \in T^+\} \wedge \prod\{(t = F) : t \in T^-\}$. Now, we take information system $S2_i = (X_i, A_i, V_i, h_i)$, $V_i = \bigcup\{V_{ij} : j \in A_i\}$ such that:

- $X_i = \{h \in \bigotimes\{V_{ij} : j \in A_i\} : \prod\{h(j) : j \in A_i\} \in T^+\}$, where $\bigotimes$ denotes cartesian product of sets.
- $h_i(w) = \{g \in X_i : g(j) = w$ for the unique j such that $w \in V_{ij}\}$.

Clearly $M_{i,K_i}(\Phi) = T$ and $M_{i,K_i}(\alpha) = F$, where $K_i = \emptyset$. This contradicts the assumption that $A \models \alpha$. □

Our query answering system is retrieving objects only if queries (terms) are conjuncts. The above completeness theorem gives us the set of axioms which is sound and sufficient to transform any global query to its equivalent DNF.

5 CONCLUSION

This paper presents a methodology and theoretical foundations of a cooperative knowledge-based system (CKBS) which is partially implemented at UNC-Charlotte on a cluster of SPARC 2 workstations. Our query answering system of CKBS identifies all k-unreachable attributes used in a query entering site k. Next it sends a message to all its neighbours that rules approximating these k-unreachable attributes are needed. This message invokes at each neighboring site a program similar LERS (see [8]) which computes rules describing k-unreachable attributes in terms of k-reachable ones. Finally, these rules are sent to site k and used by the query answering system to replace k-unreachable values of attributes in a query by k-reachable terms.

REFERENCES

[1] Bazan, J., Skowron, A, Synak, P., "Dynamic reducts as atool for extracting laws from decision tables", in *Methodologies for Intelligent Systems, Proceedings of the 8th International Symposium* (ed. Z. Ras, M. Zemankova), Lecture Notes in Artificial Intelligence, Springer Verlag,No. 869, 1994, 346-355

[2] Bosc, P., Pivert, O., "Some approaches for relational databases flexible querying", in *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 1, 1992, 355-382

[3] Chu, W.W., "Neighborhood and associative query answering", in *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 1, 1992, 355-382

[4] Chu, W.W., Chen, Q., Lee, R., "Cooperative query answering via type abstraction hierarchy", in *Cooperating Knowledge-based Systems* (ed. S.M. Deen), North Holland, 1991, 271-292

[5] Cuppers, F., Demolombe, R., "Cooperative answering: a methodology to provide intelligent access to databases", in *Proceedings 2nd International Conference on Expert Database Systems*, Virginia, USA, 1988

[6] Deen, S.M., "A general framework for coherence in a CKBS", in *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 2, 1993, 83-107

[7] Gaasterland, T., Godfrey, P., Minker, J., "An overview of cooperative answering", *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 1, 1992, 123-158

[8] Grzymala-Busse, J., *Managing uncertainty in expert systems*, Kluwer Academic Publishers, 1991

[9] Kacprzyk, J., "On measuring the specificity of if-then rules", in *International Journal of Approximate Reasoning*, Vol. 11, No. 1, 1994, 29-53

[10] Maitan, J., Ras, Z.W., Zemankova, M., "Query handling and learning in a distributed intelligent system", in *Methodologies for Intelligent Systems*, 4, (ed. Z.W. Ras), North Holland, 1989, 118-127

[11] Nakamura, A., "On rough logic based on incomplete knowledge systems", in *Proceedings of the Third International Workshop on Rough Sets and Soft Computing*, Society for Computer Simulation, 1994, 36-39

[12] Pawlak, Z., "Rough Sets - theoretical aspects of reasoning about data", Kluwer Academich Publishers, 1991

[13] Pawlak, Z., "Rough sets and decision tables", in *Proceedings of the Fifth Symposium on Computation Theory*, Springer Verlag, Lecture Notes in Computer Science, Vol. 208, 1985, 118-127

[14] Pawlak, Z., "Mathematical foundations of information retrieval", *CC PAS Reports*, No. 101, Warsaw, 1973

[15] Ras, Z.W., "Dictionaries in a distributed knowledge-based system", in *Proceedings of Concurrent Engineering: Research and Applications Conference*, Pittsburgh, August 29-31, 1994, Concurrent Technologies Corporation, 383-390

[16] Ras, Z.W., "Query processing in distributed information systems", in *Fundamenta Informaticae Journal*, Special Issue on Logics for Artificial Intelligence, IOS Press, Vol. XV, No. 3/4, 1991, 381-397

[17] Ras, Z., Chilumula, N., "Answering queries by cooperative knowledge based system", in *Proceedings of the Third International Workshop on Rough Sets and Soft Computing*, Society for Computer Simulation, 1994, 263-266

[18] Skowron, A., "Boolean reasoning for decision rules generation", in *Methodologies for Intelligent Systems, Proceedings of the 7th International Symposium on Methodologies for Intelligent Systems*, (eds. J. Komorowski, Z. Ras), Lecture Notes in Artificial Intelligence, Springer Verlag, No. 689, 1993, 295-305

14

SYNTHESIS OF DECISION SYSTEMS FROM DATA TABLES

Andrzej Skowron, Lech Polkowski*

Institute of Mathematics, Warsaw University,
Banacha 2, 02-097 Warsaw, Poland,
e-mail:skowron@mimuw.edu.pl

** Institute of Mathematics, Warsaw University of Technology*
Pl. Politechniki 1, 00-650 Warsaw, Poland
e-mail: polk@mimuw.edu.pl

ABSTRACT

We discuss two basic questions related to the synthesis of decision algorithms.

The first question can be formulated as follows: what strategies can be used in order to discover the decision rules from experimental data? Answering this question, we propose to build these strategies on the basis of rough set methods and Boolean reasoning techniques. We present some applications of these methods for extracting decision rules from decision tables used to represent experimental data.

The second question can be formulated as follows: what is a general framework for approximate reasoning in distributed systems? Answering this question, we assume that distributed systems are organized on rough mereological principles in order to assembly (construct) complex objects satisfying a given specification in a satisfactory degree. We discuss how this approach can be used for building the foundations for approximate reasoning. Our approach is based on rough mereology, the recently developed extension of mereology of Leśniewski.

1 INTRODUCTION

Different aspects of theory of decision systems are extensively investigated (see e.g. [13], [19], [20], [22], [23], [24], [25], [29], [42], [47], [51]). We adopt here the point of view that decision systems are built as hierarchies of teams of

intelligent agents, and we discuss some logical tools for synthesis of this kind of systems. Our approach is computationally efficient.

The main control parameters which are adjusted by agents belonging to a system are [29], [47]: an information function (defined on objects with values being attribute value vectors), a similarity (tolerance) relation between information (attribute value) vectors, and a strategy for conflict resolution among possible decisions for a given information vector.

Any particular agent (or team of agents) is realizing its local goal by means of decision rules extracted from low level knowledge represented by decision tables [25]. Adjustment of information function is related to information reduction (see e.g. [25], [42]) or feature extraction (see e.g. [3-4], [22], [45]). We present several applications of rough set methods [25] and Boolean reasoning techniques [8] for extracting the decision rules from decision tables. In particular we discuss exemplary methods: dynamic reducts and rules [3-4]; stable coverings by dynamic reducts [6]; feature extraction, in particular quantization of real value attributes [45] and automatic synthesis of features for structural objects [5]; boundary region thinning [41], [53]; decision rules [26], [38-40], [47] and approximate rules [21], [47] generation; data filtration [43]; tolerance reducts [46] and absorbents [50].

We point out the role of tolerance (similarity) relation for extracting laws from decision tables and for composing information from different agents. In particular we discuss the problem of information reduction in tolerance information systems. The Boolean reasoning can be applied to reduce the set of attributes as well as the set of objects. The reduced sets of attributes are called (relative) tolerance reducts and the reduced sets of objects are called absorbents. We also outline a general scheme for decision function approximation with strategies for conflict resolution between possible decisions for a given information vector.

Boolean reasoning and rough set methods are the basic low-level building blocks. Using these methods, we explain connections to other approaches for reasoning with uncertainty e.g. Dempster-Shafer theory of evidence [36], [44] and fuzzy sets [9], machine learning and pattern recognition (feature extraction, decision rules generation, and clustering), [20], [22], mathematical morphology (data filtration) [35], knowledge representation and modelling of complex systems [23-24], [29]. The results are applied to construct tools for extracting decision rules from experimental data. The effectiveness of these tools has been proven in applications to market data analysis, medical diagnosis, handwritten digit recognition or synthesis of real-time decision algorithms (see [3], [4], [5], [47], [48]).

One of the main problems concerning distributed systems of cooperating intelligent agents is related to the construction of a general framework for approximate reasoning about the system behaviour and performance. We have proposed an approach based on rough mereology introduced in [30-31] and extended in [15], [33] to systems of intelligent agents. We discuss how this approach can be used for building a foundational basis for approximate reasoning in and about distributed systems. In a nutshell, for a given specification, a system of intelligent agents [33] is organized to assemble a complex object satisfying a given specification to a satisfactory degree. The constructed complex object may also be interpreted as a proof in which approximate inference rules are used. This proof supports a specified belief (specification) to some degree. We are convinced that the approximate rules and more general approximation logic structures (i.e. logics for reasoning under uncertainty) should be extracted from low-level logic by algorithmic tools (e.g. by decomposition of decision tables). Algorithmic methods for extracting approximation logic from low-level knowledge bases create a bridge between logics for reasoning under uncertainty and practical applications. In this way we are building a much-needed connection between theoretical investigations in logic and applications in Artificial Intelligence.

The paper is structured as follows. In Section 2 we present preliminaries of rough set methods and Boolean reasoning methods. Applications of these methods to data reduction and decision rules extraction are discussed in Section 3 and applications to feature extraction are presented in Section 4. Section 5 consists of basic ideas on which our approach to approximate reasoning is developed. We conclude with some suggestions for further research.

The paper summarizes and extends the results presented in [3], [4], [5], [15], [26], [29], [30]-[34], [37]-[47].

2 ROUGH SET AND BOOLEAN REASONING PRELIMINARIES

In this section we recall some basic notions related to information systems and rough sets (for more details see [25]). An *information system* is a pair $\mathbb{A} = (U, A)$, where U is a non-empty, finite set called the *universe* and A — a non-empty, finite set of *attributes*, i.e. $a : U \to V_a$ for $a \in A$, where V_a is called the *value* set of a. Every information system $\mathbb{A} = (U, A)$ and non-empty set $B \subseteq A$ determine a *B-information function* $Inf_B : U \longrightarrow \mathbb{P}(B \times \bigcup_{a \in B} V_a)$

defined by $Inf_B(x) = \{(a, a(x)) : a \in B\}$. The set $\{Inf_A(x) : x \in U\}$ is called the *A-information set* and it is denoted by $INF(\mathbb{A})$. A *decision table* is any information system of the form $\mathbb{A} = (U, A \cup \{d\})$, where $d \notin A$ is a distinguished attribute called the *decision*. The elements of A are called *conditions*. For simplicity of notation we assume that the set V_d of values of the decision d is equal to $\{1, \ldots, r(d)\}$. Let us observe that the decision d determines the partition $\{X_1, \ldots, X_{r(d)}\}$ of the universe U, where $X_k = \{x \in U : d(x) = k\}$ for $1 \leq k \leq r(d)$. The set X_i is called the *i-th* decision class of $\mathbb{A}$.

Let $\mathbb{A} = (U, A)$ be an information system. With every subset of attributes $B \subseteq A$, an equivalence relation, denoted by $IND_{\mathbb{A}}(B)$ (or $IND(B)$) called the *B-indiscernibility relation*, is associated and defined by $IND(B) = \{(x, x') \in U^2$: for every $a \in B$, $a(x) = a(x')\}$. Objects x, x' satisfying relation $IND(B)$ are indiscernible by attributes from B.

Let $\mathbb{A}$ be an information system with n objects. By $M(\mathbb{A})$ we denote an $n \times n$ matrix (c_{ij}), called the *discernibility matrix* of $\mathbb{A}$ such that

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j)\} \quad \text{for} \quad i, j = 1, \ldots, n.$$

A *discernibility function* $f_{\mathbb{A}}$ for an information system $\mathbb{A}$ is a Boolean function of m Boolean variables $a_1^*, \ldots, a_m^*$ corresponding to the attributes $a_1, \ldots, a_m$ respectively, and defined by

$$f_{\mathbb{A}}(a_1^*, \ldots, a_m^*) = \bigwedge \Big\{ \bigvee c_{ij}^* : 1 \leq j < i \leq n,\ c_{ij} \neq \phi \Big\}$$

where $c_{ij}^* = \{a^* : a \in c_{ij}\}$. The set of all *prime implicants* of $f_{\mathbb{A}}$ determines the set of all *reducts* of $\mathbb{A}$ [37].

If $\mathbb{A} = (U, A)$ is an information system, $B \subseteq A$ is a set of attributes and $X \subseteq U$ is a set of objects, then the sets $\{s \in U : [s]_B \subseteq X\}$ and $\{s \in U : [s]_B \cap X \neq \phi\}$ are called *B-lower* and *B-upper approximation* of X in $\mathbb{A}$, and they are denoted by $\underline{B}X$ and $\overline{B}X$, respectively. The set $BN_B(X) = \overline{B}X - \underline{B}X$ will be called the *B-boundary* of X. When $B = A$ we also write $BN_{\mathbb{A}}(X)$ instead of $BN_A(X)$. Sets which are unions of some classes of the indiscernibility relation $IND(B)$ are called *definable* by B. The set X is *B-definable* iff $\underline{B}X = \overline{B}X$. Some subsets (categories) of objects in an information system cannot be expressed exactly by employing available attributes, but they can be defined *roughly*. If $X_1, \ldots, X_{r(d)}$ are decision classes of $\mathbb{A}$ then the set $\underline{B}X_1 \cup \ldots \cup \underline{B}X_{r(d)}$ is called *the B-positive region of* $\mathbb{A}$ and is denoted by $POS(B, \{d\})$.

If $\mathbb{A} = (U, A \cup \{d\})$ is a decision table, then we define a function $\partial_A : U \longrightarrow \mathbb{P}(\{1, \ldots, r(d)\})$, called the *generalized decision in* $\mathbb{A}$, by $\partial_A(x) = \{i : \exists x' \in$

$U\, x' IND(A) x$ and $d(x) = i\}$. A decision table $\mathbb{A}$ is called *consistent (deterministic)* if $|\partial_A(x)| = 1$ for any $x \in U$, otherwise $\mathbb{A}$ is *inconsistent (nondeterministic)*.

A *decision rule* of a decision table $\mathbb{A} = (U, A \cup \{d\})$ is any expression of the form $\tau \Longrightarrow (d, i)$ where $i \in V_d$ and τ is a Boolean combination of descriptors i.e. expressions (a, v) where $a \in A$ and $v \in V_a$. If τ is a Boolean combination of descriptors then by $\tau_{\mathbb{A}}$ we denote the meaning of τ in the decision table $\mathbb{A}$, i.e. the set of all objects in U with the property τ, defined inductively as follows:

(i) if τ is of the form (a, v) then $\tau_{\mathbb{A}} = \{x \in U : a(x) = v\}$;
(ii) $(\tau \wedge \tau')_{\mathbb{A}} = \tau_{\mathbb{A}} \cap \tau'_{\mathbb{A}}$; $(\tau \vee \tau')_{\mathbb{A}} = \tau_{\mathbb{A}} \cup \tau'_{\mathbb{A}}$.

The decision rule $\tau \Longrightarrow (d, i)$ for $\mathbb{A}$ is *true* in $\mathbb{A}$ iff $\tau_{\mathbb{A}} \subseteq ((d, i))_{\mathbb{A}}$; if $\tau_{\mathbb{A}} = ((d, i))_{\mathbb{A}}$ then we say that the rule is $\mathbb{A}$-*exact*.

We now recall some notions introduced in [38]. First we show how to construct a description of the decision classes by exact in $\mathbb{A}$ decision rules $\alpha_i \Longrightarrow (d, i)$ where $\alpha_i \in \mathbb{C}(A, V)$ is a disjunction $\bigvee \bigwedge \gamma_i$ of conjunctions $\bigwedge \gamma_i$ of minimal sets γ_i of descriptors for $i = 1, \ldots, r(d)$. The set γ_i defines in $\mathbb{A}$ a non-empty set of objects, i.e. $(\gamma_i)_{\mathbb{A}} \neq \phi$ and is minimal in the following sense: the decision rule $\bigwedge \gamma'_i \Longrightarrow (d, i)$ is no longer valid in $\mathbb{A}$ for any $\phi \neq \gamma'_i \subset \gamma_i$. A decision rule with the above property is called *minimal with respect to the descriptors in* $\mathbb{A}$. The method allows to generate decision rules with one more property, namely, if $\bigvee \bigwedge \gamma_i \Longrightarrow (d, i)$ is a minimal with respect to descriptors rule in $\mathbb{A}$ and δ is a set of descriptors such that $\bigwedge \delta \Longrightarrow (d, i)$ is valid in $\mathbb{A}$ and $(\bigwedge \delta)_{\mathbb{A}} \neq \phi$ then $\gamma_i \subseteq \delta$ for some i. The decision rules with the above property are called *complete with respect to the descriptors in* $\mathbb{A}$.

The work reported in [38] and [39] gives a simple method that allows to compute, for a given consistent decision table $\mathbb{A}$, the description of all decision classes of $\mathbb{A}$ in the form of decision rules exact in $\mathbb{A}$. These rules are complete and minimal with respect to descriptors.

Let $\mathbb{A} = (U, A \cup \{d\})$ be a consistent decision table and $M'(\mathbb{A}) = (c'_{ij})$ be its relative discernibility matrix. We construct new matrices (columns of the relative discernibility matrix)

$$M(\mathbb{A}, k) = (c'_{jk}) \quad \text{for any} \quad x_k \in U$$

The matrix $M(\mathbb{A}, k)$ is called *the k-relative discernibility matrix of* $\mathbb{A}$. This enables us to construct the *k-relative discernibility function* $f_{M(\mathbb{A},k)}$ of $M(\mathbb{A}, k)$ in

the same way as the discernibility function was constructed from the discernibility matrix. Let $Atr(\tau)$ denote the set of all attributes occurring in the prime implicant τ of $f_{M(\mathbb{A},k)}$ and let $Trace(\mathbb{A},k)$ be the following set of descriptor conjunctions:

$$\left\{\bigwedge\{(a,a(x_k)) : a \in Atr(\tau)\} : \quad \tau \text{ is a prime implicant of } \ f_{M(\mathbb{A},k)}\right\}.$$

Now let α_i for any $i \in \{1,...,r(d)\})$ be a disjunction of all formulas from the set $\cup\{Trace(\mathbb{A},k) : d(x_k) = i\}$.

Proposition 2.1 [38] Let $\mathbb{A} = (U, A \cup \{d\})$ be a consistent decision table. The decision rules:

$$\alpha_i \Longrightarrow (d,i) \quad \text{where} \quad i \in \{1,\ldots,r(d)\}$$

are complete and minimal with respect to descriptors in $\mathbb{A}$. □

3 APPLICATIONS OF ROUGH SET AND BOOLEAN REASONING METHODS FOR DATA REDUCTION AND EXTRACTION OF DECISION RULES

Given an information system (a decision table) $\mathbb{A}$, some strategies are applied to produce decision rules. The primary technique offered by rough set theory has been here the reduct generation. Reducts offer the same classificational ability as the whole system but with a smaller set of attributes. As this approach is insufficient, however, we have implemented additional tools. We discuss in more detail some of the following strategies:

the approximation of reducts [3-4], dynamic reducts and rules [3-4], [5], stable coverings [6], voting strategies [3-4], boundary region thinning [41], [53], data filtration [43], tolerance reducts [46] and absorbents [50].

3.1 Approximation of reducts

The discernibility matrix [37] and the reduct approximation [3-4] provide methods for extracting laws from decision tables. Several strategies for searching for

a subset of the set of discernibility matrix entries sufficient for the generation of laws encoded in the decision table are implemented in our system for object classification. One of the techniques for reduct approximation is based on the approximation of the positive region. The following algorithm computes this kind of reduct approximation.

Algorithm For $R \in RED(\mathbb{A}, d)$ and N equal to the number of objects in $\mathbb{A}$:

Step 1: Calculate positive regions $POS_{R-\{a\}}$ for all $a \in R$.

Step 2: Choose from the reduct R one attribute a_0 satisfying the condition: $\forall a \in R \;\; POS_{R-\{a_0\}} \geq POS_{R-\{a\}}$.

Step 3: **if** $POS_{R-\{a_0\}} > k \cdot N$ (e.g. $k = 0.9$) **then**

begin

$R := R - \{a_0\}$

go to step 1

end

Step 4: The new set of attributes (R) is called the *S-reduct*.

S-reducts can help to extract interesting laws from decision tables. Applying reduct approximation instead of reducts, we slightly decrease the quality of classification of objects from the training set, but we expect to receive more general rules with a higher quality of classification for new objects.

3.2 Boundary region thinning

Objects are classified on the basis of information about them. In a given table we associate with any information vector the distribution of objects corresponding to this vector into decision classes. When this probability distribution is non-uniform, we regard objects corresponding to small values of this distribution as in a sense abnormal or noisy objects. The generalized decision for a given information vector can be then modified by removing from it the decision values corresponding to these small values of the probability distribution. The decision rules generated for the modified generalized decision can give a better quality of classification of new, as yet unseen objects. Various techniques of this approach called boundary region thinning have been proposed in [53] and [41]. Boundary region thinning gives a new decision table to which the methods of synthesis of decision rules discussed in [26], [42] can be applied.

The methods discussed above can be treated as a special case of methods based on a new version of reduct approximation proposed in [49]. Let $\mathbb{A} = (U, A \cup \{d\})$

be a consistent decision table and let $B \subseteq A$. The *conditional entropy of* B in $\mathbb{A}$ is defined by

$$H_{\mathbb{A}}(B) = -\frac{1}{U} \sum_{u \in INF(B)} |u_B| \sum_{i \in I(u)} p_i(u) \log p_i(u)$$

where $u_B = \{x \in U : Inf_B(x) = u\}$, $p_i(u) = |u_B \cap X_i|/|u_B|$, $X_i = \{x \in U : d(x) = i\}$, $I(u) = \{i : p_i(u) > 0\}$. The α-reduct of $\mathbb{A}$ is a minimal set $B \subseteq A$ such that $H_{\mathbb{A}}(B) \leq \alpha$, where α is a non-negative integer. In [49] a genetic algorithm for computing α-reducts is presented. The α-reducts with a properly tuned up parameter α can be applied to generate approximate rules (like default rules).

3.3 Dynamic reducts and rules

We now show an example of communication among cooperating agents working on synthesis of decision algorithms. In this example the information sharing among agents leads to the extraction from the subtables processed by agents of the most stable reducts called *dynamic reducts.*

The underlying idea of dynamic reducts stems from the observation that reducts generated from the information system are unstable in the sense that they are sensitive to changes in the information system introduced by removing a randomly chosen set of objects. The notion of a dynamic reduct encompasses the stable reducts, i.e. reducts that are the most frequent reducts in random samples created by subtables of the given decision table [3-4]. We show here how to compute dynamic reducts from reduct (approximations) and how to generate dynamic rules from dynamic reducts. The dynamic reducts have shown their utility in various experiments with data sets of various kinds e.g. market data [14], monk's problems [19], handwritten digits recognition [5] or medical data [3-4]. The quality of unseen objects classification by decision rules generated from dynamic reducts increases especially when data are very noisy e.g. market data [14]. In all cases we have obtained a substantial reduction of the decision rule set without decreasing the classification quality of unseen objects. The results of experiments with dynamic reducts show that attributes from these reducts can be treated as relevant features [3-4].

In order to capture the fact that some reducts are chaotic, we consider random samples forming subtables of a given decision table $\mathbb{A} = (U, A \cup \{d\})$; we will call a *subtable* of $\mathbb{A}$ any information system $\mathbb{B} = (U', A \cup \{d\})$ such that $U' \subseteq U$.

Let $\mathcal{F}$ be a family of subtables of $\mathbb{A}$ and let ε be a real number from the unit interval $[0,1]$. The set $DR_\varepsilon(\mathbb{A},\mathcal{F})$ of $(\mathcal{F},\varepsilon)$-*dynamic reducts* is defined by

$$DR_\varepsilon(\mathbb{A},\mathcal{F}) = \{C \in RED(\mathbb{A},d) : \frac{|\{\mathbb{B} \in \mathcal{F} : C \in RED(\mathbb{B},d)\}|}{|\mathcal{F}|} \geq 1-\varepsilon\}$$

For $C \in RED(\mathbb{B},d)$, the number $|\{\mathbb{B} \in \mathcal{F} : C \in RED(\mathbb{B},d)\}/|\mathcal{F}|$ is called the *stability coefficient* of C *relative* to $\mathcal{F}$.

We present a technique for computing the dynamic reducts [3-4]. The experiments with different data sets have shown that this type of dynamic reducts allows generating decision rules with better quality of classification of new objects than the other methods. The method consists of the following steps.

Step 1: A random set of subtables is taken from the given table; for example:
10 samples with the size of 90% of the decision table,
10 samples with the size of 80% of the decision table,
10 samples with the size of 70% of the decision table,
10 samples with the size of 60% of the decision table,
10 samples with the size of 50% of the decision table.

Step 2: Reducts for all of these tables are calculated; for example reducts for any of the 50 randomly chosen tables.

Step 3: Reducts with the stability coefficients higher than a fixed threshold are extracted.

These reducts selected in step 3 are regarded as true dynamic reducts.

The processes of decision rules generation based on reduct sets have high computational complexity. For example, the problem of computing a minimal reduct is NP-hard [37], and, therefore, we are forced to apply some approximation algorithms in order to obtain knowledge about reduct sets. One possibility is to use approximation algorithms that do not give an optimal solution but have the acceptable time complexity e.g. algorithms based on simulated annealing and Boltzmann machines, genetic algorithms and algorithms using neural networks. We use these algorithms in experiments for generating a large number of reducts. The other possibility is to use standard computational techniques on modified information systems, e.g. by conceptual clustering of values of attributes or groups of attributes, conceptual clustering of objects and extracting new attributes from existing decision tables.

Dynamic reducts can be computed using approximations of reducts instead of reducts to generate dynamic reducts. If a set of dynamic reducts (with the

stability coefficients greater than a given threshold) has been computed, then it is necessary to decide how to compute the set of decision rules. To this end we have implemented several methods. The first one is based on the (F, ε)-dynamic core of A, i.e. on the set $\bigcup DR_\varepsilon(\mathbb{A}, \mathcal{F})$. We apply the methods based on Boolean reasoning presented in [26], [38] to generate decision rules (with minimal number of descriptors) from conditional attributes belonging to the dynamic core. The second one is based on the decision rule set construction for any chosen dynamic reduct. The final decision rule set is equal to the union of all these sets. In our experiments we have received slightly better results of tests applying the second method. If an unseen object has to be classified then first it is matched against all decision rules from the constructed decision rule set. Next, the final decision is predicted by applying a strategy predicting the final decision from all "votes" of decision rules. The simplest strategy we have tested was the majority voting i.e. the final decision is the one supported by the majority of decision rules. One can also apply fuzzy methods to predict the proper decision.

The idea of dynamic reducts can be adapted to a new method of dynamic rules computation. From a given data table a random set of subtables is chosen. For example:

10 samples with the size 90% of the decision table,
10 samples with the size 80% of the decision table,
10 samples with the size 70% of the decision table,
10 samples with the size 60% of the decision table,
10 samples with the size 50% of the decision table.

Thus we receive 50 new decision tables. Then the decision rule sets for all these tables are calculated. In the next step the rule memory is constructed where all rule sets are stored. Intuitively, the dynamic rule is appearing in all (or almost all) of experimental subtables. The decision rules can also be computed from the so-called local reducts used to generate decision rules with minimal number of descriptors [26].

Several experiments performed with different data tables (see e.g. [3-5]) show that our strategies for synthesis of decision algorithms increase the quality of unseen object classification and/or allow to reduce the number of decision rules without decreasing the classification quality of unseen objects.

Stable coverings by dynamic reducts

If $R_1, \ldots, R_r$ are dynamic reducts with stability coefficients $\alpha_1, \ldots, \alpha_r < 1$, then it is possible to find descriptions $\beta_1, \ldots, \beta_r$ (in terms of conditional attributes) of regions in the set U of training objects such that for any $x \in U$:

if β_i is true on x
then the decision on x produced by decision rules generated from R_i is correct

The formulas $\beta_1, \ldots, \beta_r$ can be synthesized by applying Boolean reasoning methods [6]. These formulas together with the dynamic reducts $R_1, \ldots, R_r$ create a *stable covering* of the set of unseen objects as long as for any such object x satisfying β_i and β_j the decision rules generated from the dynamic reducts R_i and R_j predict the same decision on x. In [6] we present a procedure searching for dynamic reducts together with a stable covering.

Two steps are performed when a new (unseen so far) object x is classified. First by applying rules corresponding to all descriptions β_i some dynamic reducts are pointed out. Next the decision rules corresponding only to these reducts are applied to predict the decision on x. The rules corresponding to regions described by β_i should be constructed by taking into account that they will be used also for new objects, not belonging to the actual universe of objects. Hence some inductive techniques [6] are proposed for constructing the rule sets allowing to select from dynamic reducts those which are appropriate for a given object x.

3.4 Approximate decision rules

We discuss certain consequences for synthesis of the decision rules implied by the relationships between rough set methods and Dempster-Shafer's theory of evidence.

In [44] it has been shown that one can compute a basic probability assignment (bpa) $m_{\mathbb{A}}$ for any decision table $\mathbb{A}$. The bpa $m_{\mathbb{A}}$ satisfies the following conditions:

$$m_{\mathbb{A}}(\emptyset) = 0 \quad \text{and} \quad m_{\mathbb{A}}(\theta) = \frac{|\{x \in U : \partial_A(x) = \theta\}|}{|U|}$$

where $\emptyset \neq \theta \subseteq \Theta_{\mathbb{A}} = \{i : d(x) = i \text{ for some } x \in U\}$.

Hence the relationships between belief functions $\mathrm{Bel}_{\mathbb{A}}$ and $Pl_{\mathbb{A}}$ related to the decision table $\mathbb{A}$ can be proven [44]:

$$\mathrm{Bel}_{\mathbb{A}}(\theta) = \frac{|\underline{A} \bigcup_{i \in \theta} X_i|}{|U|} \quad \text{and} \quad Pl_{\mathbb{A}}(\theta) = \frac{|\overline{A} \bigcup_{i \in \theta} X_i|}{|U|}$$

for any $\theta \subseteq \Theta_{\mathbb{A}}$.

Boolean reasoning can be also applied to generate rules with certainty coefficients from inconsistent decision table $\mathbb{A}$. These certainty coefficients can be expressed by values of belief functions [26].

The belief functions related to decision tables can be applied to generate approximate decision rules. There are at least two reasons why we search for approximate decision rules for some subsets of $\Theta_{\mathbb{A}}$. The first one can be called an economical reason and could be roughly formulated as follows: "A small number of approximate decision rules is preferred to a large number of exact rules", e.g. to be able to take decisions in real-time. The second one is a consequence of the assumption that the values of conditional attributes in decision tables are influenced by noise. This is valid for many real decision tables. In this case one can expect to obtain a better classification quality of unseen objects by using approximate rules which are more general than exact rules computed from noisy data.
One of possible approaches to applying the belief functions to extract some approximate decision rules is to search for solutions of the following problem:

APPROXIMATION PROBLEM (AP)
INPUT: A decision table $\mathbb{A} = (U, A \cup \{d\})$, $\theta \subseteq \Theta_{\mathbb{A}}$ and rational numbers $\varepsilon, tr \in (0, 1]$.
OUTPUT: Minimal (with respect to the inclusion) sets $B \subseteq A$ satisfying two conditions:

(i) $|Pl_{\mathbb{A}|B}(\theta) - \mathrm{Bel}_{\mathbb{A}|B}(\theta)| < \varepsilon$
(ii) $\mathrm{Bel}_{\mathbb{A}|B}(\theta) > tr$.

where $\mathbb{A}|B = (U, B \cup \{d\})$.

The above conditions (i) and (ii) are equivalent to

$$|\overline{B} \bigcup_{i \in \theta} X_i - \underline{B} \bigcup_{i \in \theta} X_i| < \varepsilon |U| \quad \text{and} \quad |\underline{B} \bigcup_{i \in \theta} X_i| > tr|U|$$

respectively. Hence (i) means that the boundary region (with respect to B) corresponding to $\cup_{i \in \theta} X_i$ is "small" (less than $\varepsilon|U|$) and the lower approximation of $\cup_{i \in \theta} X_i$ is "sufficiently large" (greater than $tr|U|$). Hence the solution for the above problem can be obtained by performing the following steps:

Step 1. Change the decision table $\mathbb{A} = (U, A \cup \{d\})$ into $\mathbb{A}' = (U, A \cup \{d_\theta\})$ where $d_\theta(x) = 1$ if $x \in \cup_{i \in \theta} X_i$ and $d_\theta(x) = 0$ otherwise.

Step 2. Compute the relative reducts of $\mathbb{A}'$ [38].

Step 3. By dropping attributes from relative reducts find minimal sets B satisfying (i-ii).

Next, one can compute the decision rules corresponding to $\overline{B} \cup_{i \in \Theta} X_i$ in $\mathbb{A}'' = (U, B \cup \{\partial_B\})$ for any set B received in Step 3 by applying Boolean reasoning [26], [38].

The received rules have a more general form (have a simpler structure) than the exact ones (computed on data influenced by noise). So, they can be better predisposed than the exact rules to recognize properly unseen objects. This approach is used in [21] to generate default rules from decision tables.

3.5 Tolerance information systems

Tolerance relations provide an attractive and general tool for studying indiscernibility phenomena. The importance of those phenomena was already noticed by Poincare and Carnap. Studies by, among others, Menger, Zadeh, and Pawlak have led to the emergence of new approaches to indiscernibility.

We call a relation $\tau \subseteq X \times X$ a *tolerance relation* on X if (i) τ is *reflexive:* $x \tau x$ for any $x \in X$ (ii) τ is *symmetric*: $x \tau y$ implies $y \tau x$ for any pair x, y of elements of X. The pair (X, τ) is called a *tolerance space*. It leads to a metric space with the distance function

$$d_\tau(x, y) = \min\{k : \exists_{x_0, x_1, \ldots, x_k} x_0 = x \wedge x_k = y \wedge (x_i \tau x_{i+1} \text{ for } i = 0, 1, \ldots, k-1)\}$$

Sets of the form $\tau(x) = \{y \in X : x \tau y\}$ are called *tolerance sets*. For $x \in X$, we define the *τ-domain of x*, $\mathrm{dom}_\tau(x)$ by $dom_\tau(x) = \cap\{\tau(z) : x \in \tau(z)\}$. We introduce the *$\tau$-indiscernibility relation* IND_τ by letting $x IND_\tau y$ iff $dom_\tau(x) = dom_\tau(y)$.

The symbol $[x]_\tau$ will stand for the equivalence class of IND_τ containing x. We collect below the basic properties of IND_τ and dom_τ.

Proposition 3.1 [42]

(i) $y \in dom_\tau(x)$ iff $\forall_z [x \in \tau(z) \Rightarrow y \in \tau(z)]$ iff $\tau(x) \subseteq \tau(y)$

(ii) $[x]_\tau \subseteq dom_\tau(x) \subseteq \tau(x)$

(iii) if $y \in \mathrm{dom}_\tau(x)$ then $\mathrm{dom}_\tau(y) \subseteq \mathrm{dom}_\tau(x)$ and $[y]_\tau \subseteq \mathrm{dom}_\tau(x)$

(iv) $\forall_x \exists_{\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_m} [x]_\tau = \mathrm{dom}_\tau(x_1)^{\varepsilon_1} \cap \mathrm{dom}_\tau(x_2)^{\varepsilon_2} \cap \ldots \cap \mathrm{dom}_\tau(x_m)^{\varepsilon_m}$
where $X = \{x_1, x_2, \ldots, x_m\}$, $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m \in \{0,1\}$ and $A^0 = X - A$, $A^1 = A$. □

There are two main parameters which can be controlled in the adaptive process described above: the *information function Inf* (related essentially to the relevant feature extraction and information reduction), and a tolerance relation τ allowing to measure the degree of similarity of information vectors. The choices of τ and *Inf* have a direct impact on the decision algorithm. A general scheme of decision algorithm approximating the decision function is based on the notion of a domain of a tolerance relation discussed above. The *decision algorithm* $\mathsf{A} = \mathsf{A}(\mathcal{C}, v, S)$ is based on two parameters:

- $\mathcal{C} = \{\mathrm{dom}_\tau(x_1), \mathrm{dom}_\tau(x_2), \ldots, \mathrm{dom}_\tau(x_k)\}$ is a set of τ-domains
- $v = \{n_1, n_2, \ldots, n_k\}$ is a vector of distances (of a given domain) from domains $\mathrm{dom}_\tau(x_1), \mathrm{dom}_\tau(x_2), \ldots, \mathrm{dom}_\tau(x_k)$

and strategy S producing from $\mathcal{C}$ and v the final decision.
In case of measurement, the set $\mathcal{C}$ is endowed with an ordering relation and corresponds to the scale; in case of control decision, algorithm $\mathcal{C}$ corresponds to regions of states with decisions known from experience ; in case of data classification, the set $\mathcal{C}$ corresponds to the cluster set.

By a *tolerance information system* [46] we understand a triple $\mathbb{A}' = (U, A, \tau)$ where $\mathbb{A}' = (U, A)$ is an information system and τ is a tolerance relation on *information vectors* $Inf_B(x) = \{(a, a(x)) : a \in B\}$ where $x \in U$, $B \subseteq A$. In particular, a tolerance information system can be realized as a pair $(\mathbb{A}, D)$ where $\mathbb{A} = (U, A)$ is an information system, while $D = (D_B)_{B \subseteq A}$ and $D_B \subseteq INF(B) \times INF(B)$ is a relation, called *the discernibility relation*, satisfying the following conditions:

(i) $INF(B) \times INF(B) - D_B$ is a tolerance relation;

(ii) $((u - v) \cup (v - u) \subseteq (u_0 - v_0) \cup (v_0 - u_0))$ & $uD_Bv \rightarrow u_0D_Bv_0$ for any $u, v, u_0, v_0 \in INF(B)$ i.e. D_B is monotonic with respect to the discernibility property;

(iii) $non(uD_Cv)$ implies $non(u|BD_Bv|B)$ for any $B \subseteq C$ and $u, v \in INF(C)$

where $INF(B) = \{Inf_B(x) : x \in U\}$ and if $u \in INF(C)$ and $B \subseteq C \subseteq A$ then $u|B = \{(a, w) \in u : a \in B\}$ i.e. $u|B$ is the restriction of u to B. A (B, D_B)-tolerance function

$$\tau_B : U \longrightarrow \mathbb{P}(U)$$

is defined by

$$y \in \tau_B(x) \quad \text{iff} \quad non(Inf_B(x) D_B Inf_B(y))$$

for any $x, y \in U$. It defines a tolerance (indiscernibility) relation by

$$y \tau_B x \quad \text{iff} \quad non(Inf_B(x) D_B Inf_B(y)).$$

A (B, D_B)-*tolerance function* $I[B, D_B] : U \longrightarrow \mathbb{P}(U)$ is defined by $I[B, D_B](x) = \tau_B(x)$.

for any $x \in U$. The set $I[B, D_B](x)$ is called *the tolerance set of* x. The relation $INF(B) \times INF(B) - D_B$ expresses similarity of objects in terms of accessible information about them. The set $RED(\mathbb{A}, D)$ is defined by

$$\{B \subseteq A : I[A, D_A] = I[B, D_B] \text{ and } I[A, D_A] \neq I[C, D_C] \text{ for any } C \subset B\}.$$

Elements of $RED(\mathbb{A}, D)$ are called *tolerance reducts of* $(\mathbb{A}, D)$ (or, *tolerance reducts*, in short). It follows from the definition that the tolerance reducts are minimal attribute sets preserving (A, D_A) - tolerance function.

The tolerance reducts of $(\mathbb{A}, D)$ can be constructed in an analogous way to reducts of information systems. More precisely, tolerance reducts of $(\mathbb{A}, D)$ are computable relatively to the family $\{(\mathbb{A}, D)[u, v] : u D_A v\}$ where

$$(\mathbb{A}, D)[u, v] = \{B \subseteq A : u|B \, D_B \, v|B \ \& \ (non(u|C \, D_C \, v|C) \text{ for any } C \subset B)\}$$

Theorem 3.2 [46] *Let* $(\mathbb{A}, D)$ *be a tolerance information system, where* $\mathbb{A} = (U, A)$ *and* $A = \{a_1, \ldots, a_m\}$. *Let* $g_{\mathbb{A},D}$ *be a Boolean function of* m *Boolean variables* $a_1^*, \ldots, a_m^*$ *corresponding to attributes* $a_1, \ldots, a_m$ *and defined by*

$$g_{\mathbb{A},D}(a_1^*, \ldots, a_m^*) = \bigwedge \{ \bigvee \{\wedge B^* : B \in (\mathbb{A}, D)[u, v]\} : u D_A v\}$$

where $B^* = \{a^* : a \in B\}$. *We have the following equivalence:*

$a_{i_1}^* \wedge \ldots \wedge a_{i_k}^*$ is a prime implicant of $g_{\mathbb{A},D}$ iff $\{a_{i_1}, \ldots, a_{i_k}\} \in RED(\mathbb{A}, D)$. □

Corollary 3.3 The problem of computing minimal tolerance reducts of $(\mathbb{A}, D)$ is NP-hard. □

The presented method can be extended to the so-called relative tolerance reducts [47].

It is possible to apply Boolean reasoning for the object set reduction in tolerance information systems. This is based on the notion of absorbent [28], [50]. A subset $Y \subseteq X$ is an *absorbent* for a tolerance relation τ (τ-*absorbent*, in short), if and only if, for each $x \in X$ there exists $y \in Y$ such that $x \tau y$. The problem of minimal absorbent construction for a given tolerance information system can easily be transformed to the problem of minimal prime implicant finding for a Boolean function corresponding to this system [29], [50]. The problem of minimal absorbent construction is NP-hard [29], so efficient heuristics have been constructed to find sub-minimal absorbents for tolerance information systems.

4 FEATURE EXTRACTION

We may define features as functions (attributes) on objects derived from the existing attributes. In this respect one may realize that a given data table presents not only a small fragment of the reality as it classifies a tiny fraction of objects, but it also employs a tiny fraction of possible attributes. The purpose of feature extraction is to obtain a set of attributes with better classifying properties with respect to new objects. An important criterion for the quality of feature extraction is the reduction of dimensionality and size of the classification space. Our perception of the data structure determines the set of possible features. In this set we look for the relevant features. We would like to point out the fact that the process of synthesis of adaptive decision algorithms should allow for adaptive search for proper (from the classification point of view) representation of object structure (knowledge representation). For example, searching for a proper representation of structure in the logical framework requires finding a proper syntax and semantics of a logical language. The applications of multi-modal logics to this problem are discussed in [5]. There are many feature extracting techniques [22]. We discuss here examples of some techniques constructed with application of rough set-theoretic methods and Boolean reasoning methods.

4.1 Feature extraction by searching in a given set of formulas

Let $\mathbb{A} = (U, A \cup \{d\})$ be a (consistent) decision table and let $U = \{x_1, \ldots, x_n\}$. By $DISC(\mathbb{A})$ we denote the set $\{(x_i, x_j) \in U^2 : d(x_i) \neq d(x_j)\}$ called the *discernibility set of* $\mathbb{A}$.
A finite family $\mathcal{F}$ of functions from U into a non-empty set W is $\mathbb{A}$-*complete* if $(x_i, x_j) \in DISC(\mathbb{A})$ implies $\mathcal{F}_A(x_i, x_j) \neq \emptyset$ for any $(x_i, x_j) \in DISC(\mathbb{A})$ where $\mathcal{F}_A(x_i, x_j) = \{f \in \mathcal{F} : f(x_i) \neq f(x_j)\}$.

Let p_f be a Boolean variable corresponding to $f \in \mathcal{F}$. The *discernibility formula* $\Phi^{\mathbb{A}}(\mathcal{F})$ of $\mathbb{A}$ (relatively to $\mathcal{F}$) is defined by

$$\bigwedge\{\vee\{p_f : f \in \mathcal{F}_A(x_i, x_j)\} : (x_i, x_j) \in DISC(\mathbb{A})\}$$

By $\mathbb{A}(\mathcal{F})$ we denote the decision table $(U, \mathcal{F} \cup \{d\})$.

One of the basic problems related to new feature extraction is the following:

(OFE): OPTIMAL FEATURE EXTRACTION PROBLEM FOR $\mathcal{F}$
INPUT: A decision table $\mathbb{A}$ such that $\mathcal{F}$ is $\mathbb{A}$-complete
OUTPUT: A minimal (with respect to the cardinality) relative reduct of $\mathbb{A}(\mathcal{F})$

The output is a minimal set of features discerning between all pairs of objects from $DISC(\mathbb{A})$.

Proposition 4.1 Let $\mathbb{A}$ be a decision table such that $\mathcal{F}$ is $\mathbb{A}$-complete. Then

(i) $p_{f_1} \wedge \ldots \wedge p_{f_k}$ is a prime implicant of the discernibility formula $\Phi^{\mathbb{A}}(\mathcal{F})$ iff $\{f_1, \ldots, f_k\}$ is a relative reduct of $\mathbb{A}(\mathcal{F})$;
(ii) OFE problem is NP-hard. □

There are many examples in literature which can be treated as special cases of the above formulated Boolean reasoning process (see e.g. [37] for reduct computation, [38] for relative reduct computation, [38] for D-reducts computation and [26], [38-39], [42] for decision rules generation. This method has also been recently applied to quantization (scaling, discretization) [17], [45] of real value attributes. We illustrate our approach by the following examples:

Example 1. Feature extraction by scaling of the real value attributes [17], [45].

Let $\mathbb{A} = (U, A \cup \{d\})$ be a decision table and let V_a be included in an open interval $(v_a^{\min}, v_a^{\max}) \subseteq \mathbb{R}$ where $\mathbb{R}$ is the set of reals. Let $a(U) = \{a(x) : x \in U\} = \{v_1, \ldots, v_k\}$ and $v_1 < \ldots < v_k$. We choose $c_1, \ldots, c_k$ to satisfy $v_a^{\min} < c_1 < v_1 < \ldots < v_k < c_k < v_a^{\max}$. We define $f_{a,c_i}(x) = 1$ if $a(x) > c_i$ and 0 otherwise for any $x \in U$. The set $\mathcal{F}_A(x_i, x_j)$ for any $(x_i, x_j) \in DISC(\mathbb{A})$ consists of functions f_{a,c_k} satisfying $f_{a,c_k}(x_i) XOR f_{a,c_k}(x_j)$ iff $a(x_i) < c_k < a(x_j)$ or $a(x_j) < c_k < a(x_i)$. Then the discernibility formula $\Phi^{\mathbb{A}}(\mathcal{F})$ (where $\mathcal{F}$ is the union of all sets $\mathcal{F}_A(x_i, x_j)$ for $(x_i, x_j) \in DISC(\mathbb{A})$)) can be used to generate the solution for optimal scaling problem. In real applications the formula $\Phi^{\mathbb{A}}(\mathcal{F})$ is rather complicated and some heuristics should be applied to obtain sub-optimal solutions. This idea is realized in [45]. One can also consider more complicated case when the pairs from $DISC(\mathbb{A})$ are distinguished by hyperplanes not necessarily parallel to axes. Genetic algorithms and genetic programming techniques are then applied to search for sub-optimal solutions of OFE problem in this case. □

Example 2. Feature extraction for structural objects [5].

In [5] objects (with structures described by finite labelled graphs with distinguished input nodes) are considered as structures for handwritten digits. The disjoint union of these graphs defines a Kripke model M. A multimodal formula discerns between objects x_i and x_j if

$$M, x_i \models \alpha \text{ and } non(M, x_j \models \alpha) \text{ or } M, x_j \models \alpha \text{ and } non(M, x_i \models \alpha)$$

Let $f_\alpha(x) = 1$ if $M, x \models \alpha$ and 0 otherwise, for any $x \in U$.

We put $\mathcal{F}(x_i, x_j) = \{f_\alpha : f_\alpha(x_i) \ XOR \ f_\alpha(x_j)\}$. A procedure for synthesis of multimodal formulas α discerning between objects in pairs of $DISC(\mathbb{A})$ is presented in [5] and applied to automatic generation of features discerning between handwritten digits. □

4.2 Feature extraction by discovery of approximate dependencies

This technique consists of finding near-to-functional relationships in data tables [32], [40], [43] described by rules of the form $\alpha \Longrightarrow \beta$ where α and β are Boolean combinations of descriptors over $B \subseteq A$ and $C \subseteq A$, respectively, such that $|\alpha_{\mathbb{A}}|$ and $|\alpha_{\mathbb{A}} \cap \beta_{\mathbb{A}}|/||\alpha_{\mathbb{A}}|$ are greater than the fixed thresholds. Let $F_{\alpha,\beta}$ be the approximation function defined by $\alpha \Longrightarrow \beta$ in $\mathbb{A}$ [32], [40], [43].

The approximation functions applied for filtration of decision tables can also be used in searching for new features. Let us assume that our system is searching for the approximation functions of the form $F_{\alpha,\beta}$ for some α, β with the following property: if $F_{\alpha,\beta}(u) = v$ where u and v are some pieces of the information, then there is a strong evidence (measured e.g. by some threshold k) that the object characterized by u and v belongs to a distinguished set of decision classes. If it is possible to discover this kind of approximation function, then one can add as a new condition (a feature, a classifier) to the decision table the binary attribute a_F defined by:

$$a_F(x) = 1 \quad \text{iff} \quad F_{\alpha,\beta}(\{(a, a(x)) : a \in B\}) = \{(a, a(x)) : a \in C\}$$

where B, C are sets of attributes occurring in α, β, respectively.

One can expect to get an efficient classification mechanism by adding to the decision table several features of the above form distinguishing a particular set of decision classes with sufficiently large evidence (related to the value of the threshold k).

In [32] we present a detailed introduction to mathematical morphology [35] and a higher-level version of mathematical morphology called analytical morphology. This is aimed at filtering data tables without any apriorical geometrical structure.

4.3 Feature extraction by clustering

Clustering may be defined informally as an attempt at imposing a distance measure (function) d on objects of some collection in such a way that the collection U can be represented as a union $C_1 \cup C_2 \cup \ldots \cup C_k$ of subcollections $C_i \subseteq U$, $i = 1, 2, \ldots, k$ which form clusters, i.e. d-distances among objects within any cluster C_i are relatively small compared to distances between distinct clusters. The vague nature of this informal description reflects the ambiguity inherent to clustering: as a rule, neither the distance function nor the size of clusters, nor the degree to which they may overlap are defined clearly; they are subject to some judicious choice. This choice can be based on different criteria: minimization of weighted least squares functionals, hierarchical clustering, and graph-theoretical methods [2].

No matter which technique we apply to clustering, it is important to realize that we interpret clusters as collections of objects similar to one another, while objects from distinct clusters are perceived by us as not being similar.

Clusters, therefore, define a similarity relation on objects from the given collection; it is seldom that this similarity has more properties besides reflexivity and symmetry. Clusters form a covering of the universe with intricate overlapping relationships as there is a trade-off between the crispness of clustering and the adaptive quality of clusters-based decision algorithms. This cluster-defined similarity is therefore a tolerance relation. We restrict ourselves to the simplest case in which a clustering $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ on the universe U determines a tolerance relation $\tau_{\mathcal{C}}$ on U by the condition

$$x \tau_{\mathcal{C}} y \quad \text{iff} \quad x, y \in C_i \quad \text{for some} \quad i = 1, 2, \ldots, k.$$

It is an important property of any tolerance relation τ that it determines a metric (distance function) d_τ on U by the formula

$$\begin{aligned} d_\tau(x, y) = \min\{n : &\text{ there exists a sequence } x_0, x_1, \ldots, x_n \text{ such that} \\ &x_0 = x,\ x_n = y \text{ and } (x_i \tau x_{i+1} \text{ for } i = 0, \ldots, n-1)\} \\ &\text{in case } x \neq y \text{ and } d_\tau(x, x) = 0. \end{aligned}$$

We will call d_τ the *tolerance iterate metric.*
Clusters C_i are therefore defined metrically by the conditions:

a) $d_{\tau_{\mathcal{C}}}$-*diameter* $C_i = 1$ for any $i = 1, 2, \ldots, k$ and

b) $d_{\tau_{\mathcal{C}}}$-$dist(C_i, C_j) = \min\{\{\max dist(x, C_j) : x \in C_i\},$
$\max\{dist(x, C_i) : x \in C_j\}\} \geq 2$ for any $i \neq j$.

The following conclusion is obtained:

Any clustering $\mathcal{C}$ on a set U of objects can be represented in the form $\mathcal{C}(\tau, k, m)$ where τ is a tolerance on U, and k, m are natural numbers with $1 \leq k < m$, and

a) d_τ-*diameter* $C_i \leq k$ for any $C_i \in \mathcal{C}$;
b) d_τ-*dist* $(C_i, C_j) \geq m$ for any pair $C_i, C_j \in \mathcal{C}$.

The above observation is the cornerstone of our clustering strategy: we will search for tolerance relations τ whose iterate metric will satisfy a) and b). Actually, we will search for tolerance relations τ satisfying a stronger condition b') for some m

b') d_τ-$Dist(C_i, C_j) = \min\{d_\tau(x, y) : x \in C_i, y \in C_j\} \geq m$ for any pair $C_i, C_j \in \mathcal{C}$.

Finding such a tolerance τ would mean that we have found a similarity of object relations whose iterates up to k separate objects into certain attractors (clusters) while iterates of higher degrees separate the clusters.

There are two basic problems into which the clustering by tolerance problem splits: the problem of clustering with a given tolerance relation as well as the problem of searching for a proper tolerance relation. We present here a proposition related to the second problem.

The searching for a proper tolerance is the crucial and most difficult task related to decision algorithm synthesis based on tolerance. Let us restrict our considerations to the tolerances defined by weighted distances between information vectors u, v:

$$d(u, v) = \sum_i w_i |u_i - v_i|$$

where w_i is a weight of the i-th attribute and $w_i \geq 0$.

Some heuristics have to be applied due to the high computational complexity of searching for the proper weights. One can use the ideas of e.g. simulated annealing [1], genetic algorithms [12] or neural networks [10].

We sketch here the basic idea of the simulated annealing application to this problem. Any choice of weights $\{w_i\}$ determines the state of the system. The transition relation can be defined by choosing a local perturbation of weights. The resulting new state (the new tolerance relation) is tested. This testing stage consists of applying the clustering strategy based on the new tolerance for synthesis of decision algorithm which is then tested on unseen objects. If the test results are satisfactory, then the new state received by the local perturbation of weights is evaluated as better than the previous state and is taken as the current state of the process of simulated annealing. If the test results are not satisfactory, then the new state is selected as a current state with probability given by the Boltzmann distribution. Iterating the process, we determine a sequence of states that is a sequence of tolerance relations. One can experimentally choose the number of iterations. Once the iteration process is completed, we decrease (slightly) the control parameter (corresponding to the temperature) and repeat the iteration. This procedure is repeated until the stop criterion chosen in advance is reached. The obtained final tolerance relation is taken as the best possible tolerance. Clearly, much more work should be done to apply this idea to specific data tables. Although the scheme is universal, the choice of specific parameters of the simulated annealing process will depend

on the particular data table, and these parameters should be tuned to fit any concrete case.

4.4 Feature extraction by optimization

In this section we will treat learning processes as optimization processes. In this setting learning can be viewed as a searching for an optimal discernibility relation. The quality of a discernibility relation R is measured by cost function related to a decision table $\mathbb{A}$ from a given family of tables. There are several components of this function. The first one describes how the discernibility relation approximates the set of pairs of objects which should be discerned. Hence the values of this component reflect the size of a boundary region, which can be described as the set of all pairs of objects (x, y) discernible by the decision and not discernible by R, i.e. $(x, y) \notin R$. The second component estimates the cost of splitting decision classes by the discernibility relation R. This cost is increasing when the discernibility relation introduces too many cuts in decision classes. The last component is related to the complexity of the discernibility relation. The complexity can be measured by the time or/and space complexity of computation of the characteristic function of the discernibility relation R. This component can also include the cost of searching in the space $\mathcal{R}$ of discernibility relations. We also include in the cost of the discernibility relation the changes in the classification quality caused by changes to the original decision table.

We now present our approach in a more formal way. We assume that value sets V_a for $a \in A$ are given. All decision tables considered in this section are assumed to have attributes from A. We consider a family $\mathcal{R}$ of binary relations $R \subseteq INF(A, V) \times INF(A, V)$ called the *discernibility relations* where $V = \cup V_a$ and $INF(A, V)$ is the set of all functions from A into V. If $\mathbb{A}$ is a decision table and $R \in \mathcal{R}$ then we define the relation $\hat{R}$ by $(x, y) \in \hat{R}$ iff $(Inf_A(x), Inf_A(y)) \in R$ for any $x, y \in U$. In the sequel we write R instead of $\hat{R}$. Let $\mathbb{B} = (U_B, B \cup \{d_B\})$ be a decision table. We introduce the following notation: $DIS(\mathbb{B}) = \{(x, y) \in U_B \times U_B : d_B(x) \neq d_B(y)\}$ and

$$X_{i,B} = \{x \in U_B : d_B(x) = i\}.$$

By τ we denote a tolerance relation defined on considered decision tables; $\tau(\mathbb{A})$ is the set $\{\mathbb{B} : \mathbb{A}\tau\mathbb{B}\}$.

The *cost function* (assuming $\mathbb{A}, \tau$ and $\mathcal{R}$ are fixed) is defined for $R \in \mathcal{R}$ by

$$\begin{aligned} \mathcal{C}_{\mathbb{A},\tau}(R) &= \sup_{\mathbb{B}\in\tau(\mathbb{A})} \Big[-C_0 \log \frac{|DIS(\mathbb{B}) \cap R|}{|DIS(\mathbb{B})|} \\ &- \sum_{i=1}^{r(d_B)} C_i |X_{i,B}| \log\Big(1 - \frac{|R \cap X_{i,B}^2|}{|X_{i,B}|^2}\Big)\Big] + D\ complexity(R) \end{aligned}$$

where $\mathcal{R}$ is a given family of relations, and C_0, C_1, D are positive reals. The *complexity(R)* is a heuristic measure of time/space complexity of the relation R. We assume $\mathcal{R} = \cup\mathcal{R}_n$ where $\mathcal{R}_n$ contains relations of size n. The value *complexity(R)* can be expressed e.g. as a function of n for $R \in \mathcal{R}_n$.

The *optimal cost* is defined by $c_{opt} = \inf_{R\in\mathcal{R}} \mathcal{C}_{\mathbb{A},\tau}(R)$. The *optimal discernibility relation* R_{opt} is a relation satisfying the equality $\mathcal{C}_{\mathbb{A},\tau}(R_{opt}) = c_{opt}$. Simulated annealing techniques [1] can be applied to find solutions near to optimal. Genetic algorithms [12], [13] can also be applied to search for optimal discernibility relation. The cost function can be used to construct fitting functions for chromosomes. Applications of these methods will be presented elsewhere.

5 APPROXIMATE REASONING FOR DECISION SYNTHESIS IN DISTRIBUTED DECISION SYSTEMS

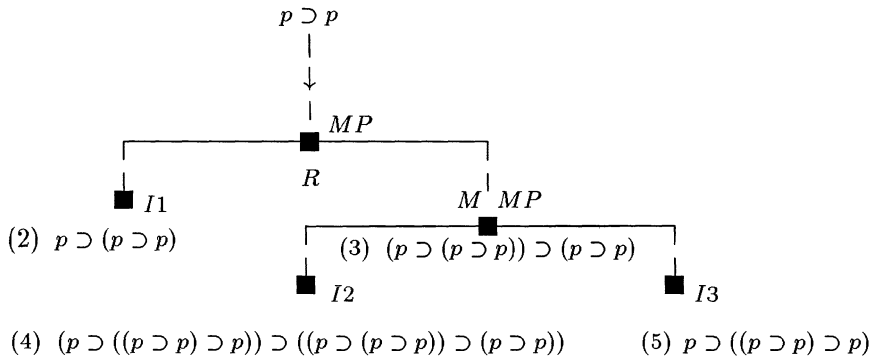
Many researchers expect that methods based on approximation logic (logic for reasoning under uncertainty) should have serious impact on the design process of decision support systems as well as on inference engines for reasoning under uncertainty embedded in these systems. Unfortunately, it is so far rather rare to find in real systems this kind of application of logics for reasoning under uncertainty. There are at least two main reasons for this situation. The first one follows from the observation that the structure of an approximate logic supporting diagnosis or belief strongly depends on a particular knowledge base. The approximate logic should be in a sense extracted from a low-level knowledge (e.g. represented by decision tables). The methods presented in the previous sections can be treated as an attempt to extract laws for such logic. Specific inference rules for approximation reasoning should also be extracted from a low-level knowledge (e.g. by decomposition of decision tables). Much more work should be done to investigate algorithmic methods for discovery of structures of approximation logics. Methods for learning Bayesian-belief networks [7] give one of the examples of possible directions for this kind of research.

The second reason is a lack of a general formalism on which approximate logic should be built. There are many approaches to this problem known in literature, e.g. Dempster-Shafer theory [52], Bayesian-based reasoning [36], belief networks [36], many-valued logics and fuzzy logics [9], non-monotonic logics [36]. Recently, we have proposed a novel approach (see [30], [31], [33]]), namely rough mereology as a foundation for approximate reasoning about complex objects. Our notion of a complex object includes, among others, proofs understood as schemes constructed in order to support within our knowledge assertions/hypotheses about reality described by our knowledge incompletely. This approach seems to be a good common framework on which approximate logics can be built. We only present here the basic ideas of that approach. The interested reader can find more information in [15].

Let us begin with a formal proof of the formula

(1) $p \supset p$

in the axiomatic propositional calculus.



Formulae in $I1, I2, I3$ are instances of axiom schemata.

We would like to consider the above derivation tree as a scheme for synthesis of solution to the problem whether the formula (1) is a theorem of the respective formal system. We wish to interpret the above scheme as a hierarchical scheme into which some intelligent agents $R, M, I1, I2, I3$ are organized which act according to the following procedural steps:

Step 1. The agent R receives the formula (1) and decomposes it into some ingredients possibly in various ways.

Step 2. R is involved in negotiations with other agents which select some ingredients of (1); the result of negotiations in the above example is that if M and $I1$ are able to validate (2) and (3), then (1) can use its rule MP to validate (1).

Step 3. M repeats steps 1, 2 with $I2, I3$ validating (4), (5), respectively.

Step 4. Synthesis of the proof follows by each agent sending to its parent the validated formula along with the validity check.

Step 5. R issues the final validity check of the proven formula.

The above scheme is the one that we adopt on the intuitive basis as the general scheme for organizing reasoning under uncertainty. We point also to the following observations which illuminate the differences between the above case and the case when we reason under uncertainty.

A. Reasoning under uncertainty involves agents whose knowledge and a fortiori logic is local and subjective.

B. The knowledge of any agent includes its local mereological knowledge which permits decomposing objects into simpler ones; different agents can as a rule have distinct proprietary knowledge. The proprietary knowledge of an agent may not be fully understood by other agents which introduces uncertainty into their cooperation. In particular, the external specification (e.g. formula to be proven) may not be fully understood by agents.

C. The decomposition process stops at the level of elementary objects (e.g. instances of axiom schemata, inventory objects etc.). The leaf (inventory) agents select objects which in their knowledge satisfy received specifications to an acceptable degree.

D. Any agent receiving an object is able within its knowledge to approximately classify this object by means of its similarity to certain model objects (e.g. logical values, local standards etc.) construed by the agent.

E. The dual (to decomposition) process of synthesis of complex objects from simpler ones sent by children of the agent, consists of the assembling process along with the classification of the synthesized object with respect to the agent model objects. This classification is based on classifications made by children and involves the agent logic.

F. The inference rules for approximate reasoning of the agent a are of the form:

if (x_1, t_1) and (x_2, t_2) and...and (x_m, t_m)
then $(\text{synt}(x_1, \ldots, x_m), \rho(t_1, t_2, \ldots, t_m))$

where x_i is the object submitted by child i, t_i is the similarity (tolerance) degree vector of child i with respect to its model objects, synt$(x_1, \ldots, x_m)$ is the object synthesized from $x_1, \ldots, x_m$ and ρ is an inference rule (which in [33] is interpreted as a mereological connective). The value $\rho(t_1, \ldots, t_m)$ is the vector of similarity degrees of synt$(x_1, \ldots, x_m)$ with respect to model objects of a.

From the above discussion we extract a general scheme $\mathcal{S}$ for synthesis of a solution under uncertainty. By an *uncertainty frame* $\mathcal{S}$ we will understand a tuple

$$\begin{aligned}\mathcal{S} = (AG, cag,\ & \{t(ag) : ag \in AG \cup \{cag\}\},\\ & \{\text{decomp}(ag) : ag \in AG \cup \{cag\}\},\\ & \{\rho(ag) : ag \in AG \cup \{cag\}\})\end{aligned}$$

where AG is a set of agent names, $cag \notin AG$ is an external agent called the customer, $t(ag)$ is in a family of tolerance relations of the agent ag measuring similarity of a given object to model objects of ag, decomp(ag) belongs to a family of mereological relations used to decompose objects at ag into simpler objects, while $\rho(ag)$ is in a family of inference rules of ag taking as arguments a number of tolerances and transforming them into a tolerance at ag.

Problem specifications are issued by the agent cag and are formulated in some languages of AG. When a specification is issued, the designing process of organization of agents from AG into a hierarchical structure for solution synthesis (searching for a proof, assembling an object etc.) is initialized. The hierarchy is formed on the basis of structural complexity of objects described by agent knowledge. The task of the formed hierarchy is to synthesize a complex object along with the evaluation that it satisfies the specification. The agent cag compares this evaluation with its own evaluation of the issued object (solution) with respect to its knowledge. The process of learning the correct synthesis of solutions is completed when the two evaluations are consistent.

The above synthesis scheme is derived from the methodological standpoint that reasoning under uncertainty should be based on different assumptions than reasoning in the classical logic viz.

a. knowledge about the world is distributed among agents possessing incomplete fragments of it;
b. agents derive their inference rules as well as model objects (truth values) from their knowledge and as a result the inference rules are local, intricate

and strongly dependent on the local knowledge. It is hardly expected in many real-life cases that a human being is able to extract these rules from empirical knowledge without help from an automated computer-based system;

c. The complex object that is issued is evaluated by composing local inference rules. The composition of local inference rules means propagating uncertainty through the scheme to the final similarity degrees. The propagation mechanism (proof) should be stable in the sense that when it starts with objects sufficiently similar to certain model objects, it should terminate with the final object satisfactorily close to certain model final objects.

Our approach is analytic in the sense that the underlying logical apparatus is extracted from experimental knowledge expressed in local agent knowledge. Also, we adhere to the point of view that rules (logic) for composing uncertainty are intensional i.e. context-depending. This is contrary to the prevailing approaches based on some a priori assumptions about logical schemes of reasoning under uncertainty, where the a priori set rules are based on some truth - valued logics.

We now present a concise elaboration of the ideas exposed in the above introduction. The exposition is based on our results [15], [29], [30-31], [33-34].

5.1 Rough mereology

The basic notion of rough mereology is that of a rough inclusion. A rough inclusion offers a most general formalism for the treatment of partial containment. Rough mereology can be regarded as a far - reaching generalization of mereology of Leśniewski [18], [30-31], [33-34]: it replaces the relation of being a (proper) part with a hierarchy of relations of being a part in a degree.

For simplicity reasons, we restrict ourselves to the case when rough inclusions take their values in the interval $[0, 1]$; in the general (case) these values may be taken in a complete lattice, e.g. a finite boolean algebra.

A real function $\mu(X, Y)$ on a universe of objects U with values in the interval $[0, 1]$ is a *rough inclusion* when it satisfies the following conditions:

(A) $\mu(X, X) = 1$ for any X;

(B) $\mu(X,Y) = 1$ implies that $\mu(Z,Y) \geq \mu(Z,X)$ for any triple X, Y, Z; if in addition $\mu(Y,X) = 1$ then $\mu(Y,Z) \geq \mu(X,Z)$;

(C) there is N such that $\mu(N,X) = 1$ for any X.

An object N satisfying (C) is a *μ-null object*: such objects are in principle excluded in mereology of Leśniewski. We let $X =_\mu Y$ iff $\mu(X,Y) = 1 = \mu(Y,X)$ and $X \neq_\mu Y$ iff $non(X =_\mu Y)$. The relation $X =_\mu Y$ is an equivalence relation and we can factor the universe U throughout this relation. For notational simplicity, we will still denote classes of $X =_\mu Y$ by symbols of objects. We now can introduce further conditions for rough inclusion:

(D) **if** objects X, Y have the property that for any Z:
if $Z \neq_\mu N$ and $\mu(Z,X) = 1$ **then** there is $T \neq_\mu N$ with $\mu(T,Z) = 1 = \mu(T,Y)$ **then** $\mu(X,Y) = 1$.

(D) is an inference rule: it can be applied to infer the relation of being a part from the relation of being a subpart.

(E) For any collection $\mathcal{F}$ of objects there is an object X with the properties:

(i) **if** $Z \neq_\mu N$ and $\mu(Z,X) = 1$ **then** there are $T \neq_\mu N$, $W \in \mathcal{F}$ such that
$\mu(T,Z) = \mu(T,W) = \mu(W,X) = 1$;

(ii) **if** $W \in \mathcal{F}$ **then** $\mu(W,X) = 1$;

(iii) **if** Y satisfies the above two conditions in place of X **then** $\mu(X,Y) = 1$.

(E) can be applied to show the existence and uniqueness of classes of objects.

An archetypal rough inclusion [31] is the rough membership function μ_A defined for an information system $A = (U, A)$ by the formula

$$\mu_A(x, X) = |X \cap [x]_A| / |[x]_A|$$

which can be in the obvious way extended to pairs of subsets of the universe U. A rough inclusion μ on a universe U induces in U a model of mereology of Leśniewski.

We define the relation *part* from a rough inclusion μ by the following

$$X part Y \quad \text{iff} \quad \mu(X,Y) = 1 \quad \text{and} \quad \mu(Y,X) < 1.$$

The relation *part* satisfies the axiom $A1$ and $A2$ of mereology of Leśniewski [31] viz.

(i) it is not true that $X part\ X$ for any object X;
(ii) if $X part\ Y$ and $Y part\ Z$ then $X part\ Z$ for any triple X, Y, Z of objects.

We can regard the relation *part* as decomposition scheme of objects extracted from the rough inclusion μ. The relation *part* expresses the property of being a (proper) part. In many cases, however, it is convenient to argue about objects in terms of the property of being an improper part i.e. either a proper part of an object or the whole object.

In mereology of Leśniewski [18] the property of being an improper part is introduced by means of relation *ingr* of being an ingredient, defined by the following condition:

$$X\ ingr\ Y \quad \text{iff either} \quad X\ part\ Y \quad \text{or} \quad X = Y.$$

Then the relation *ingr* defined from *part* satisfies the condition $X ingr\ Y$ iff $\mu(X, Y) = 1$.

Mereology of Leśniewski owes its specific metamathematical features to the definitions of the notions of a set of objects and of a class of objects [18]. In rough mereology, we extend these notions by introducing, for any collection $\mathcal{F}$ of objects of the universe U, the notions of a set of objects in $\mathcal{F}$, set $\mathcal{F}$ in short, and of a class of objects in $\mathcal{F}$, class $\mathcal{F}$ in short.

The notions *set* $\mathcal{F}$ and *class*$\mathcal{F}$ are defined from *ingr* by conditions (E)(i) and (E)(i)-(iii) with $\mathcal{F}$, respectively. It turns out [18] that the above relations induced by μ satisfy all axioms of mereology of Leśniewski on non-null objects of the universe: any rough inclusion μ introduces a model of mereology on the collection of non-μ-null objects of the universe U. It is well - known [18] that in mereology the notions of a subset, an element, and an ingredient are all equivalent. Therefore rough mereological containment $\mu(X, Y)$ can be interpreted as the membership degree of X in Y and therefore rough mereological approach encompasses fuzzy set approach.

We will comment briefly here on the way in which rough mereology encompasses fuzzy logic. In the light of the afore - mentioned equivalence between the notion of an element and the notions of a subset, which the given rough inclusion μ introduces on the objects in the universe U, we can interpret the value $\mu(X, Y)$ of the rough inclusion μ as the degree in which the object X is an element

of the object Y. This interpretation will be stressed by usage of the symbol $\mu_Y(X)$ instead $\mu(X,Y)$.

In fuzzy set theory [9], the values of fuzzy membership functions are propagated by means of logical connectives derived from many - valued logic, e.g. t-norms and t-conorms.
The following proposition (cf. [34]), whose proofs will be given elsewhere, demonstrates that a rough inclusions are preserved with respect to the fuzzy set-theoretic inclusion operators and, the decomposition schemes are invariant under change of inclusion operators.

Proposition 5.1.
Assume that μ is a rough inclusion on a universe U of objects and $\top$ is a continuous norm with $\vec{\top}$ the residual implication induced by $\top$ [9]. Then

(i) the function $\tau_\top(X,Y) = \inf_Z\{\vec{\top}(\mu_X(Z),\mu_Y(Z))\}$ is a rough inclusion on the universe U;

(ii) $\tau_\top(X,Y) = 1$ iff $\mu(X,Y) = 1$. □

We can therefore regard t-norm-induced connectives as modifiers of numerical degrees of partial containment, which have no impact on the proper containment i.e. on the decomposition scheme. The formula (i) expresses the intensional character of a rough inclusion.

Propagating rough inclusions can be effected by means of rough mereological connectives.

An n-rough mereological connective F is a relation $F \subseteq [0,1]^n \times [0,1]$ such that

$$[1,1,\ldots,1] \in F^{-1}(1).$$

Examples of mereological connectives are some connectives of many-valued logic e.g. $F(x,y) = min(x,y)$ (Zadeh), $F(x,y) = xy$ (Menger). When a connective F is chosen, we define the (F,μ)-*closeness relation* $E_{F,\mu}(X,Y,r)$ by the formula

$$E_{F,\mu}(X,Y,r) \quad \text{iff there exist} \quad s,t$$

such that

$$F(\mu(X,Y),\mu(Y,X),s) \quad \text{and} \quad F(\mu(Y,X),\mu(X,Y),t) \quad \text{and} \quad s,t \geq r.$$

For a chosen threshold $k \in [0,1]$, we define a tolerance $\tau_{F,\mu,k}$ via

$$X \tau_{F,\mu,k} Y \quad \text{iff} \quad E_{F,\mu}(X,Y,r) \quad \text{and} \quad r \geq k.$$

The connective F in distributed systems of 5.2, below may vary from agent to agent; for simplicity of notation, we assume that all agents apply a fixed connective F.

To the tolerance relation of the form $\tau_{F,\mu,k}$, all procedures described in Section 3.5. can be applied.

The analytic approach requires that values of rough inclusions be generated from data tables. We would like to comment on the way in which one can generate rough inclusion from the knowledge encoded in data tables.

Consider an information system $\mathbb{A} = (U, A)$. We call a function $\mu_0 : U \times U \to [0,1]$ a *pre-rough inclusion* when μ_0 satisfies the following conditions:

(i) $\mu_0(x,x) = 1$ for any object x in U;
(ii) if $\mu_0(x,y) = 1$ then $\mu_0(z,y) \geq \mu_0(z,x)$ for any triple x, y, z of objects in U;
(iii) $\mu_0(x,y) = \mu_0(y,x)$.

Pre-rough inclusion can be generated from the information system U; for instance, for a given partition $P = \{A_1, \ldots, A_k\}$ of the set A of attributes into non-empty sets $A_1, \ldots, A_k$, and a given set $W = \{w_1, \ldots, w_k\}$ of weights, $w_i \in [0,1]$ for $i = 1, 2, \ldots, k$ and $\sum_{i=1}^{k} w_i = 1$ we can let

$$\mu_{o,P,W}(x,y) = \sum_{i=1}^{k} w_i \cdot \frac{|IND(x,y,i)|}{|A_i|}$$

where $IND(x,y,i) = \{a \in A_i : a(x) = a(y)\}$.

Clearly, $\mu_{o,P,W}$ is a pre-rough inclusion for any choice of the parameters P, W.

Once a pre-rough inclusion μ_0 is selected, it can be extended to a rough inclusion on the set 2^U. We state the following proposition to this end, whose proof will be given elsewhere.

Proposition 5.2.
For a given: t-norm $\top$, t-conorm $\bot$ and pre-rough inclusion μ_0 on the universe U of an information system $\mathbb{A} = (U, A)$, the formula

$$\mu(X,Y) = \top\{\bot\{\mu_o(x,y) : y \in Y\} : x \in X\}$$

where $X, Y \subseteq U$, defines a rough inclusion μ on the set 2^U of sets of objects in U. □

Remark. The correctness of the right hand side of the above equality follows from the associativity and commutativity of $\top, \perp$. We assume the conventions $\top\phi = 1$, $\perp\phi = 0$, $\top r = \perp r = r$ for $r \in [0, 1]$. □

We can therefore, in the light of Proposition 5.2, restrict ourselves to pre-rough inclusions.

5.2 Rough mereological logic and distributed systems of decisions

We begin with a set Ag of agent (team) names, an inventory I of objects and a language $Link \subseteq Ag^+$ where Ag^+ is the set of all finite non-empty strings over Ag and a variable b_{ag} for any agent ag to store complex objects at the agent ag. If

$$\mathbf{ag} = ag_1 ag_2 \ldots ag_k ag \in Link$$

then it will mean that the **ag**-*target agent* ag can receive messages from the team of **ag**-*sources* $ag_1 ag_2, \ldots ag_k$. For **ag** we let $set(\mathbf{ag}) = \{ag_1, ag_2, \ldots, ag_k, ag\}$. For $L \subseteq Link$, we let $Ag(L) = \cup\{set(\mathbf{ag}) : \mathbf{ag} \in L\}$ and we denote by $\leq$ a relation on $Ag(L)$ defined by: $ag \leq ag'$ if and only if there exists $\mathbf{ag} \in L$ such that $ag, ag' \in set(\mathbf{ag})$ and ag is the **ag**-target. A set $L \subseteq Link$ is a *construction support* in case $(Ag(L), \leq)$ is a tree. A subset $Ag(I)$ of Ag is the name set of *inventory agents* (these agents have access to inventory of atomic parts).

We define an *elementary construction* c: if $\mathbf{ag} = ag_1 ag_2 \ldots ag_k ag \in L$, then an expression $c = (lab(ag_1), lab(ag_2), \ldots, lab(ag_k), lab(ag))$ will be called an *elementary construction associated with* **ag** with the *leaf set* $Leaf(c) = \{ag_1, ag_2, \ldots, ag_k\}$, the root $Root(c) = ag$ and the *agent set* $Ag(c) = \{ag_1, ag_2, ..., ag_k, ag\}$; we will write *lab(ag ,c)* instead of *lab(ag)* to stress the fact that the agent ag has the label *lab(ag)* in c. The *label*, *lab(ag)*, of an agent ag is the set:

$$\{U(ag), L(ag), \mu(ag), decomp\_rule(ag), uncertainty\_rule(ag), F(ag)\}$$

where $U(ag)$ is the universe of ag, $L(ag)$ is a set of unary predicates at ag, $\mu(ag) \subseteq U(ag) \times U(ag) \times [0, 1]$ is a pre-rough inclusion at ag, $F(ag)$ is a set of mereological connectives at ag and $decomp\_rule(ag)$ is the set of relations of the form $decomp\_rule_j$ of type $(\Phi_1, \Phi_2, \ldots, \Phi_k, \Phi)$ where $\Phi_i \in L(ag_i)$ and $\Phi \in L(ag)$; the meaning of this relation is that it is satisfied by objects $c_1, .., c_k$

submitted by $ag_1, \ldots, ag_k$ iff c_i satisfies Φ_i for $i = 1, \ldots, k$ and the object $c(c_1, \ldots, c_k)$ constructed by ag from $c_1, ..., c_k$ satisfies Φ. $Uncertainty\_rule(ag)$ is a set of relations of type $(f, \mu(ag_1), \ldots, \mu(ag_k), \mu(ag))$ such that for some $x_i, y_i \in U(ag_i)$ where $i = 1, \ldots, k$ if $E_{F,\mu(ag_i)}(x_i, y_i, r_i)$ for $i = 1, 2, \ldots, k$ then we have

$$E_{F,\mu(ag)}(c(x_1, \ldots, x_k), c(y_1, \ldots, y_k), r), \text{ where}$$
$$f(r_1, r_2, \ldots, r_k, r) \text{ holds and } f \in F(ag).$$

For two constructions c, c' such that

$$Ag(c) \cap (Ag(c') = \{ag\}$$

where $ag = Root(c) \in Leaf(c')$ and $lab(ag, c) = lab(ag, c')$ we define a new construction $c \bullet_{ag} c'$ called the *ag-composition* of c and c' with $Root(c \bullet_{ag} c') = Root(c')$,

$$Leaf(c \bullet_{ag} c') = Leaf(c) \cup Leaf(c') - \{ag\},$$
$$Ag(c \bullet_{ag} c') = Ag(c) \cup Ag(c').$$

We call a *construction* any expression obtained from a set of elementary constructions by applying the composition operation a finite number of times. If a construction $c = c_1 \bullet_{ag1} c_2 \bullet_{ag2} c_3 \bullet \ldots \bullet_{agk} c_{k+1}$ is composed in the prescribed order out of elementary constructions $c_1, \ldots, c_{k+1}$ then any c_i will be called an *elementary construction* in c. The construction c is a *communication scheme* of agents which is the result of the first stage of the design process. In this stage *communication routes* among agents are established. We now define a *design scheme sch(c) over c* (*scheme* for short) by choosing for any agent $ag \in Ag(c)$ a rule $decomp\_rule_j\,(ag)$ of type $(\Phi_1, \Phi_2, \ldots, \Phi_k, \Phi(ag))$ in such way that if ag is a leaf agent in an elementary construction c_i of c with the root ag^* and

$$decomp\_rule_j(ag^*)$$

of type $(\Psi_1, \Psi_2, \ldots, \Psi_k, \Psi(ag^*))$ then $\Phi(ag)$ is one of $\Psi_1, \Psi_2, \ldots, \Psi_k$. An elementary scheme will be a scheme restricted to an elementary construction. A given design scheme $sch(c)$ assigns to any of its agents ag a unique decomposition formula

$$\Phi(ag, sch(c))$$

and thus if we begin with inventory objects c_i which satisfy leaf decomposition formulae then the complex object c assembled at the root agent satisfies its decomposition formula. The process of forming a design scheme requires negotiations among agents along already established communication routes; we propose to apply boolean reasoning [8] for the negotiation process.

We work with a given construction c. We consider a family

$$Sch(c) = \{sch_1(c), \ldots, sch_m(c)\}$$

of design schemes over c called a *synthesis pre - scheme*. We describe in this section the communication and negotiation process making $Sch(c)$ into a *synthesis scheme under uncertainty*. For any agent $ag \in Ag(c)$, the *extended label Lab(ag)* will be defined as the union $lab(ag) \cup \{st(ag)_i : i = 1, 2, \ldots, m\}$ where $lab(ag)$ is the label of ag in c and $st(ag)_i$ is the *i-th standard object at ag* which means that $st(ag)_i \in U(ag_i)$ and $st(ag)_i$ satisfies the formula $\Phi(ag, sch(c)_i)$. Informally, the formulae $\Phi(ag, sch(c)_i)$ describe the corresponding standards and they are inferred e.g. as conditional formulae from a *priorical* decision systems of agents. A fortiori, any design scheme $sch(c)_i$ of $Sch(c)$ can be regarded as *elementary synthesis scheme* leading from the set $\{st(ag)_i : ag \in Leaf(c)\}$ of i-th standard objects at leaf agents of c to the standard object $st(ag^*)_i$ where $ag^* = Root(c)$. For any agent $ag \in Ag(c)$, and any object $x \in U(ag)$, the agent ag can calculate by means of an F-closeness relation $E_{F,\mu(ag)}$ associated with the rough inclusion $\mu(ag)$ the vector

$$dist(ag)(x) = [r_i : i = 1, 2, \ldots, m], \text{ where}$$
$$E_{F,\mu(ag)}(x, st(ag_i), r_i) \text{ holds for } i = 1, 2, \ldots, m,$$

from x to standards at ag. For any elementary construction

$$c_0 = (lab(ag_1), lab(ag_2), \ldots, lab(ag_k), lab(ag)) \text{ in } c$$

and a standard $st(ag)_i$ at ag, we can check whether there exists a relation f of type

$$(f, \mu(ag_1), \ldots, \mu(ag_k), \mu(ag))$$

in $uncertainty\_rule(ag)$ with the property that

if objects $x_1, .., x_k$ are such that each x_j satisfies

$$E_{F,\mu(ag_j)}(x_j, st(ag_j), r_j) \text{ and } r_j \geq \varepsilon(ag_j) \quad \text{for} \quad j = 1, 2, \ldots, k$$

and the object $x = c(x_1, x_2, \ldots, x_k)$ satisfies

$$E_{F,\mu(ag)}(x, st(ag), r) \text{ and } r \geq \varepsilon(ag)$$

then

$$f(\varepsilon(ag_1), \varepsilon(ag_2), \ldots, \varepsilon(ag_k)), \varepsilon(ag)) \text{ holds.}$$

Let us emphasize that the relations satisfying this property are extracted from experiments with samples of objects in the manner discussed in Section 5.1. If

such $f = f(st(ag)_i)$ exists for any pair c_0, $st(ag)_i$ in c then we say that the construction c has a *proper uncertainty propagation.*

We now give an informal description of the process of synthesis. Our considerations will be based on the notion of approximate satisfiability of a predicate by an object viz. for a predicate $\Phi \in L(ag)$, a real number ε and an object x at ag, we will say that the object x *satisfies the predicate* Φ *in degree* ε in the case when there exists a standard $st(ag)_i$ such that $st(ag)_i$ satisfies Φ and

$$E_{F,\mu(ag)}(x, st(ag)_i, r) \text{ and } r \geq \varepsilon \text{ hold for some } r \in [0,1].$$

The root agent ag of c, on receiving a specification Φ from the customer, can be able to select a standard $st(ag)_i$ which satisfies Φ (by this choice the root agent ensures that the design scheme $sch(c)_i$ will provide the support for construction of a complex object satisfying the specification Φ). Next, he can be able to set a value ε such that any object x with the property that $E_{F,\mu(ag)}(x, st(ag)_i), r)$ and $r \geq \varepsilon$ will according to him satisfy the specification Φ from the customer. The main reason for setting ε is that the leaf agents may be not able to deliver standards as required to construct $st(ag)_i$ but they can deliver objects as close to these standards as to construct from them the object which satisfies the specification Φ in degree satisfactory for the customer. The approximate specification (Φ, ε) at the root agent of c is then to be decomposed into a set

$$\{(\Phi(ag), \varepsilon(ag)) : ag \neq Root(c)\}$$

of approximate specifications for non-root agents of c in such a way that the following conditions are satisfied for any $ag \in Ag(c)$ and for the elementary construction c_0 in c with the $Leaf(c_0) = \{ag_1, ag_2, \ldots, ag_k\}$ and $Root(c_0) = ag$:

(i) there exists $f(st(ag)_i) \in F(ag)$ such that $f(st(ag)_i)(\varepsilon(ag_1)), \varepsilon(ag_2), \ldots, \varepsilon(ag_k), r)$ and $r \geq \varepsilon(ag)$;

(ii) if objects $x_1, \ldots, x_k$ are such that x_i satisfies $(\Phi(ag_i), \varepsilon(ag_i))$ for $i = 1, 2, \ldots, k$, then the object $c(x_1, x_2, \ldots, x_k)$ satisfies $(\Phi(ag), \varepsilon(ag))$.

This decomposition is effected in the top-down negotiation process starting at $Root(c)$, proceeding throughout subsequent elementary constructions and ending at leaf agents. The decomposition of $\varepsilon(ag)$ into values $\varepsilon(ag)_i$ is achieved by means of the relation $f(st(ag)_i)$ which permits for a given $\varepsilon(ag)$ to find acceptable values of $\varepsilon(ag)_i$. Let us observe that in this negotiation process the following set

$$\{(\Phi(ag), \varepsilon(ag), f(st(ag)_i)) : ag \in Ag(c)\}$$

called the (c, Φ, ε) - *uncertainty propagation scheme* is established as the successful result. The negotiation process can be carried in either parallel or sequential way leading to a non - empty family $Psch(c)$ called a *c-uncertainty propagation scheme* of (c, Φ, ε)-uncertainty propagation schemes for various specifications Φ delivered by the customer and various values ε. We will call a *synthesis scheme under uncertainty* a *synthesis pre - scheme Sch(c)* endowed with a *c-uncertainty propagation scheme* $Psch(c)$. We would like to emphasize the fact that a synthesis scheme under uncertainty is the result of the negotiation and learning process among agents on the basis of their knowledge. In any synthesis scheme under uncertainty one can notice top-down and bottom - up communication. The communication process is an instance of a much more general idea of approximate reasoning consisting of decomposing a global specification along with a given global uncertainty bound into local specifications with given local uncertainty bounds (top-down communication) and then synthesizing a complex object satisfying the global specification in degree exceeding the global uncertainty bound by assuring that complex objects assembled at local nodes satisfy local specifications in degrees exceeding local uncertainty bounds.

6 CONCLUSIONS

The rough set methods combined with Boolean reasoning techniques have been used to develop efficient tools for extracting decision rules from low-level knowledge of agents.

Adaptive systems of cooperating agents based on rough mereological approach have been proposed as a general framework for reasoning under uncertainty. It allows to express higher-level reasoning under uncertainty related to, for instance, non-monotonic reasoning, and reasoning about knowledge in distributed systems of computing agents. The agents extract from their data tables all constructs which they need: rough inclusions, predicates, decomposition rules, uncertainty rules, deep structure formulas. Automated design manufacturing and negotiations are carried out along the negotiated synthesis schemes.

We expect to obtain more practical applications of Boolean reasoning methods and rough set methods by merging them with genetic programming and neural networks. This seems especially interesting for automatic relevant features synthesis.

One of the main research areas which could help build a bridge between existing logics for reasoning under uncertainty and practical applications is related to algorithmic methods for extracting the structures of these logics from low-level knowledge bases. Decomposition techniques of decision tables can be used as the main tool in searching for these structures.

This work has been supported by a grant from the State Committee for Scientific Research (Komitet Badań Naukowych).

REFERENCES

[1] Aarts E., Korst J., "Simulated Annealing and Boltzmann Machines", Wiley, New York 1989.

[2] Anderberg M.R., "Cluster Analysis for Applications", Academic Press, New York 1973.

[3] Bazan J., Skowron A., Synak P., "Discovery of Decision Rules from Experimental Data", Soft Computing, T.Y.Lin, A.M. Wildberger (eds.), Simulation Councils, San Diego 1995, pp. 276–279.

[4] Bazan J., Skowron A., Synak P., "Dynamic Reducts as a Tool for Extracting Laws from Decision Tables", Proc. of the Symp. on Methodologies for Intelligent Systems, Charlotte, NC, October 16-19, 1994, Lecture Notes in Arificial Intelligence 869, Springer-Verlag, Berlin 1994, pp. 346-355.

[5] Bazan J., Nguyen S.H., Nguyen T.T., Skowron A., Stepaniuk J., "Applications of Modal Logics and Rough Sets for Classifying Objects", In: Second World Conference on Fundamentals of Artificial Intelligence, De Glas M., Pawlak Z. (eds.), 3-7 July 1995, Angkor, Paris 1995, pp. 15–26.

[6] Bazan J., Skowron A., "Dynamic Reducts and Stable Coverings of the Objects Set", in preparation.

[7] Bouckaert R.R., "Properties of Bayesian Belief Networks Learning Algorithm", In: Proc. of the 10-th Conf. on Uncertainty in AI, University of Washington, Seattle 1994, de Mantarnas R.L., Poole D. (eds.) Morgan Kaufmann, San Franciso 1994, pp. 102-109.

[8] Brown E.M., "Boolean Reasoning", Kluwer, Dordrecht 1990.

[9] Dubois D., Prade H., Yager R.R., "Readings in Fuzzy Sets and Intelligent Systems", Morgan Kaufmann, San Mateo 1993.

[10] Freeman J.D., Skapura D.M., "Neural Networks: Algorithms, Applications and Programming Techniques", Addison Wesley, Reading, MA 1992.

[11] Garey M.S., Johnson D.S., "Computers and Intractability", W.M. Freeman, New York 1979.

[12] Goldberg D.E., "Genetic Algorithms in Search Optimization and Machine Learning", Addison-Wesley, Reading, MA 1989.

[13] Holland J.H., "Adaptation in Natural and Artificial Systems", The MIT Press, Cambridge, MA 1993.

[14] Market Data, manuscript from Hughes Research Laboratories.

[15] Komorowski J., Polkowski L., Skowron A., "Towards a Rough Mereology - Based Logic for Approximate Solution Synthesis. Part 1", Studia Logica, to appear.

[16] Low B.T., : Neural-Logic Belief Networks - a Tool for Knowledge Representation and Reasoning", Proc. of the 5-th IEEE International Conference on Tools with Artificial Intelligence, Boston 1993, pp. 34-37.

[17] Lenarcik A., Piasta Z., "Deterministic Rough Classifiers", ICS Research Report 46/94, Warsaw University of Technology 1994.

[18] Leśniewski S., "Foundations of the General Theory of Sets" (in Polish), Moscow, 1916; also in: Surma, Srzednicki, Barnett, Rickey (eds.), "Stanisław Leśniewski Collected Works", Kluwer. Dordrecht 1992, pp. 128–173.

[19] Michie D., Spiegelhalter D.J., Taylor C.C., "Machine Learning: Neural and Statistical Classification", Ellis Horwood, New York 1994.

[20] Michalski R., Tecuci G., "Machine Learning. A Multistrategy Approach vol.IV", Morgan Kaufmann, San Mateo 1994.

[21] Mollestad T., Skowron A., "Learning Propositional Default Rules Using Rough Set Approach", In: Proc. of the Fifth Scandinavian Conference on Artifical Intelligence SCAI - 95, Aamodt A., Komorowski J. (eds.), IOS Press, Amsterdam 1995, pp.208–219.

[22] Nadler M., Smith E.P., "Pattern Recognition Engineering", Wiley, New York 1993.

[23] Pao Y.H., "Adaptive Pattern Recognition and Neural Networks", Addison Wesley, Reading, MA 1989.

[24] Payne J.W., Bettman, Johnson E.J., "The Adaptive Decision Maker", Cambridge University Press, Cambridge 1993.

[25] Pawlak Z., "Rough Sets: Theoretical Aspects of Reasoning About Data", Kluwer, Dordrecht 1991.

[26] Pawlak Z., Skowron A., "A Rough Set Approach for Decision Rules Generation", ICS Research Report 23/93, Warsaw University of Technology 1993, Proc. of the IJCAI'93 Workshop: The Management of Uncertainty in AI, France 1993.

[27] Pearl J., "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Beliefs", Morgan Kaufmann 1988.

[28] Pogonowski, J., "Tolerance Spaces with Applications to Linguistics", UAM Press, Poznań 1981.

[29] Polkowski L., Skowron A., "Decision Support Systems: A Rough Set Approach", pp. 1-312 (manuscript).

[30] Polkowski L., Skowron A., "Introducing Rough Mereological Controllers: Rough Quality Control", Soft Computing, T.Y.Lin, A.M.Wildberger (eds.), Simulation Councils, San Diego 1995, pp. 240–243.

[31] Polkowski L., Skowron A., "Rough Mereology", Proc. of Lecture Notes in Artificial Intelligence 869, Springer-Verlag, Berlin 1994, pp. 85-94.

[32] Polkowski L., Skowron A. : Analytical Morphology: Mathematical Morphology of Rough Sets", ICS Research Report 22/94, Warsaw University of Techology 1994, also in: Fund. Informaticae, to appear.

[33] L.Polkowski, Skowron A., "Adaptive Decision-Making by Systems of Cooperating Intelligent Agents Organized on Rough Mereological Principles", ICS Research Report 71/94, Warsaw University of Techology 1994, also in: Intelligent Automation and Soft Computing, to appear.

[34] Polkowski L., Skowron A., "Rough Mereology: Logic of Rough Inclusion", ICS Research Report 16/94, Warsaw University of Technology 1994.

[35] Serra J., "Image Analysis and Mathematical Morphology", Academic Press, New York 1982.

[36] Shafer G., Pearl J., "Readings in Uncertainty Reasoning", Morgan Kaufmann, San Mateo 1990.

[37] Skowron, A. and Rauszer C., "The Discernibility Matrices and Functions in Information Systems", In: R. Słowiński (ed.): Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory. Kluwer, Dordrecht 1992, pp. 331-362.

[38] Skowron A., "A Synthesis of Decision Rules: Applications of Discernibility Matrices", Proc. of the Conf. Intelligent Information Systems, Augustów, June 7-11, 1993, pp. 30-46.

[39] Skowron A., "Boolean Reasoning for Decision Rules Generation", Proc. of the 7-th International Symposium ISMIS'93, Trondheim, Norway 1993, In: J. Komorowski and Z. Ras (eds.): Lecture Notes in Artificial Intelligence, Vol.689. Springer-Verlag 1993, pp. 295-305.

[40] Skowron A., "Extracting Laws from Decision Tables", Computational Intelligence, 11(2) 1995, pp. 371-388.

[41] Skowron A., Stepaniuk J., "Decision Rules Based on Discernibility Matrices and Decision Matrices", Conference Proceedings (RSSC'94) The Third International Workshop on Rough Sets and Soft Computing, San Jose State University, CA, November 10-12, 1994, pp. 602-609.

[42] Skowron A., Polkowski L., "Adaptive Decision Algorithms", Proc. of the Workshop on Intelligent Systems, Wigry, Poland, 6-10 June, 1994, Institute of Foundations of Computer Science PAS, Warsaw 1995, pp. 103–120.

[43] Skowron A., "Data Filtration: A Rough Set Approach", In: Rough Sets, Fuzzy Sets and Knowledge Discovery (ed.) W.Ziarko, Workshops in Computing, Springer-Verlag & British Computer Society 1994, pp. 108-118.

[44] Skowron A., Grzymała-Busse J., "From Rough Set Theory to Evidence Theory", In: Advances in the Dempster-Shafer Theory of Evidence, R.R.Yager, M.Fedrizzi, J.Kacprzyk (eds.), John Wiley & Sons, New York 1994, pp. 193-236.

[45] Skowron A., Son N.H., "Quantization of Real Value Attributes", Second Joint Annual Conference on Information Sciences, Wrightsville Beach, North Carolina, September 28-October 1, 1995.

[46] Skowron A., Stepaniuk J., "Generalized Approximation Spaces", Soft Computing, T.Y.Lin, A.M.Wildberger (eds.), Simulation Councils, San Diego 1995, pp. 18-21.

[47] Skowron A., "Synthesis of Adaptive Decision Systems from Experimental Data", In: Proc. of the Fifth Scandinavian Conference on Artificial Intelligence SCAI-95, Aamodt A., Komorowski J.(eds.), IOS Press, Amsterdam 1995, pp. 220–238.

[48] Skowron A., Suraj Z., "A Rough Set Approach to the Real Time State Identification", Bulletin EATCS, 50 (1993) pp. 264-275.

[49] Ślęzak D., "Approximate Reducts in Decision Tables", In: Proc. Information Professing and Management of Uncertainty in Knowledge-Based Systems IPMU-96, to appear.

[50] Tentush I., "On Minimal Absorbent Sets for some Types of Tolerance Relations", Bull. Polish Acad. Sci. Tech., 43(1995),79-88.

[51] Widrow B., Stearns S., "Adaptive Signal Processing", Signal Processing Series, Prentice Hall, Englewood Cliffs, NJ 1985.

[52] Yager R.R., Fedrizzi M., Kacprzyk J., "Advances in the Dempster-Shafer Theory of Evidence", Wiley, New York 1994.

[53] Ziarko W., "Variable Precision Rough Set Model, Journal of Computer and System Sciences", 46(1993), pp. 39-59.

15

COMBINATION OF ROUGH AND FUZZY SETS BASED ON α-LEVEL SETS

Y.Y. Yao

Department of Computer Science, Lakehead University
Thunder Bay, Ontario, Canada P7B 5E1
E-mail: yyao@flash.lakeheadu.ca

ABSTRACT

A fuzzy set can be represented by a family of crisp sets using its α-level sets, whereas a rough set can be represented by three crisp sets. Based on such representations, this paper examines some fundamental issues involved in the combination of rough-set and fuzzy-set models. The rough-fuzzy-set and fuzzy-rough-set models are analyzed, with emphasis on their structures in terms of crisp sets. A rough fuzzy set is a pair of fuzzy sets resulting from the approximation of a fuzzy set in a crisp approximation space, and a fuzzy rough set is a pair of fuzzy sets resulting from the approximation of a crisp set in a fuzzy approximation space. The approximation of a fuzzy set in a fuzzy approximation space leads to a more general framework. The results may be interpreted in three different ways.

1 INTRODUCTION

Theories of rough sets and fuzzy sets are distinct and complementary generalizations of set theory [20, 30]. A fuzzy set allows a membership value other than 0 and 1. A rough set uses three membership functions, a reference set and its lower and upper approximations in an approximation space. There are extensive studies on the relationships between rough sets and fuzzy sets [4, 21, 25, 26]. Many proposals have been made for the combination of rough and fuzzy sets. The results of these studies lead to the introduction of the notions of rough fuzzy sets and fuzzy rough sets [2, 3, 5, 6, 10, 11, 15, 16].

In the theory of fuzzy sets, two equivalent representations of fuzzy sets have been suggested [9]. A fuzzy set can be represented either by a membership function, or by a family of crisp sets called the α-level sets of the fuzzy set. In many situations, it may be more convenient and simpler to use the set-method, i.e., the use of α-level sets of a fuzzy set [13, 19]. In contrast to the functional approach, i.e., the use of membership function of a fuzzy set, the set-method has many advantages in the definition, analysis, and operation with fuzzy concepts [22, 23]. Most of the studies on the combination of rough and fuzzy sets are based on the functional approach. Nakamura [14, 15] used the α-level sets of a fuzzy similarity relation in the study of fuzzy rough sets. The use of set-method in the combination of rough and fuzzy sets is briefly described by Klir and Yuan in a more general framework [9] recently. One may expect the same advantages of set-method in studying these extended notions.

The present study examines some of fundamental issues in the combination of rough and fuzzy sets from the perspective of α-level sets. One of the main objectives is to identify the relationships among rough fuzzy sets, fuzzy rough sets, and crisp sets. This will help us understand the inherent structures of these extended sets. In particular, a rough fuzzy set is defined as the approximation of a fuzzy set in a crisp approximation space, while a fuzzy rough set as the approximation of a crisp set in a fuzzy approximation space. Another objective is to study a more general framework in which a fuzzy set is approximated in a fuzzy approximation space. The results of the approximation can be interpreted in three different ways, a family of rough sets, a family of rough fuzzy sets, and a family of fuzzy rough sets.

2 FUZZY SETS

Let U be a set called universe. A fuzzy set $\mathcal{F}$ on U is defined by a membership function $\mu_{\mathcal{F}} : U \longrightarrow [0,1]$. A crisp set can be regarded as a special case of fuzzy sets in which the membership function is restricted to the extreme points $\{0,1\}$ of $[0,1]$. The membership function of a crisp set is also referred to as a characteristic function. Given a number $\alpha \in [0,1]$, an α-cut, or α-level set, of a fuzzy set is defined by:

$$\mathcal{F}_\alpha = \{x \in U \mid \mu_{\mathcal{F}}(x) \geq \alpha\}, \tag{1.1}$$

which is a subset of U. A strong α-cut is defined by:

$$\mathcal{F}_{\alpha+} = \{x \in U \mid \mu_{\mathcal{F}}(x) > \alpha\}. \tag{1.2}$$

Through either α-cuts or strong α-cuts, a fuzzy set determines a family of nested subsets of U. Conversely, a fuzzy set $\mathcal{F}$ can be reconstructed from its α-level sets as follows:

$$\mu_{\mathcal{F}}(x) = \sup\{\alpha \mid x \in \mathcal{F}_\alpha\}. \tag{1.3}$$

The fuzzy-set equality and inclusion are expressed component-wise as:

$$\begin{aligned} \mathcal{A} = \mathcal{B} &\iff \mu_{\mathcal{A}}(x) = \mu_{\mathcal{B}}(x), \quad \text{for all } x \in U, \\ \mathcal{A} \subseteq \mathcal{B} &\iff \mu_{\mathcal{A}}(x) \leq \mu_{\mathcal{B}}(x), \quad \text{for all } x \in U. \end{aligned} \tag{1.4}$$

Using the notion of α-level sets, they can be equivalently defined by:

$$\begin{aligned} \mathcal{A} = \mathcal{B} &\iff \mathcal{A}_\alpha = \mathcal{B}_\alpha, \quad \text{for all } a \in [0,1], \\ \mathcal{A} \subseteq \mathcal{B} &\iff \mathcal{A}_\alpha \subseteq \mathcal{B}_\alpha, \quad \text{for all } a \in [0,1]. \end{aligned} \tag{1.5}$$

Therefore, we can use either definition of fuzzy sets. Each of these two representations has its advantages in the study of fuzzy sets. One of the main advantage of set based representation is that it explicitly establishes a connection between fuzzy sets and crisp sets. Such a linkage shows the inherent structure of a fuzzy set.

For an arbitrary family of subsets of U, $(\mathcal{A}_\alpha)_\alpha$, $\alpha \in [0,1]$, there is no guarantee that $\mathcal{A}_\alpha$ will be the α-level set of a fuzzy set. The necessary and sufficient conditions on $(\mathcal{A}_\alpha)_\alpha$ are given in the following representation theorems proved by Negoita and Ralescu [17, 18, 22].

Theorem 1 *Let $(\mathcal{A}_\alpha)_\alpha$, $\alpha \in [0,1]$, be a family of subsets of U. The necessary and sufficient conditions for the existence of a fuzzy set $\mathcal{F}$ such that $\mathcal{F}_\alpha = \mathcal{A}_\alpha$, $\alpha \in [0,1]$, are:*

$$\text{(i)} \quad \alpha_1 \leq \alpha_2 \Longrightarrow \mathcal{A}_{\alpha_1} \supseteq \mathcal{A}_{\alpha_2},$$

$$\text{(ii)} \quad \alpha_1 \leq \alpha_2 \leq \ldots, \text{ and } \alpha_n \longrightarrow \alpha \Longrightarrow \bigcap_{n=1}^{\infty} \mathcal{A}_{\alpha_n} = \mathcal{A}_\alpha.$$

Theorem 2 *Let $\psi : [0,1] \longrightarrow [0,1]$ be a given function, and $(\mathcal{A}_\alpha)_\alpha$, $\alpha \in [0,1]$, be a family of subsets of U. The necessary and sufficient conditions for the existence of a fuzzy set $\mathcal{F}$ such that $\mathcal{F}_{\psi(\alpha)} = \mathcal{A}_\alpha$, $\alpha \in [0,1]$, are:*

$$\text{(i')} \quad \psi(\alpha_1) \leq \psi(\alpha_2) \Longrightarrow \mathcal{A}_{\alpha_1} \supseteq \mathcal{A}_{\alpha_2},$$

$$\text{(ii')} \quad \psi(\alpha_1) \leq \psi(\alpha_2) \leq \ldots, \text{ and } \psi(\alpha_n) \longrightarrow \psi(\alpha) \Longrightarrow \bigcap_{n=1}^{\infty} \mathcal{A}_{\alpha_n} = \mathcal{A}_\alpha.$$

An implication of Theorem 1 is that the family of α-level sets of a fuzzy set satisfies conditions (i) and (ii).

There are a number of definitions for fuzzy-set complement, intersection, and union. We choose the standard max-min system proposed by Zadeh [30], in which fuzzy-set operations are defined component-wise as:

$$\begin{aligned} \mu_{\neg\mathcal{A}}(x) &= 1 - \mu_{\mathcal{A}}(x), \\ \mu_{\mathcal{A}\cap\mathcal{B}}(x) &= \min[\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)], \\ \mu_{\mathcal{A}\cup\mathcal{B}}(x) &= \max[\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)]. \end{aligned} \tag{1.6}$$

In terms of α-level sets, they can be expressed by:

$$\begin{aligned} (\neg\mathcal{A})_\alpha &= \neg\mathcal{A}_{(1-\alpha)+}, \\ (\mathcal{A}\cap\mathcal{B})_\alpha &= \mathcal{A}_\alpha \cap \mathcal{B}_\alpha, \\ (\mathcal{A}\cup\mathcal{B})_\alpha &= \mathcal{A}_\alpha \cup \mathcal{B}_\alpha. \end{aligned} \tag{1.7}$$

An important feature of fuzzy-set operations is that they are truth-functional. One can obtain membership functions of the complement, intersection, and union of fuzzy sets based solely on the membership functions of the fuzzy sets involved.

3 ROUGH SETS

Let U denote a finite and non-empty set called the universe, and let $R \subseteq U \times U$ denote an equivalence relation on U, i.e., R is a reflexive, symmetric and transitive relation. If two elements x, y in U belong to the same equivalence class, i.e., xRy, we say that they are indistinguishable. The pair $apr_R = (U, R)$ is called an approximation space. The equivalence relation R partitions the set U into disjoint subsets. It defines the quotient set U/R consisting of equivalence classes of R. The equivalence class $[x]_R$ containing x plays dual roles. It is a subset of U if considered in relation to the universe, and an element of U/R if considered in relation to the quotient set. The empty set $\emptyset$ and equivalent classes are called the elementary sets. The union of one or more elementary sets is called a composed set. The family of all composed sets is denoted by $\mathrm{Com}(apr)$. It is a subalgebra of the Boolean algebra 2^U formed by the power set of U.

Given an arbitrary set $A \subseteq U$, it may not be possible to describe A precisely in the approximation space $apr_R = (U, R)$. Instead, one may only characterize

A by a pair of lower and upper approximations. This leads to the concept of rough sets. In this study, a rough set is interpreted by three ordinary sets:

$$\begin{aligned} \text{Reference set}: &\quad A \subseteq U, \\ \text{Lower approximation}: &\quad \underline{apr}_R(A) = \{x \in U \mid [x]_R \subseteq A\}, \\ \text{Upper approximation}: &\quad \overline{apr}_R(A) = \{x \in U \mid [x]_R \cap A \neq \emptyset\}. \end{aligned} \tag{1.8}$$

By definition, $\underline{apr}_R(A) \subseteq A \subseteq \overline{apr}_R(A)$ and $\overline{apr}_R(A) = \neg\underline{apr}_R(\neg A)$. The pair $(\underline{apr}_R(A), \overline{apr}_R(A))$ is called a rough set with a reference set A.

The characteristic functions of $\underline{apr}_R(A)$ and $\overline{apr}_R(A)$ are called strong and weak membership functions of a rough set [20]. Let μ_A and μ_R denote the membership functions of A and R, respectively. The physical meaning of lower and upper approximations may be understood better by the following two expressions:

$$\begin{aligned} \mu_{\underline{apr}_R(A)}(x) &= \inf\{\mu_A(y) \mid y \in U, (x,y) \in R\}, \\ \mu_{\overline{apr}_R(A)}(x) &= \sup\{\mu_A(y) \mid y \in U, (x,y) \in R\}, \end{aligned} \tag{1.9}$$

and

$$\begin{aligned} \mu_{\underline{apr}_R(A)}(x) &= \inf\{1 - \mu_R(x,y) \mid y \notin A\}, \\ \mu_{\overline{apr}_R(A)}(x) &= \sup\{\mu_R(x,y) \mid y \in A\}. \end{aligned} \tag{1.10}$$

For the two special sets $\emptyset$ and U, definition (1.10) is not defined. In this case, we simply define $\mu_{\underline{apr}_R(U)}(x) = 1$ and $\mu_{\overline{apr}_R(\emptyset)}(x) = 0$ for all $x \in U$. In subsequent discussion, we will not explicitly state these definitions for boundary cases. Based on the two equivalent definitions, lower and upper approximations may be interpreted as follows. An element x belongs to the lower approximation $\underline{apr}_R(A)$ if all elements equivalent to x belong to A. In other words, x belongs to the lower approximation of A if any element not in A is not equivalent to x, namely, $\mu_R(x,y) = 0$. An element x belongs to the upper approximation approximation $\overline{apr}_R(A)$ if at least one element equivalent to x belongs to A. That is, x belongs to the upper approximation of A if any element in A is equivalent to x, namely, $\mu_R(x,y) = 1$. Therefore, the weak and strong membership functions of a rough set can be computed from the membership function of the reference set if the equivalence relation is used to select elements to be considered. Alternatively, they can also be computed from the membership function of the equivalent relation if the reference set is used to select elements to be considered. These two views are important in the combination of rough and fuzzy sets. For convenience, the strong and weak membership functions of a

rough set can be expressed as:

$$\begin{aligned} \mu_{\underline{apr}_R(A)}(x) &= \inf\{\max[\mu_A(y), 1-\mu_R(x,y)] \mid y \in U\}, \\ \mu_{\overline{apr}_R(A)}(x) &= \sup\{\min[\mu_A(y), \mu_R(x,y)] \mid y \in U\}. \end{aligned} \tag{1.11}$$

Although both membership functions of R and A are used, the inf and sup operations are in fact performed only on one membership function.

For two rough sets $(\underline{apr}_R(A), \overline{apr}_R(A))$ and $(\underline{apr}_R(B), \overline{apr}_R(B))$, their intersection and union are given by $(\underline{apr}_R(A \cap B), \overline{apr}_R(A \cap B))$ and $(\underline{apr}_R(A \cup B), \overline{apr}_R(A \cup B))$, with reference sets $A \cap B$ and $A \cup B$, respectively. The rough-set complement is defined by $(\underline{apr}_R(\neg A), \overline{apr}_R(\neg A))$, with a reference set $\neg A$. In contrast to fuzzy sets, rough-set intersection and union are not truth-functional as indicated by the properties:

$$\begin{aligned}
&\text{(R0)} && \underline{apr}_R(\neg A) = \neg\overline{apr}_R(A), \\
& && \overline{apr}_R(\neg A) = \neg\underline{apr}_R(A), \\
&\text{(R1)} && \underline{apr}_R(U) = U, \\
& && \overline{apr}_R(\emptyset) = \emptyset, \\
&\text{(R2)} && \underline{apr}_R(A \cap B) = \underline{apr}_R(A) \cap \underline{apr}_R(B), \\
& && \overline{apr}_R(A \cup B) = \overline{apr}_R(A) \cup \overline{apr}_R(B), \\
& && \underline{apr}_R(A \cup B) \supseteq \underline{apr}_R(A) \cup \underline{apr}_R(B), \\
& && \overline{apr}_R(A \cap B) \subseteq \overline{apr}_R(A) \cap \overline{apr}_R(B), \\
&\text{(R3)} && \underline{apr}_R(A) \subseteq A, \\
& && A \subseteq \overline{apr}_R(A), \\
&\text{(R4)} && A \subseteq \underline{apr}_R(\overline{apr}_R(A)), \\
& && \overline{apr}_R(\underline{apr}_R(A)) \subseteq A, \\
&\text{(R5)} && \underline{apr}_R(A) \subseteq \underline{apr}_R(\underline{apr}_R(A)), \\
& && \overline{apr}_R(\overline{apr}_R(A)) \subseteq \overline{apr}_R(A).
\end{aligned}$$

Property (R0) shows that lower and upper approximations are dual to each other. The above pairs of properties may be considered as dual properties. It is sufficient to only define one of the approximations and to define the other one using property (R0). The two conditions with equality sign in (R2) imply the other two conditions. They state that in general it is impossible to calculate the weak membership function of rough-set intersection and the strong membership function of rough-set union based only on the membership functions of two rough sets involved. One must also take into consideration the interaction between two reference sets, and their relationships to the equivalent classes

of R. Properties (R0)-(R2) follow from the definition of lower and upper approximations. Property (R3) follows from the reflexivity of binary relations, property (R4) follows from the symmetry, and property (R5) follows from the transitivity. By removing the last two properties from (R2), properties (R1)-(R5) form an independent set. They are also sufficient in the sense that any other properties of rough sets can be derived from them [12].

Rough sets are monotonic with respect to set inclusion:

$$\text{(RM1)} \qquad A \subseteq B \Longrightarrow \underline{apr}_R(A) \subseteq \underline{apr}_R(B),$$
$$\text{(RM2)} \qquad A \subseteq B \Longrightarrow \overline{apr}_R(A) \subseteq \overline{apr}_R(B).$$

Let R^1 and R^2 be two equivalence relations on U. R^1 is a refinement of R^2, or R^2 is a coarsening of R^1, if $R^1 \subseteq R^2$. A refinment relation further divides the equivalence classes of a coarsening relation. That is, R^1 is a refinement of R^2 if and only if $[x]_{R^1} \subseteq [x]_{R^2}$ for all $x \in U$. The finest equivalence relation is the identity relation, whereas the coarsest relation is the Cartesian product $U \times U$. Rough sets are monotonic with respect to refinement of equivalence relations. If an equivalence relation R^1 is a refinement of another equivalence relation R^2, for any $A \subseteq U$ we have:

$$\text{(rm1)} \qquad R^1 \subseteq R^2 \Longrightarrow \underline{apr}_{R^1}(A) \supseteq \underline{apr}_{R^2}(A),$$
$$\text{(rm2)} \qquad R^1 \subseteq R^2 \Longrightarrow \overline{apr}_{R^1}(A) \subseteq \overline{apr}_{R^2}(A).$$

Approximation of a set in a refined approximation space is more accurate in the sense that both lower and upper approximations are closer to the set. The two monotonicities of rough sets are useful in the combination of rough and fuzzy sets.

4 COMBINATION OF ROUGH AND FUZZY SETS

The combination of rough and fuzzy sets leads to the notions of rough fuzzy sets and fuzzy rough sets. Different proposals have been suggested for defining such notions. Before presenting rigorous analysis of these concepts based on α-level sets, we briefly review the main results of existing studies.

4.1 Overview

Given an equivalence relation R on a universe U, it defines a quotient set U/R of equivalent classes. For any subset A of the universe, Dubois and Prade [6, 7] defined a rough set as a pair of subsets of U/R:

$$\begin{aligned} \underline{qapr}_R(A) &= \{[x]_R \mid [x]_R \subseteq A\}, \\ \overline{qapr}_R(A) &= \{[x]_R \mid [x]_R \cap A \neq \emptyset\}. \end{aligned} \tag{1.12}$$

The first $[x]_R$ is used as an element of U/R, and the second $[x]_R$ is used as a subset of U. The pair $(\underline{qapr}_R(A), \overline{qapr}_R(A))$ is called a rough set on U/R with reference set A. Although this definition differs from the original proposal of Pawlak [20], they are consistent with each other. Pawlak's lower and upper approximations may be viewed as extensions of $\underline{qapr}_R$ and $\overline{qapr}_R$:

$$\begin{aligned} \underline{apr}_R(A) &= \bigcup_{[x]_R \in \underline{qapr}_R(A)} [x]_R, \\ \overline{apr}_R(A) &= \bigcup_{[x]_R \in \overline{qapr}_R(A)} [x]_R. \end{aligned} \tag{1.13}$$

The notion of rough fuzzy sets defined by Dubois and Prade deals with the approximation of fuzzy sets in an approximation space [6, 7]. Given a fuzzy set $\mathcal{F}$, the result of approximation is a pair of fuzzy sets on the quotient set U/R:

$$\begin{aligned} \mu_{\underline{qapr}_R(\mathcal{F})}([x]_R) &= \inf\{\mu_{\mathcal{F}}(y) \mid y \in [x]_R\}, \\ \mu_{\overline{qapr}_R(\mathcal{F})}([x]_R) &= \sup\{\mu_{\mathcal{F}}(y) \mid y \in [x]_R\}. \end{aligned} \tag{1.14}$$

By using the extension principle, the pair can be extended to a pair of rough sets on the universe U:

$$\begin{aligned} \mu_{\underline{apr}_R(\mathcal{F})}(x) &= \inf\{\mu_{\mathcal{F}}(y) \mid y \in [x]_R\}, \\ \mu_{\overline{apr}_R(\mathcal{F})}(x) &= \sup\{\mu_{\mathcal{F}}(y) \mid y \in [x]_R\}. \end{aligned} \tag{1.15}$$

Similar to equation (1.11), they can be expressed as:

$$\begin{aligned} \mu_{\underline{apr}_R(\mathcal{F})}(x) &= \inf\{\max[\mu_{\mathcal{F}}(y), \mu_R(x,y)] \mid y \in U\}, \\ \mu_{\overline{apr}_R(\mathcal{F})}(x) &= \sup\{\min[\mu_{\mathcal{F}}(y), 1 - \mu_R(x,y)] \mid y \in U\}. \end{aligned} \tag{1.16}$$

The pair $(\underline{qapr}_R(\mathcal{F}), \overline{qapr}_R(\mathcal{F}))$ is called a rough fuzzy set on U/R, and the pair $(\underline{apr}_R(\mathcal{F}), \overline{apr}_R(\mathcal{F}))$ is called a rough fuzzy set on U, with reference fuzzy set $\mathcal{F}$.

The notion of fuzzy rough sets defined by Dubois and Prade [6] is originated from Willaeys and Malvache [24] for defining a fuzzy set with respect to a family of fuzzy sets. It deals with the approximation of fuzzy sets in a fuzzy approximation space defined by a fuzzy similarity relation $\Re$ or defined by a fuzzy partition. We only review the results obtained from a fuzzy similarity relation. A fuzzy similarity relation $\Re$ is a fuzzy subset of $U \times U$ and has three properties:

$$\begin{aligned}
&\text{reflexivity}: \quad \text{for all } x \in U, \mu_{\Re}(x,x) = 1, \\
&\text{symmetry}: \quad \text{for all } x,y \in U, \mu_{\Re}(x,y) = \mu_{\Re}(y,x), \\
&\text{transitivity}: \quad \text{for all } x,y,x \in U, \mu_{\Re}(x,z) \geq \min[\mu_{\Re}(x,y), \mu_{\Re}(y,z)].
\end{aligned}$$

Given a fuzzy similarity relation $\Re$, the pair $apr_{\Re} = (U, \Re)$ is called a fuzzy approximation space. A fuzzy similarity relation can be used to define a fuzzy partition of the universe. A fuzzy equivalence class $[x]_{\Re}$ of elements close to x is defined by:

$$\mu_{[x]_{\Re}}(y) = \mu_{\Re}(x,y). \tag{1.17}$$

The family of all fuzzy equivalence classes is denoted by $U/\Re$. For a fuzzy set $\mathcal{F}$, its approximation in $apr_{\Re}$ is called a fuzzy rough set, which is a pair of fuzzy sets on $U/\Re$:

$$\begin{aligned}
\mu_{\underline{qapr}_{\Re}(\mathcal{F})}([x]_{\Re}) &= \inf\{\max[\mu_{\mathcal{F}}(y), 1 - \mu_{[x]_{\Re}}(y)] \mid y \in U\}, \\
\mu_{\overline{qapr}_{\Re}(\mathcal{F})}([x]_{\Re}) &= \sup\{\min[\mu_{\mathcal{F}}(y), \mu_{[x]_{\Re}}(y)] \mid y \in U\}.
\end{aligned} \tag{1.18}$$

They can be extended to a pair of fuzzy sets on the universe:

$$\begin{aligned}
\mu_{\underline{apr}_{\Re}(\mathcal{F})}(x) &= \inf\{\max[\mu_{\mathcal{F}}(y), 1 - \mu_{\Re}(x,y)] \mid y \in U\}, \\
\mu_{\overline{apr}_{\Re}(\mathcal{F})}(x) &= \sup\{\min[\mu_{\mathcal{F}}(y), \mu_{\Re}(x,y)] \mid y \in U\}.
\end{aligned} \tag{1.19}$$

The approximation of a crisp set in a fuzzy approximation space may be considered as a special case. By comparing equations (1.16) and (1.19), one can conclude that rough fuzzy sets are special cases of fuzzy rough sets as defined by Dubois and Prade. Although the names of rough fuzzy sets and fuzzy rough sets are symmetric, the role played by them are not symmetric.

Nakamura [14, 15] defined a fuzzy rough set by using a family of equivalence relations induced by different level sets of a fuzzy similarity relation $\Re$. For a $\beta \in [0,1]$, the level set $\Re_\beta$ is an equivalence relation. It defines an approximation space $apr_{\Re_\beta} = (U, \Re_\beta)$. The approximation of a fuzzy set $\mathcal{F}$ in $apr_{\Re_\beta}$ turns out to be a rough fuzzy set $(\underline{apr}_{\Re_\beta}(\mathcal{F}), \overline{apr}_{\Re_\beta}(\mathcal{F}))$. The family of rough fuzzy sets, $(\underline{apr}_{\Re_\beta}(\mathcal{F}), \overline{apr}_{\Re_\beta}(\mathcal{F}))$, $\beta \in [0,1]$, is related to a fuzzy rough set of Dubois and

Prade [6, 7]. Lin [11] studied the concept of fuzzy rough sets from the view point of the topology of function spaces. However, it needs to be clarified that fuzzy rough sets called by Lin are in fact rough fuzzy sets called by Dubois and Prade.

All above proposals agree with Pawlak's formulation regarding the interpretation of rough-set intersection and union. These operations are defined based on the approximations of the intersection and union of reference sets or fuzzy sets. Some other studies on the combination of rough and fuzzy sets do not have this feature. Iwinski [8] suggested an alternative definition of rough sets, which is related to but quite different from the one proposed by Pawlak [20]. An Iwinski rough set is defined as a pair of subsets taking from a sub-Boolean algebra of 2^U, without reference to a subset of the universe. For simplicity, we consider the sub-Boolean algebra formed by the set of all composed sets Com(apr). An Iwinski rough set is defined as pair of sets (A_L, A_U) with $A_L \subseteq A_U$ from Com(apr). We may refer to A_L and A_U as lower and upper bounds, respectively. The intersection and union are defined component-wise as:

$$\begin{aligned}(A_L, A_U) \cap (B_L, B_U) &= (A_L \cap B_L, A_U \cap B_U),\\ (A_L, A_U) \cup (B_L, B_U) &= (A_L \cup B_L, A_U \cup B_U).\end{aligned} \tag{1.20}$$

One of the difficulties with such a definition is that the physical meaning of (A_L, A_U) is not entirely clear. This notion may perhaps be interpreted in relation to the concept of interval sets [27]. Biswas [3] adopted the same definition of rough fuzzy sets from Dubois and Prade. For rough-fuzzy-set intersection and union, a definition similar to that of Iwinski is used. Their use of two different models may lead to inconsistency in the interpretation of the concepts involved.

Nanda and Majumdar [16] suggested a different proposal for the definition of fuzzy rough sets by extending the work of Iwinski. Their definition is based on a fuzzification of the lower and upper bounds of Iwinski rough sets. It may be related to the concept of interval-valued fuzzy sets, also known as Φ-fuzzy sets [5, 31]. The same definition was also used by Biswas [2].

Kuncheva [10] defined the notion of fuzzy rough sets which models the approximation of a fuzzy set based on a weak fuzzy partition. It uses measures of fuzzy-set inclusion. A number of of different definitions may indeed be obtained with various measures of fuzzy-set inclusion. The intersection and union operations were not explicitly discussed by Kuncheva. This model is different from the above mentioned works. It is related to the probabilistic rough set model [29] and the variable precision rough set model [32].

The review of existing results shows that the same notions of rough fuzzy set and fuzzy rough sets are used with different meanings by different authors. The functional approaches clearly define various notions mathematically. However, the physical meanings of these notions are not clearly interpreted. In the rest of this section, we attempt to address these issues. The approximation of a fuzzy set in a crisp approximation space is called a rough fuzzy set, to be consistent with the naming of rough set as the approximation of a crisp set in a crisp approximation space. The approximation of a crisp set in a fuzzy approximation space is called a fuzzy rough set. Such a naming scheme has been used by Klir and Yuan [9], and Yao [28]. Under this scheme, these two models are complementary to each other, in a similar way that rough sets and fuzzy sets complementary to each other. In contrast to the proposal of Dubois and Prade [7], rough fuzzy sets are not considered as special cases of fuzzy rough sets. As a result, the framework of the approximation of a fuzzy set in a fuzzy approximation space is considered to be a more general model which unifies rough fuzzy sets and fuzzy rough sets. All these notions are interpreted based on the concept of α-level sets, which may be useful for their successful applications.

4.2 Approximation of fuzzy sets in crisp approximation spaces: rough fuzzy sets

Consider the approximation of a fuzzy set $\mathcal{F} = (\mathcal{F}_\alpha)_\alpha$, $\alpha \in [0,1]$, in an approximation space $apr_R = (U, R)$, where R is an equivalence relation. For each α-level set $\mathcal{F}_\alpha$, we have a rough set:

$$\begin{aligned} \text{Reference set}: &\quad \mathcal{F}_\alpha, \\ \text{Lower approximation}: &\quad \underline{apr}_R(\mathcal{F}_\alpha) = \{x \in U \mid [x]_R \subseteq \mathcal{F}_\alpha\}, \\ \text{Upper approximation}: &\quad \overline{apr}_R(\mathcal{F}_\alpha) = \{x \in U \mid [x]_R \cap \mathcal{F}_\alpha \neq \emptyset\}. \end{aligned} \quad (1.21)$$

That is, $(\underline{apr}_R(\mathcal{F}_\alpha), \overline{apr}_R(\mathcal{F}_\alpha))$ is a rough set with reference set $\mathcal{F}_\alpha$. For the family of α-level sets, we have a family of lower and upper approximations, $(\underline{apr}_R(\mathcal{F}_\alpha))_\alpha$ and $(\overline{apr}_R(\mathcal{F}_\alpha))_\alpha$, $\alpha \in [0,1]$. A crucial question is whether they are the families of α-level sets of two fuzzy sets. Since the family $\mathcal{F}_\alpha$, $\alpha \in [0,1]$, is constructed from a fuzzy set $\mathcal{F}$, we have $\alpha_1 \leq \alpha_2 \Longrightarrow \mathcal{F}_{\alpha_1} \supseteq \mathcal{F}_{\alpha_2}$. By the monotonicity of lower and upper approximations with respect to set inclusion, i.e., properties (RM1) and (RM2), one can conclude that both $(\underline{apr}_R(\mathcal{F}_\alpha))_\alpha$ and $(\overline{apr}_R(\mathcal{F}_\alpha))_\alpha$ satisfy condition (i). The α-level sets of the fuzzy set $\mathcal{F}$ satisfy condition (ii). This implies that both families, $(\underline{apr}_R(\mathcal{F}_\alpha))_\alpha$ and $(\overline{apr}_R(\mathcal{F}_\alpha))_\alpha$, $\alpha \in [0,1]$, satisfy condition (ii). By Theorem 1, they define a pair of fuzzy sets

$\underline{apr}_R(\mathcal{F})$ and $\overline{apr}_R(\mathcal{F})$ such that,

$$\begin{aligned}(\underline{apr}_R(\mathcal{F}))_\alpha &= \underline{apr}_R(\mathcal{F}_\alpha),\\ (\overline{apr}_R(\mathcal{F}))_\alpha &= \overline{apr}_R(\mathcal{F}_\alpha).\end{aligned} \tag{1.22}$$

They are defined by the following membership functions:

$$\begin{aligned}\mu_{\underline{apr}_R(\mathcal{F})}(x) &= \sup\{\alpha \mid x \in (\underline{apr}_R(\mathcal{F}))_\alpha\}\\ &= \sup\{\alpha \mid \underline{apr}_R(\mathcal{F}_\alpha)\}\\ &= \sup\{\alpha \mid [x]_R \subseteq \mathcal{F}_\alpha\},\\ \mu_{\overline{apr}_R(\mathcal{F})}(x) &= \sup\{\alpha \mid x \in (\overline{apr}_R(\mathcal{F}))_\alpha\}\\ &= \sup\{\alpha \mid x \in \overline{apr}_R(\mathcal{F}_\alpha)\}\\ &= \sup\{\alpha \mid [x]_R \cap \mathcal{F}_\alpha \neq \emptyset\}.\end{aligned} \tag{1.23}$$

For an equivalence class $[x]_R$, $[x]_R \subseteq \mathcal{F}_\alpha$ if and only if $\mu_{\mathcal{F}}(y) \geq \alpha$ for all $y \in [x]_R$, and $[x]_R \cap \mathcal{F}_\alpha \neq \emptyset$ if and only if there exists a $y \in [x]_R$ such that $\mu_{\mathcal{F}}(y) \geq \alpha$. Therefore, the membership value of x belonging to $\underline{apr}_R(\mathcal{F})$ is the minimum of membership values of elements in the equivalent class containing x, and the membership value of x belonging to $\overline{apr}_R(\mathcal{F})$ is the maximum. They can be equivalently defined by:

$$\begin{aligned}\mu_{\underline{apr}_R(\mathcal{F})}(x) &= \sup\{\alpha \mid [x]_R \subseteq \mathcal{F}_\alpha\}\\ &= \sup\{\alpha \mid \text{for all } y, y \in [x]_R \Longrightarrow \mu_{\mathcal{F}}(y) \geq \alpha\}\\ &= \inf\{\mu_{\mathcal{F}}(y) \mid y \in [x]_R\}\\ &= \inf\{\mu_{\mathcal{F}}(y) \mid (x,y) \in R\}\\ &= \inf\{\max[\mu_{\mathcal{F}}(y), 1 - \mu_R(x,y)] \mid y \in U\},\\ \mu_{\overline{apr}_R(\mathcal{F})}(x) &= \sup\{\alpha \mid [x]_R \cap \mathcal{F}_\alpha \neq \emptyset\}\\ &= \sup\{\alpha \mid \text{there exists a } y \text{ such that } y \in [x]_R \text{ and } \mu_{\mathcal{F}}(y) \geq \alpha\}\\ &= \sup\{\mu_{\mathcal{F}}(y) \mid y \in [x]_R\}\\ &= \sup\{\mu_{\mathcal{F}}(y) \mid (x,y) \in R\}\\ &= \sup\{\min(\mu_{\mathcal{F}}(y), \mu_R(x,y)) \mid y \in U\}.\end{aligned} \tag{1.24}$$

They may be considered as a generalization of a rough set based on the interpretation of rough sets given by equation (1.9). Moreover, these membership functions can be expressed conveniently by the same formula (1.16).

The use α-level sets provides a clear interpretation of rough fuzzy sets. A fuzzy set $\mathcal{F}$ is described by a pair of fuzzy sets in an approximation space. It lies between the lower and upper approximations $\underline{apr}_R(\mathcal{F})$ and $\overline{apr}_R(\mathcal{F})$. We call

the pair $(\underline{apr}_R(\mathcal{F}), \overline{apr}_R(\mathcal{F}))$ a rough fuzzy set with a reference fuzzy set $\mathcal{F}$. In other words, a rough fuzzy set is characterized by three fuzzy sets:

$$\begin{aligned}
\text{Reference fuzzy set :} \quad & \mu_{\mathcal{F}}, \\
\text{Lower approximation :} \quad & \mu_{\underline{apr}_R(\mathcal{F})}(x) = \inf\{\mu_{\mathcal{F}}(y) \mid y \in U, (x,y) \in R\}, \\
\text{Upper approximation :} \quad & \mu_{\overline{apr}_R(\mathcal{F})}(x) = \sup\{\mu_{\mathcal{F}}(y) \mid y \in U, (x,y)R\} \quad (1.25)
\end{aligned}$$

An α-level set of a rough fuzzy set is defined by in terms of the α-level sets of a fuzzy set $\mathcal{F}$:

$$\begin{aligned}
(\underline{apr}_R(\mathcal{F}), \overline{apr}_R(\mathcal{F}))_\alpha &= (\underline{apr}_R(\mathcal{F}_\alpha), \overline{apr}_R(\mathcal{F}_\alpha)) \\
&= ((\underline{apr}_R(\mathcal{F}))_\alpha, (\overline{apr}_R(\mathcal{F}))_\alpha), \qquad (1.26)
\end{aligned}$$

which is a rough set. By combining the results given in equations (1.5), (1.7), and the properties (R0)-(R5) of rough sets, rough fuzzy sets have properties: for two fuzzy sets $\mathcal{A}$ and $\mathcal{B}$,

$$\begin{aligned}
\text{(RF0)} \quad & \underline{apr}_R(\neg\mathcal{A}) = \neg\overline{apr}_R(\mathcal{A}), \\
& \overline{apr}_R(\neg\mathcal{A}) = \neg\underline{apr}_R(\mathcal{A}), \\
\text{(RF1)} \quad & \underline{apr}_R(U) = U, \\
& \overline{apr}_R(\emptyset) = \emptyset, \\
\text{(RF2)} \quad & \underline{apr}_R(\mathcal{A} \cap \mathcal{B}) = \underline{apr}_R(\mathcal{A}) \cap \underline{apr}_R(\mathcal{B}), \\
& \overline{apr}_R(\mathcal{A} \cup \mathcal{B}) = \overline{apr}_R(\mathcal{A}) \cup \overline{apr}_R(\mathcal{B}), \\
& \underline{apr}_R(\mathcal{A} \cup \mathcal{B}) \supseteq \underline{apr}_R(\mathcal{A}) \cup \underline{apr}_R(\mathcal{B}), \\
& \overline{apr}_R(\mathcal{A} \cap \mathcal{B}) \subseteq \overline{apr}_R(\mathcal{A}) \cap \overline{apr}_R(\mathcal{B}), \\
\text{(RF3)} \quad & \underline{apr}_R(\mathcal{A}) \subseteq \mathcal{A}, \\
& \mathcal{A} \subseteq \overline{apr}_R(\mathcal{A}), \\
\text{(RF4)} \quad & \mathcal{A} \subseteq \underline{apr}_R(\overline{apr}_R(\mathcal{A})), \\
& \overline{apr}_R(\underline{apr}_R(\mathcal{A})) \subseteq \mathcal{A}, \\
\text{(RF5)} \quad & \underline{apr}_R(\mathcal{A}) \subseteq \underline{apr}_R(\underline{apr}_R(\mathcal{A})), \\
& \overline{apr}_R(\overline{apr}_R(\mathcal{A})) \subseteq \overline{apr}_R(\mathcal{A}).
\end{aligned}$$

Rough fuzzy sets are monotonic with respect to fuzzy set inclusion: namely, for two fuzzy sets $\mathcal{A}, \mathcal{B}$,

$$\begin{aligned}
\text{(RFM1)} \quad & \mathcal{A} \subseteq \mathcal{B} \Longrightarrow \underline{apr}_R(\mathcal{A}) \subseteq \underline{apr}_R(\mathcal{B}), \\
\text{(RFM2)} \quad & \mathcal{A} \subseteq \mathcal{B} \Longrightarrow \overline{apr}_R(\mathcal{A}) \subseteq \overline{apr}_R(\mathcal{B}).
\end{aligned}$$

They are also monotonic with respect to refinement of equivalence relations. For two equivalence relations R^1 and R^2 and a fuzzy set $\mathcal{F}$, we have:

$$\text{(rfm1)} \qquad R^1 \subseteq R^2 \Longrightarrow \underline{apr}_{R^1}(\mathcal{F}) \supseteq \underline{apr}_{R^2}(\mathcal{F}),$$

$$\text{(rfm2)} \qquad R^1 \subseteq R^2 \Longrightarrow \overline{apr}_{R^1}(\mathcal{F}) \subseteq \overline{apr}_{R^2}(\mathcal{F}).$$

4.3 Approximation of crisp sets in fuzzy approximation spaces: fuzzy rough sets

The concept of approximation spaces can be generalized by using fuzzy relations [1, 6]. Consider a fuzzy approximation space $apr_{\Re} = (U, \Re)$, where $\Re$ is a fuzzy similarity relation. Each of $\Re$'s β-level sets is an equivalence relation [15]. One can represent $\Re$ by a family of equivalence relations:

$$\Re = (\Re_\beta)_\beta, \qquad \beta \in [0,1]. \tag{1.27}$$

This family defines a family of approximation spaces:

$$apr_{\Re} = (apr_{\Re_\beta} = (U, \Re_\beta))_\beta, \qquad \beta \in [0,1]. \tag{1.28}$$

Given a subset A of U, consider its approximation in each of the approximation spaces. For a $\beta \in [0,1]$, we have a rough set:

$$\begin{aligned} &\text{Reference set :} && A \subseteq U, \\ &\text{Lower approximation :} && \underline{apr}_{\Re_\beta}(A) = \{x \in U \mid [x]_{\Re_\beta} \subseteq A\}, \\ &\text{Upper approximation :} && \overline{apr}_{\Re_\beta}(A) = \{x \in U \mid [x]_{\Re_\beta} \cap A \neq \emptyset\}. \end{aligned} \tag{1.29}$$

With respect to a fuzzy approximation space, we obtain a family of rough sets:

$$(\underline{apr}_{\Re_\beta}(A), \overline{apr}_{\Re_\beta}(A))_\beta, \qquad \beta \in [0,1]. \tag{1.30}$$

Consider the family of lower approximations $(\underline{apr}_{\Re_\beta}(A))_\beta$, $\beta \in [0,1]$. Recall that $\Re_\beta$'s are derived from a fuzzy similarity relation $\Re$. If $\beta_2 \leq \beta_1$, then $\Re_{\beta_1} \subseteq \Re_{\beta_2}$, i.e., $\Re_{\beta_1}$ is a refinement of $\Re_{\beta_2}$. By property (rm1), it follows that $\underline{apr}_{\Re_{\beta_1}}(A) \supseteq \underline{apr}_{\Re_{\beta_2}}(A)$. Let $\psi(\beta) = 1 - \beta$. We have $\psi(\beta_1) \leq \psi(\beta_2) \Longrightarrow \underline{apr}_{\Re_{\beta_1}}(A) \supseteq \underline{apr}_{\Re_{\beta_2}}(A)$. Therefore, property (i') holds. Since $\Re_\beta$'s are derived from a fuzzy similarity relation $\Re$, they satisfy property (ii) in Theorem 1. Combining this result with the definition of lower approximation and property (rm1), one can conclude:

$$\psi(\beta_1) \leq \psi(\beta_2) \leq \ldots \text{ and } \psi(\beta_n) \longrightarrow \psi(\beta) \Longrightarrow \bigcap_{n=1}^{\infty} \underline{apr}_{\Re_{\beta_n}}(A) = \underline{apr}_{\Re_\beta}(A). \tag{1.31}$$

Hence, property (ii′) holds. By Theorem 2, there exists a fuzzy set $\underline{apr}_{\Re}(A)$ such that $(\underline{apr}_{\Re}(A))_{\psi(\beta)} = \underline{apr}_{\Re_\beta}(A)$. Similarly, one can use Theorem 1 to show the existence of a fuzzy set $\overline{apr}_{\Re}(A)$ for the family of upper approximations $(\overline{apr}_{\Re_\beta}(A))_\beta$ such that $(\overline{apr}_{\Re}(A))_\beta = \overline{apr}_{\Re_\beta}(A)$. In this case, property (rm2) is used.

The membership functions of the derived two fuzzy sets are given by:

$$\begin{aligned}
\mu_{\underline{apr}_{\Re}(A)}(x) &= \sup\{\psi(\beta) \mid x \in (\underline{apr}_{\Re}(A))_{\psi(\beta)}\} \\
&= \sup\{1-\beta \mid x \in \underline{apr}_{\Re_\beta}(A)\} \\
&= \sup\{1-\beta \mid [x]_{\Re_\beta} \subseteq A\} \\
&= \sup\{1-\beta \mid \text{for all } y, \mu_{\Re}(x,y) \geq \beta \Longrightarrow y \in A\} \\
&= \sup\{1-\beta \mid \text{for all } y, y \notin A \Longrightarrow \mu_{\Re}(x,y) < \beta\} \\
&= \inf\{1-\mu_{\Re}(x,y) \mid y \notin A\} \\
&= \inf\{\max[\mu_A(y), 1-\mu_{\Re}(x,y)] \mid y \in U\}, \\
\mu_{\overline{apr}_{\Re}(A)}(x) &= \sup\{\beta \mid x \in (\overline{apr}_{\Re}(A))_\beta\} \\
&= \sup\{\beta \mid x \in \overline{apr}_{\Re_\beta}(A)\} \\
&= \sup\{\beta \mid [x]_{\Re_\beta} \cap A \neq \emptyset\} \\
&= \sup\{\beta \mid \text{these exists a } y \text{ such that } \mu_{\Re}(x,y) \geq \beta \text{ and } y \in A\} \\
&= \sup\{\mu_{\Re}(x,y) \mid y \in A\} \\
&= \sup\{\min[\mu_A(y), \mu_{\Re}(x,y)] \mid y \in U\}. \qquad (1.32)
\end{aligned}$$

They may be regarded as a generalization of rough set according to the interpretation given by equation (1.10). They also conform to the general formula (1.16).

We call the pair of fuzzy sets $(\underline{apr}_{\Re}(A), \overline{apr}_{\Re}(A))$ a fuzzy rough sets with reference set A. A fuzzy rough set is characterized by a crisp set and two fuzzy sets:

$$\begin{aligned}
\text{Reference set :} \quad & A \subseteq U, \\
\text{Lower approximation :} \quad & \mu_{\underline{apr}_{\Re}(A)}(x) = \inf\{1-\mu_{\Re}(x,y) \mid y \notin A\}, \\
\text{Upper approximation :} \quad & \mu_{\overline{apr}_{\Re}(A)}(x) = \sup\{\mu_{\Re}(x,y) \mid y \in A\}. \qquad (1.33)
\end{aligned}$$

An β-level set of a fuzzy rough sets is in terms of the β-level sets of the fuzzy similarity relation as:

$$\begin{aligned}
(\underline{apr}_{\Re}(A), \overline{apr}_{\Re}(A))_\beta &= (\underline{apr}_{\Re_\beta}(A), \overline{apr}_{\Re_\beta}(A)) \\
&= ((\underline{apr}_{\Re}(A))_{(1-\beta)}, (\overline{apr}_{\Re}(A))_\beta), \qquad (1.34)
\end{aligned}$$

which is a rough set with reference set A in the approximation space $apr_{\Re_\beta} = (U, \Re_\beta)$.

Based on the properties of rough sets, one can see that fuzzy rough sets satisfy the properties: for $A, B \subseteq U$,

$$
\begin{array}{ll}
\text{(FR0)} & \underline{apr}_{\Re}(\neg A) = \neg\overline{apr}_{\Re}(A), \\
& \overline{apr}_{\Re}(\neg A) = \neg\underline{apr}_{\Re}(A), \\
\text{(FR1)} & \underline{apr}_{\Re}(U) = U, \\
& \overline{apr}_{\Re}(\emptyset) = \emptyset, \\
\text{(FR2)} & \underline{apr}_{\Re}(A \cap B) = \underline{apr}_{\Re}(A) \cap \underline{apr}_{\Re}(B), \\
& \overline{apr}_{\Re}(A \cup B) = \overline{apr}_{\Re}(A) \cup \overline{apr}_{\Re}(B), \\
& \underline{apr}_{\Re}(A \cup B) \supseteq \underline{apr}_{\Re}(A) \cup \underline{apr}_{\Re}(B), \\
& \overline{apr}_{\Re}(A \cap B) \subseteq \overline{apr}_{\Re}(A) \cap \overline{apr}_{\Re}(B), \\
\text{(FR3)} & \underline{apr}_{\Re}(A) \subseteq A, \\
& A \subseteq \overline{apr}_{\Re}(A).
\end{array}
$$

For fuzzy rough sets, we do not have properties similar to (R4) and (R5), or (RF4) and (RF5). This stems from the fact the result of approximating a crisp set is a pair of fuzzy sets. Further approximations of the resulting fuzzy sets are not defined in this framework.

Fuzzy rough sets are monotonic with respect to set inclusion:

$$
\begin{array}{ll}
\text{(FRM1)} & A \subseteq B \Longrightarrow \underline{apr}_{\Re}(A) \subseteq \underline{apr}_{\Re}(B), \\
\text{(FRM2)} & A \subseteq B \Longrightarrow \overline{apr}_{\Re}(A) \subseteq \overline{apr}_{\Re}(B).
\end{array}
$$

They are monotonic with respect to the refinement of fuzzy similarity relations. A fuzzy similarity relation $\Re^1$ is a refinement of another fuzzy similarity relation $\Re^2$ if $\Re^1 \subseteq \Re^2$, which is a straightforward generalization of the refinement of crisp relations. The monotonicity of fuzzy rough sets with respect to refinement of fuzzy similarity relation can be expressed as:

$$
\begin{array}{ll}
\text{(frm1)} & \Re^1 \subseteq \Re^2 \Longrightarrow \underline{apr}_{\Re^1}(A) \supseteq \underline{apr}_{\Re^2}(A), \\
\text{(frm2)} & \Re^1 \subseteq \Re^2 \Longrightarrow \overline{apr}_{\Re^1}(A) \subseteq \overline{apr}_{\Re^2}(A).
\end{array}
$$

4.4 Approximation of fuzzy sets in fuzzy approximation spaces

This section examines the approximation of a fuzzy set in a fuzzy approximation space. In this framework, on the one hand, we have a family of α-level sets $(\mathcal{F}_\alpha)_\alpha$, $\alpha \in [0,1]$, representing a fuzzy set $\mathcal{F}$, on the other hand, we have a family of β-level sets $(\Re_\beta)_\beta$, $\beta \in [0,1]$, prepresenting a fuzzy similarity relation $\Re$. Each α-level set $\mathcal{F}_\alpha$ is a crisp set, and each β-leval relation $\Re_\beta$ is an equivalence relation. Rough sets, rough fuzzy sets, and fuzzy rough sets can therefore be viewed as special cases of the generalized model.

For a fixed pair of numbers $(\alpha, \beta) \in [0,1] \times [0,1]$, we obtain a submodel in which a crisp set $\mathcal{F}_\alpha$ is approximated in a crisp approximation space $apr_{\Re_\beta} = (U, \Re_\beta)$. The result is a rough set $(\underline{apr}_{\Re_\beta}(\mathcal{F}_\alpha), \overline{apr}_{\Re_\beta}(\mathcal{F}_\alpha))$ with the reference set $\mathcal{F}_\alpha$. For a fixed β, we obtain a submodel in which a fuzzy set $(\mathcal{F}_\alpha)_\alpha$, $\alpha \in [0,1]$, is approximated in a crisp approximation space $apr_{\Re_\beta} = (U, \Re_\beta)$. The result is a rough fuzzy set $(\underline{apr}_{\Re_\beta}(\mathcal{F}), \overline{apr}_{\Re_\beta}(\mathcal{F}))$ with the reference fuzzy set $\mathcal{F}$. On the other hand, for a fixed α, we obtain a submodel in which a crisp set $\mathcal{F}_\alpha$ is approximated in a fuzzy approximation space $(apr_{\Re_\beta} = (U, \Re_\beta))_\beta$, $\beta \in [0,1]$. The result is a fuzzy rough set $(\underline{apr}_{\Re}(\mathcal{F}_\alpha), \overline{apr}_{\Re}(\mathcal{F}_\alpha))$ with the reference set $\mathcal{F}_\alpha$. In the generalized model, both α and β are not fixed. The result may be interpreted in three different views.

A family of rough sets: The first interpretation is based on a family of rough sets:

$$(\underline{apr}_{\Re_\beta}(\mathcal{F}_\alpha), \overline{apr}_{\Re_\beta}(\mathcal{F}_\alpha)), \qquad \alpha \in [0,1], \beta \in [0,1], \tag{1.35}$$

which represents the rough set approximation of each α-level set of a fuzzy set $\mathcal{F}$ in an approximation space induced by an β-level relation of a fuzzy similarity relation $\Re$. Under this interpretation, the relationships between different α-level sets of $\mathcal{F}$, and the relationships between different β-level relations of $\Re$, are not taken into consideration.

A family of rough fuzzy sets: In the second view, we consider the following family of rough fuzzy sets:

$$(\underline{apr}_{\Re_\beta}(\mathcal{F}), \overline{apr}_{\Re_\beta}(\mathcal{F}))_\beta, \qquad \beta \in [0,1], \tag{1.36}$$

which takes into consideration the relationships between different α-level sets of a fuzzy set $\mathcal{F}$. The relationships between different β-level relations of a fuzzy similarity relation $\Re$ are not considered.

A family of fuzzy rough sets: By employing the relationship between different β-level relations of a fuzzy relation $\Re$, we obtain a family of fuzzy rough sets:

$$(\underline{apr}_{\Re}(\mathcal{F}_\alpha), \overline{apr}_{\Re}(\mathcal{F}_\alpha)), \qquad \alpha \in [0,1]. \tag{1.37}$$

It does not take account the relationships between different α-level sets of a fuzzy set $\mathcal{F}$.

The above interpretations depend on the ways in which the family of rough sets $(\underline{apr}_{\Re_\beta}(\mathcal{F}_\alpha), \overline{apr}_{\Re_\beta}(\mathcal{F}_\alpha))$, $\alpha \in [0,1]$, $\beta \in [0,1]$, are grouped. An interesting problem is how to take into consideration both relationships between different α-level sets of fuzzy sets, and the relationships between different β-level relations of fuzzy similarity relations. By comparing equations (1.11), (1.24), and (1.32), one can conclude that the membership functions of rough sets, rough fuzzy sets, and fuzzy rough sets can be computed uniformly using the same scheme:

$$\begin{aligned} \mu_{\underline{apr}_\Gamma(\Delta)}(x) &= \inf\{\max[\mu_\Delta(y), 1 - \mu_\Gamma(x,y)] \mid y \in U\}, \\ \mu_{\overline{apr}_\Gamma(\Delta)}(x) &= \sup\{\min[\mu_\Delta(y), \mu_\Gamma(x,y)] \mid y \in U\}, \end{aligned} \tag{1.38}$$

where Γ is a variable that takes either an equivalence relation or a fuzzy similarity relation as its value, and Δ is a variable that takes either a crisp set or a fuzzy set as its value. The same scheme is used by Dubois and Prade [6] to define a pair of fuzzy sets as the result of approximating a fuzzy set in a fuzzy approximation space. This involves the combination of degrees of memberships of a fuzzy set and a fuzzy similarity relation. The physical meaning is not entirely clear. It is questionable that an element with α degree membership belonging to a fuzzy set would have the same physical interpretation as a pair with α degree membership belonging to a fuzzy relation, as the universes of the former and latter are quit different. For this reason, in this study we do not mix the membership functions of a fuzzy set and a fuzzy similarity relation. As seen from equations (1.11), (1.24), and (1.32), the inf and sup operations are indeed performed on one membership function. The use of other membership function is only for the seek of convenience.

5 CONCLUSION

A rough set is the approximation of a crisp set in a crisp approximation space. It is a pair of crisp set. A rough fuzzy set is derived from the approximation of a fuzzy set in a crisp approximation space. It is a pair of fuzzy sets in which

all elements in the same equivalence class have the same membership. The membership of an element is determined by the original memberships of all those elements equivalent to that element. A fuzzy rough set is derived from the approximation of a crisp set in a fuzzy approximation space. It is a pair of fuzzy sets in which the membership of an element is determined by the degrees of similarity of all those elements in the set. By combining these submodels, we have proposed a more generalized model. In this model, we have studied the approximation of a fuzzy set in a fuzzy approximation space. The result of such an approximation is interpreted from three different point of views, a family of rough sets, a family of rough fuzzy sets, and a family of fuzzy rough sets.

By using a family of α-level sets for representing a fuzzy set, this study offered a different and complimentary perspective in understanding the combination of rough and fuzzy sets. More importantly, the investigation has clearly demonstrated the relationships among rough sets, rough fuzzy sets, fuzzy rough sets and ordinary sets. The inherent structures in each of these sets have also been exposed.

Acknowledgements

The author is grateful for financial support from NSERC Canada and the Senate Research Committee of Lakehead University.

REFERENCES

[1] Bezdek, J.C. and Harris, J.D., "Fuzzy partitions and relations: an axiomatic basis for clustering," Fuzzy Sets and Systems, **1**, pp. 111-127, 1978.

[2] Biswas, R., "On rough sets and fuzzy rough sets," Bulletin of the Polish Academy of Sciences, Mathematics, **42**, pp. 345-349, 1994.

[3] Biswas, R., "On rough fuzzy sets," Bulletin of the Polish Academy of Sciences, Mathematics, **42**, pp. 351-355, 1994.

[4] Chanas, S. and Kuchta, D., "Further remarks on the relation between rough and fuzzy sets," Fuzzy Sets and Systems, **47**, pp. 391-394, 1992.

[5] Dubois, D. and Prade, H., "Twofold fuzzy sets and rough sets – some issues in knowledge representation," Fuzzy Sets and Systems, **23**, pp. 3-18, 1987.

[6] Dubois, D. and Prade, H., "Rough fuzzy sets and fuzzy rough sets," International Journal of General Systems, **17**, pp. 191-209, 1990.

[7] Dubois, D. and Prade, H., "Putting rough sets and fuzzy sets together," in: Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, Slowinski, R., Ed., Kluwer Academic Publishers, Boston, pp. 203-222, 1992.

[8] Iwinski, T.B. "Algebraic approach to rough sets," Bulletin of the Polish Academy of Sciences, Mathematics, **35**, 673-683, 1987.

[9] Klir, G.J. and Yuan, B., Fuzzy Sets and Fuzzy Logic, Theory and Applications, Prentice Hall, New Jersey, 1995.

[10] Kuncheva, L.I., "Fuzzy rough sets: application to feature selection," Fuzzy Sets and Systems, **51**, pp. 147-153, 1992.

[11] Lin, T.Y., "Topological and fuzzy rough sets," in: Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, Slowinski, R., Ed., Kluwer Academic Publishers, Boston, pp. 287-304, 1992.

[12] Lin, T.Y. and Liu, Q., "Rough approximate operators: axiomatic rough set theory," in: Rough Sets, Fuzzy Sets and Knowledge Discovery, Ziarko, W.P., Ed., Springer-Verlag, London, pp. 256-260, 1994.

[13] Mizumoto, M. and Tanaka, K., "Some properties of fuzzy numbers," in: Advances in Fuzzy Set Theory and Applications, Gupta, M.M, Ragade, R.K., and Yager, R.R., Eds., North-Holland, New York, pp. 153-164, 1979.

[14] Nakamura, A., "Fuzzy rough sets," Note on Multiple-valued Logic in Japan, **9**, pp. 1-8, 1988.

[15] Nakamura, A. and Gao, J.M., "A logic for fuzzy data analysis," Fuzzy Sets and Systems, **39**, pp. 127-132, 1991.

[16] Nanda, S. and Majumdar, S., "Fuzzy rough sets," Fuzzy Sets and Systems, **45**, pp. 157-160, 1992.

[17] Negoita, C.V. and Ralescu, D.A., "Representation theorems for fuzzy concepts," Kybernetes, **4**, pp. 169-174, 1975.

[18] Negoita, C.V. and Ralescu, D.A., Applications of Fuzzy Sets to Systems Analysis, Wiley, New York, 1975.

[19] Nguyen, H.T., "A note on the extension principle for fuzzy sets," Journal of Mathematical Analysis and Applications, **64**, pp. 369-380, 1978.

[20] Pawlak, Z., "Rough sets," International Journal of Computer and Information Sciences, **11**, pp. 341-356, 1982.

[21] Pawlak, Z., "Rough sets and fuzzy sets," Fuzzy Sets and Systems, **17**, pp. 99-102, 1985.

[22] Ralescu, D.A., "A generalization of the representation theorem," Fuzzy Sets and Systems, **51**, pp. 309-311, 1992.

[23] Uehara, K. and Fujise, M., "Fuzzy inference based on families of α-level sets," IEEE Transactions on Fuzzy Systems, **1**, pp. 111-124, 1993.

[24] Willaeys, D. and Malvache, N., "The use of fuzzy sets for the treatment of fuzzy information by computer," Fuzzy Sets and Systems, **5**, pp. 323-328, 1981.

[25] Wong, S.K.M. and Ziarko, W., "Comparison of the probabilistic approximate classification and the fuzzy set model," Fuzzy Sets and Systems, **21**, pp. 357-362, 1987.

[26] Wygralak, M., "Rough sets and fuzzy sets – some remarks on interrelations," Fuzzy Sets and Systems, **29**, pp. 241-243, 1989.

[27] Yao, Y.Y. "Interval-set algebra for qualitative knowledge representation," Proceedings of the 5th International Conference on Computing and Information, pp. 370-375, 1993.

[28] Yao, Y.Y., "On combining rough and fuzzy sets," Proceedings of the CSC'95 Workshop on Rough Sets and Database Mining, Lin, T.Y. (Ed.), San Jose State University, 9 pages, 1995.

[29] Yao, Y.Y., and Wong, S.K.M., "A decision theoretic framework for approximating concepts," International Journal of Man-machine Studies, **37**, pp. 793-809, 1992.

[30] Zadeh, L.A., "Fuzzy sets," Information and Control, **8**, pp. 338-353, 1965.

[31] Zadeh, L.A., "The concepts of a linguistic variable and its application to approximate reasoning," Information Sciences, **8**, pp. 199-249, 1975.

[32] Ziarko, W., "Variable precision rough set model," Journal of Computer and System Sciences, **46**, pp. 39-59, 1993.

16

THEORIES THAT COMBINE MANY EQUIVALENCE AND SUBSET RELATIONS

Jan M. Żytkow† and Robert Zembowicz

Department of Computer Science, Wichita State University, Wichita, KS 67260-0083

† *also Institute of Computer Science, Polish Academy of Sciences*

zytkow@cs.twsu.edu, robert@cs.twsu.edu

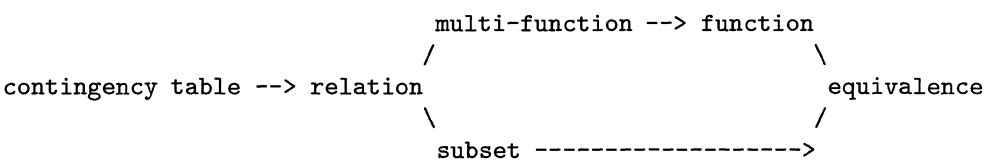
ABSTRACT

Knowledge comes in different species, such as equations, contingency tables, taxonomies, rules, and concepts. We show how starting from contingency tables, simple tests can distinguish various special forms of 2-D knowledge. Our experience in data mining with the application of the 49er system shows that the exploration of even a modestly sized database frequently leads to large numbers of regularities. The problem that we address in this paper comes from the recurring observation of users who show serious confusion when faced with thousands of regularities discovered in a database. As a response, 49er uses tests which classify regularities into different categories and applies automated methods which combine large numbers of regularities in each category into concise, useful forms of taxonomies, inclusion graphs and other multi-dimensional theories. We focus on detection of 2-D regularities which can be represented by equivalence and implication relation, and we show how taxonomies and subset graphs can capture large numbers of regularities in those categories. We illustrate the presented algorithms by applications on two databases.

1 FROM CONTINGENCY TABLES TO OTHER FORMS OF KNOWLEDGE

Elsewhere (Żytkow & Zembowicz, 1993) we argue that contingency tables are the basic form of 2-D regularities while other forms of knowledge can be treated as their special cases. While a relation can be viewed as a subset of the Carte-

sian product of the values of several attributes, a contingency table provides, for each cell in a Cartesian product, the count of all events in the data with the combination of attribute values characteristic for that cell. Relations are special cases, with the count of zero in some cells and without information about the relative frequency of events. The following partial order with contingency tables on the left shows other patterns as special cases:



Contingency tables are easy to generate and investigate. They express statistical regularities, distinguishing statistically probable combinations of events from those improbable (Fienberg, 1980; Gokhale & Kullback, 1978). The majority of contingency tables do not lead to simple patterns. We will focus on those contingency tables which lead to the basic logical relations of implication (inclusion) and equivalence. Many relations in those categories are typically discovered in biological databases, and in various questionnaires. When a number of contingency tables in each of those categories are inferred from data, they can be combined into concept hierarchies and subset graphs.

1.1 Logical relations inferred from contingency tables

Consider two types of tables, depicted in Figure 1, for Boolean attributes A1 and A2. Non-zero numbers of occurrences of particular value combinations are indicated by n1, n2, and n3. Zeros in the cells indicate that the corresponding combinations of values do not occur in data from which the table has been generated. The upper table in Figure 1, for instance, shows that 0 objects are both A1 and non-A2 (labeled $\neg$A2 in the table), while n2 objects are neither A1 nor A2. From the zero values we can infer inductively, with the significance that increases as we consider more data, that these value combinations do not occur in the population represented by data.

The upper table motivates the partition of all data into two classes: (1) objects which are both A1 and A2, and (2) objects which are neither A1 nor A2. Each class has empirical contents, because we can determine class membership by the

A1	0	n1
¬A1	n2	0
	¬A2	A2

The regularity expressed in this table is equivalence:
For all x, (A1(x) if and only if A2(x))
2 classes can be defined: (1) A1 and A2, (2) non-A1 and non-A2

A1	0	n1
¬A1	n2	n3
	¬A2	A2

The regularity expressed in this table is implication:
For all x, (if A1(x) then A2(x)) or equivalently:
For all x, (if non-A2(x) then non-A1(x))

Figure 1 Contingency table that leads to concepts of empirical contents.

value of one attribute, and then predict the value of the other attribute. The lower table in Figure 1 leads to weaker conclusions. Only the values A1 and non-A2 carry predictive contents: objects which are A1, are also A2. Equivalently, objects which are non-A2, are non-A1.

The interpretation of zeros, illustrated in Figure 1 can be generalized to zeros that occur in tables of any size, but for large tables the inferred concepts and their properties may be too many and too weak.

1.2 Approximate equivalence

In real databases, rarely we see regularities without exceptions. Instead of cells with zero counts, we can expect cells with numbers small compared to those in other cells. We want to tolerate limited exceptions. Rather than directly compare the numbers in different cells to determine whether a table approximates equivalence, we use Cramer's V, set at a threshold close to 1.0. The Cramer's V coefficient is based on χ^2, which measures the distance between tables of actual and expected counts. For a given $M_{row} \times M_{col}$ contingency table

$$V = \sqrt{\frac{\chi^2}{N \min(M_{row} - 1, M_{col} - 1)}},$$

where N is the number of records. Cramer's V measures the predictive power of a regularity. The strongest, unique predictions are possible when for each value of one attribute there is exactly one corresponding value of the other attribute. For ideal correlation, χ^2 is equal to $N \min(M_{row} - 1, M_{col} - 1)$, so

Cramer's $V = 1$. On the other extreme, when the actual distribution is equal to expected, then $\chi^2 = 0$ and $V = 0$.

2 FROM EQUIVALENCE RELATIONS TO TAXONOMIES

As an example that illustrates our method for taxonomy formation we selected the small soybean database of 47 records and 35 attributes, because it has been extensively studied, for instance by Stepp (1984) and Fisher (1987).

We used the 49er system (Żytkow & Zembowicz, 1993) to discover statistically significant two-dimensional regularities in soybean data, for all combinations of attributes in all data and in a large number of subsets. Systems such as EXPLORA (Klösgen, 1992) could be also applied to derive the relevant contingency tables. We set 49er parameters so the system seeks only the contingency tables, for which the Cramer's V values ≥ 0.90. An example of such a finding, reported in Table 1, is a regularity between the attributes Stem-Cankers and Fruiting-Bodies, with the Cramer's V rating of 1. Note that the value of Fruiting-Bodies (0 or 1) can be uniquely predicted for each value of Stem-Cankers. 49er found many such regularities, which strongly suggests that the database is conducive to taxonomy formation.

FRUITING-BODIES

1	0	0	0	10
0	10	18	9	0
	0	1	2	3

STEM-CANKERS

Range: All records (47)
Cramer's V = 1.0
Chi-square = 47.0

Table 1 A regularity found in the small soybean dataset by 49er's search for regularities. The numbers in the cells represent the numbers of records with combinations of values indicated at the margins of the table.

After detecting the equivalence relations, 49er uses the following algorithm to build the hierarchy of concepts (for details see Troxel et. al, 1994):

Main Algorithm: Build concept hierarchy
Create hierarchy units from equivalence relations
Merge similar hierarchy units
Sort merged hierarchy units by the decreasing number of descriptors

hierarchy ← single node labeled "ALL"
for each hierarchy unit, add hierarchy unit to hierarchy

In the following subsections we describe details of this algorithm.

2.1 Hierarchy Units Generated from Equivalences

Each contingency table, for which Cramer's $V \geq 0.90$, is used to build an elementary hierarchical unit:

Procedure: Create hierarchy units
for each regularity
if regularity strength exceeds threshold
Form hierarchy unit from regularity

A hierarchy unit is a simple tree, comprised of 3 classes: the root and two children (see Figure 2). The root is labeled with the description of the class of records, in which the regularity holds. Each child is labeled with the descriptors which hold for that child based on the considered regularity. An example of a descriptor is Stem-Cankers(0,1,2): "the values of Stem-Canker are 0, 1, or 2". The children classes are approximately disjoint and they exhaustively cover the range of the regularity.

In our example, the contingency table of Table 1 contains the knowledge that Fruiting-Bodies (the vertical coordinate) has the value of 1, if and only if Stem-Cankers (the horizontal coordinate) has the value 3. Knowing all the other values of both attributes, this is equivalent to "the value of Fruiting-Bodies is 0 if and only if the value for Stem-Cankers falls in the range of 0,1,2". The corresponding hierarchy unit is depicted in the left part of Figure 2. Each class in that unit contains the right attribute/value combination from the corresponding contingency table.

2.2 Merging the Hierarchy Units

If the same class can be derived from different regularities, it will occur in different hierarchy units and will be characterized by many descriptors. To

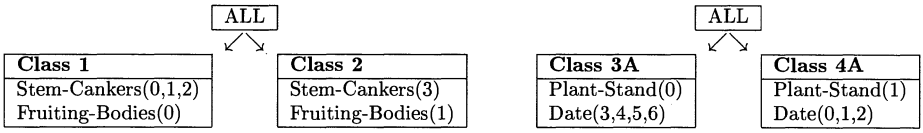


Figure 2 Examples of two hierarchical units built from the regularities for (i) Fruiting-Bodies and Stem-Cankers, and (ii) Plant-Stand and Date. Both regularities hold for all data, hence the root is ALL in both cases.

identify different occurrences, after each hierarchy unit is created, it is compared to all other units that apply to the same range of records, in search for common descriptors. If they were found in the same data set and have a common descriptor (the same attribute and equal value sets), the classes are identical (approximately identical, because of exceptions, as discussed above). The complementary classes must be also (approximately) identical. Both hierarchy units are collapsed into one and the descriptors for the corresponding children are merged (Figure 3).

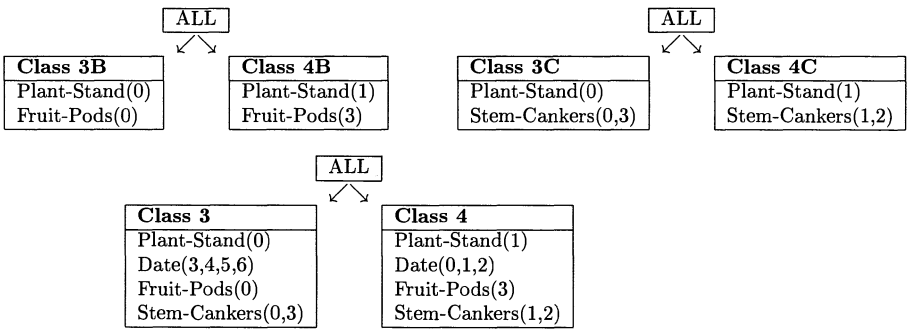


Figure 3 Hierarchy units formed from the regularities between (i) Plant-Stand and Fruit-Pods (upper left) and (ii) Plant-Stand and Stem-Cankers (upper right), share the same descriptor (Plant-Stand) as the hierarchy unit found for the regularity between Plant-Stand and Date (right part of Figure 2). All three hierarchy units are merged together (lower part).

Procedure: Merge similar hierarchy units

for each pair of hierarchy units

if both units share a common descriptor in each subclass

Merge two hierarchy units into one

The same algorithm applies recursively to regularities found in subsets of data. For a regularity in a subset described by condition C, the root of the hierarchy unit is labeled by C (in Figure 4, C =Fruit-Pods(0)).

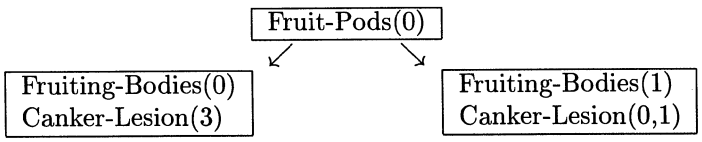


Figure 4 A hierarchy unit over a subrange of the soybean data defined by the descriptor Fruit-Pods(0).

The search through the soybean database produced 15 regularities holding for all data, initially leading to 30 classes and to 8 classes (4 hierarchy units) after merging.

2.3 Taxonomy Formation

The following procedure transform the list of hierarchy units into a multi-level taxonomy, which is exhaustive and (approximately) disjoint at each level:

Procedure: Add hierarchy unit to hierarchy
for each leaf in the hierarchy
Attach children in the hierarchy unit to the leaf
Remove children incompatible with the path to the root
if there is only one child left, merge this child with its parent

For our soybean example, this algorithms puts classes 5 and 6 at the uppermost level of the hierarchy tree, then classes 3 and 4, etc. (Table 2). We position the classes with greater number of descriptors above those with less descriptors, to minimize the number of times each descriptor occurs in the taxonomy.

Some nodes in the nascent hierarchy can be empty. This possibility is examined as soon as new node is added, by computing the intersection of the value sets for each common attributes from the new node upward to the root of the taxonomy. If for a common attribute this intersection is empty, no objects in the dataset can possibly belong to the new node. This node is then eliminated. For example, Class 6 contains the attribute Stem-Cankers with the value range (0), and under it Class 4 contains the same attribute with the value range (1,2).

No records in the data can be in both these classes simultaneously, therefore the class 4 under Class 6 is eliminated (shown in *italic* in Table 2).

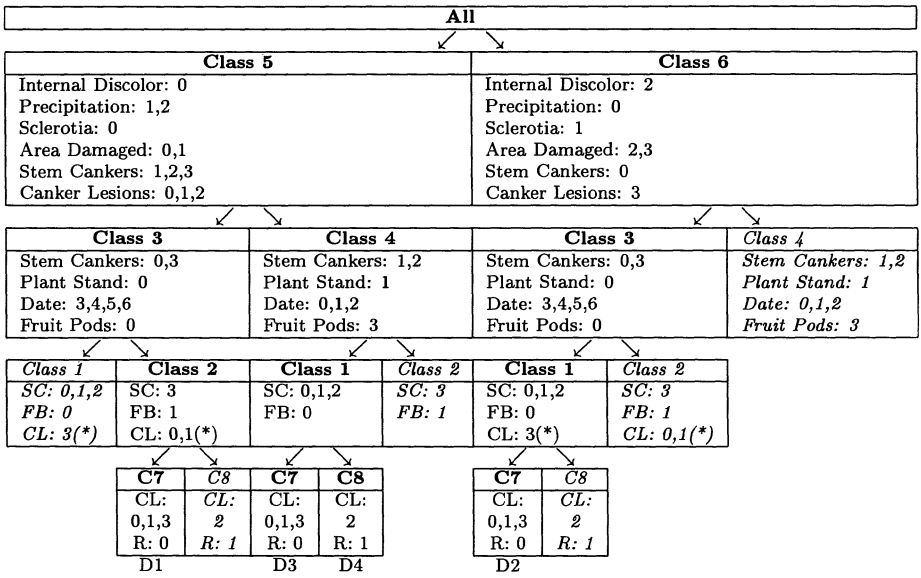


Table 2 The taxonomy generation process, depicted from the top (Classes 5 and 6) till the bottom (Classes 7 and 8). Empty classes are shown in *italic* font. Abbreviations: FB = Fruiting-Bodies; R = Roots; SC = Stem-Cankers; CL = Canker-Lesions

In Table 2 we have shown only the use of regularities for all data, with one exception, marked with (*). The values of Canker-Lesion (CL-3 and CL-0,1), shown in two locations under Class 3, come from a regularity found in the subset of data defined by Fruit-Pods=0. That regularity, depicted in Figure 4, links Canker-Lesion to Fruiting-Bodies. Since the same values of Fruiting-Bodies define Class 1 and Class 2, the corresponding values of Canker-Lesion become inferred descriptors in the subclasses of Class 1 and Class 2 within Class 3.

When after elimination of empty classes only one child class remains under a parent class, the descriptors of this class are added to the descriptor list of the parent class, expanding the intent of the parent node. The descriptors acquired from the lower class become inferred properties in the parent class, because all

objects in the parent class also belong to the remaining lower class. We see in Table 2 that Class 4 is eliminated under Class 6, therefore any object in Class 6 is included in Class 3. However, Class 3 also contains objects that belong in Class 5, so the merged properties from Class 3 to Class 6 cannot be definitional, but merely inferred. The descriptors of Class 7 and Class 1 under Class 6 are also included as inferred descriptors for Class 6.

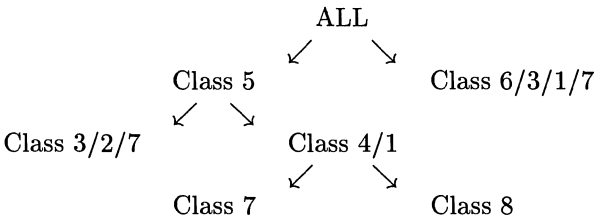


Figure 5 The finished taxonomy, after pruning empty nodes and merging single children.

After all the pruning, we are left with the four nodes at the bottom level in Table 2. These nodes, along with three internal nodes (cf Figure 5) form the taxonomy. We can hypothesize that this taxonomy describes the natural divisions of the soybean database's diseases. It turns out that the extents of the four leaf concepts in our taxonomy are equal to the four diseases, listed under each leaf in Table 2 as D1 through D4. We can hypothesize that the internal nodes correspond to natural classes of diseases.

Our algorithm places concepts with higher empirical contents at the top of the taxonomy. This minimizes the number of descriptors which must be stored in the taxonomy. Our taxonomy, shown in Table 2 requires 33 descriptors, while a taxonomy which puts classes with the fewest attributes at the top (classes 7&8, followed by classes 1&2, 3&4, and 5&6 at the bottom, requires 44 descriptors.

2.4 Empirical contents of taxonomy

The children classes in each hierarchy unit contain a number of descriptors. In our soybean example, after all possible merges of hierarchy units, Classes 3 and 4 each contain four descriptors (Figure 3). Each of those descriptors can be used as definitional and all other descriptors can be deduced.

Each definitional descriptor D for a class C in a hierarchy, under a parent node P, is sufficient to determine whether a given record belongs to C only within the range of P. To obtain a complete definition of C, we must use D

Attribute	Attribute values for different diseases			
	Disease 1	Disease 3	Disease 4	Disease 2
Internal Discolor	**0**	**0**	**0**	**2**
Precipitation	*1,2*	*1,2*	*1,2*	*0*
Sclerotia	*0*	*0*	*0*	*1*
Area Damaged	*0,1*	*0,1*	*0,1*	*2,3*
Stem Cankers	*1,2,3**	*1,2,3**	*1,2,3**	*0*
Canker Lesions	*0,1,2**	*0,1,2**	*0,1,2**	*3*
Stem Cankers	**3**	**1,2**	**1,2**	
Plant Stand	*0*	*1*	*1*	*0*
Date	*3,4,5,6*	*0,1,2*	*0,1,2*	*3,4,5,6*
Fruit Pods	*0*	*3*	*3*	*0*
Fruiting Bodies	*1*	*0*	*0*	*0*
Canker Lesions	*0,1*			
Canker Lesions		**0,1**	**2**	
Roots	*0*	*0*	*1*	*0*
Number of records	10	10	17	10
Number of definitional descriptors	2	3	3	1
Number of inferred descriptors	9	8	8	10
Empirical Contents	8.4	7.8	7.8	12.4

Table 3 Empirical contents for the soybean example. Attributes chosen as definitional are marked by both underline and bold typeface of the corresponding values, while inferred attributes—by slanted typeface. Descriptors marked with an asterisk (*) must not be included into the sum of the inferred descriptors because they repeat in the same column and should not be counted twice.

in conjunction with the definition of the range of the hierarchy unit, which can be assembled from descriptors which are definitional for each node on the path from the root. Being able to make a choice of a definitional descriptor at each level, we can assemble the complete definition in a very flexible way.

We measure the empirical contents (EC) of a class as

$$EC = -\sum_{i \epsilon I} \log_2 \frac{\text{Number of values predicted by i}}{\text{Number of all values of attribute in i}},$$

the sum is computed for the set I of all inferred descriptors for that class. Table 3 shows the empirical contents of each of the four soybean diseases, based on Table 2.

3 FROM SUBSET TABLES TO SUBSET NETWORKS

We will now describe a way to detect and combine many subset relations.

3.1 Testing for subset

The inclusion relation between the scopes of two Boolean attributes may be determined from the "triangular", 4-cell contingency tables. For example, the following table represents inclusion of attribute A in B, $A \subset B$,

A	a	0
$\neg A$	b	d
	B	$\neg B$

a, b, d indicate non-zero counts of records, $c = 0$. This table could be also interpreted as implication of property B from property A, that is by $A \Rightarrow B$.

Four inclusion cases are possible for 2×2 contingency tables, depending on which cell has zero counts. To allow some small level of noise and error, we admit some exceptions. For example, instead of $c = 0$ we require $max(c/(a + c), c/(d + c)) < \delta$, where δ is the tolerance (typically 10%).

Similarly to equivalence, subset relation is not limited to Boolean attributes and can be inferred from large contingency tables. For instance, if a table includes 0 counts for the combination of values d_3 and e_7 of attributes D and E, then $D = d_3 \subset \neg E = e_7$. If D and E have many values, there is little practical value in such subset relations because they cover only very limited cases and possess little empirical content.

3.2 Inclusion network

A significant number of 2-D regularities in a database can represent inclusions (Moraczewski et. al, 1995). Inclusions can be conveniently combined into an inclusion graph, in which vertices represent attributes while directed edges stand for inclusion between extensions of attributes.

Constructing the inclusion graph 49er stores equivalent attributes on the same node to reduce the subset edges. For example, if $A \subset B$ and $B \equiv C$, then $A \subset C$ is removed. Transitivity of the inclusion also permits reduction of the graph: if $A \subset B$ and $B \subset C$, then the inclusion $A \subset C$ is removed. These steps reduce the number of nodes and edges in the inclusion graph while preserving its predictive content.

To divide the typically large inclusion graphs into meaningful pieces, 49er uses the maximum subsets, which are not subsets of any other node in the inclusion graph. For each maximum node M, it extracts the subgraph G_M of all subsets of M. Graphs G_M typically have strong domain-specific interpretation, representing, for instance, all species which inhabit various selective parts of the environment settled by the species M. After removing all edges shared with any other subgraph $G_{M'}$, the remaining subgraph G^*_M has the interpretation of all species which are more selective than M but not more selected than any other species, except those under M. Each G^*_M graph can be interpreted in domain-specific terms. The following algorithm describes inclusion graph formation and analysis.

Algorithm: Build Inclusion Graph and extract subgraphs G^*_M
given the list of significant contingency tables
Select inclusions from contingency tables
Build inclusion graph
Remove redundant edges
for each maximum node M
$G_M \leftarrow$ graph of all subsets of M
$G^*_M \leftarrow G_M$ after removing edges shared with any other graph $G_{M'}$

3.3 Geobotanical database exploration

Consider application of 49er on a geobotanical database that resulted from 20-year studies of Warsaw flora (Sudnik-Wójcikowska 1987) and consists of 1181 Boolean attributes (presence of plant taxa in a given area) and 225 records (areas in the grid covering the whole city). The search in the space of 2-D attribute combinations without slices evaluated $0.5 \times 1181 * 1180 \approx 7 \times 10^5$ hypotheses. Although many interesting regularities are usually found in data subsets, the number of attributes in this database prohibits data slicing. Allowing only a partition into two slices per attribute leads to $0.5 \times 1180 \times 1179 \times 2 \times 1181 \approx 1.6 \times 10^9$ hypotheses to be tested. 49er would take some 2 years to complete this task. In 10 hour search without slices 49er discovered 16577

statistically significant contingency tables. 49er determined that 594 tables capture equivalences of attributes, and applied the taxonomy formation method described in the previous section. The merging of hierarchy units resulted in interesting discoveries. One grid element included 29 species which turned out to be not established permanently in the flora; it showed that they occur in the vicinity of a mill. Another 9-attribute class, occupying just two grid elements, contained species fugitive from gardens. Several 6-attribute classes have been also detected. Since these and all other merged hierarchy units were extremely unbalanced, and hierarchy formation did not lead to any node reduction, the resultant multi-level hierarchy has not been interesting.

6364 contingency tables passed the test for approximate subset. After merging equivalent nodes and removing redundant edges, this number was reduced to 2475. 2358 contingency tables express "positive" inclusions (that is, $A \subset B$) while the remaining 117 represent exclusions of ranges of the corresponding plant taxa A and B ($A \subset \neg B$ or $\neg A \subset B$). These are not used in the inclusion graph. The algorithm described earlier in this chapter formed 302 subgraphs G^*_M, each corresponding to one maximum element M. The species in each G^*_M are characteristic to a specific environment. The further away from M, the narrower is the ecological amplitude of the species. For instance, in a subgraph of species preferring moist areas, the root corresponds to a common meadow plant *Ranunculus acris*, growing on wet soils, that is missing only in the downtown and several dryer areas, while one of the lowest descendents is *Calla palustris*, growing only in peat-bogs, very wet, natural habitats (Moraczewski et. al, 1995).

4 SUMMARY

We have presented an algorithm for conceptual hierarchy formation from knowledge in the form of contingency tables. We used the soybean database as an example. It turned out that four diseases hidden in the soybean data coincide with the four leaves in taxonomies generated by our algorithm. In contrast to fixed rules which define concepts in many machine learning systems, our approach offers the choice among definitional descriptors in selecting a recognition procedure. Depending on the observable concepts available in a given situation, alternative definitional descriptors can be used, so that missing values do not pose a problem in our approach in contrast to many machine learning systems.

We also have shown that many subset relations can be combined into subset graph, which is useful in making inferences and interpreting knowledge about many subsets.

Acknowledgments: Special thanks to Kim Swarm and Molly Troxel for their contributions to research on taxonomy formation, and to Ireneusz Moraczewski for his work on subset graphs.

References

1. Fienberg, S.E. 1980. *The analysis of cross-classified categorical data*, MIT Press.
2. Fisher, D.H. 1987. Knowledge Acquisition Via Incremental Conceptual Clustering, *Machine Learning* 2: 139-172.
3. Gokhale, D.V. & Kullback, S. 1978. *The Information in Contingency Tables*, New York: M.Dekker.
4. Klösgen, W. (1992). Patterns for Knowledge Discovery in Databases in: ed *Proceedings of the ML-92 Workshop on Machine Discovery (MD-92)*, Aberdeen, UK. July 4, p.1-10.
5. Moraczewski, I., Zembowicz, R. & Żytkow, J.M. 1995. Geobotanical Database Exploration. In Working Notes of the AAAI Spring Symposium. Forthcoming.
6. Sudnik-Wojcikowska, B. 1987. *The flora of Warsaw and its changes in 19th and 20th Centuries* (in Polish), Warsaw: Warsaw University Publ.
7. Troxel, M., Swarm, K., Zembowicz, R. & Żytkow, J.M. 1994. Concept Hierarchies: a Restricted Form of Knowledge Derived from Regularities. In: Proc. of Methodologies for Intelligent Systems, ed. M. Zemankova & Z. Ras, Springer-Verlag.
8. Żytkow, J., & Zembowicz, R., (1993) Database Exploration in Search of Regularities, *Journal of Intelligent Information Systems*, 2, p.39-81.

PART IV

GENERALIZATION

17

GENERALIZED ROUGH SETS IN CONTEXTUAL SPACES

Edward Bryniarski*
Urszula Wybraniec-Skardowska**

University of Opole, Oleska 48, 45-951 Opole, POLAND

** Department of Computer Science,*
e-mail: edlog@uni.opole.pl
*** Department of Psychology,*
e-mail: uws@uni.opole.pl

ABSTRACT

This paper presents a generalization of Pawlak's conception of rough sets [6] and [7]. It is more general than Pawlak's solution of the problem of the definability of sets, the knowledge of which is incomplete and vague. The authors' conception is based on conception of contextual space [4], which was inspired by Ziarko's approach [12] to rough sets. Rough sets introduced by Pawlak [6] are particular cases of contextual rough sets defined in the contextual approximation space. This space is defined axiomatically by means of so called context relations. Every contextual rough set determined by set X can be determined by the union of the lower approximation of X and a subset of the boundary of X. One of the important notions of the conception is the notion of an element of a contextual rough set which allows for formulating and proving the counterpart of the axiom of extensionality for contextual rough sets.

1 INTRODUCTION

Identification of objects making use of knowledge, knowledge bases, databases, information systems, and expert systems is usually imprecise. Objects about which we have the same knowledge are simply *indiscernible*. This takes place, for example, when a machine for detecting forged banknotes has imprecise information about the notes. It is impossible for it to tell which are true. Another good case in point is a colour-blind person. Objects differentiated only by colours may be indiscernible for her/him. Such a state of affairs influences gradation of accuracy estimates of recognising one set of objects in the *context*

of another, i.e. recognising correlated data about elements of those sets (cf. [12]). For example for sets X and Y one can assume the following degree measurement from 0 to 4:
0 – all elements of set X are elements of set Y ($X \subseteq Y$),
1 – almost all elements of set X are elements of set Y,
2 – majority of elements of set X are elements of set Y,
3 – only some elements of set X are elements of set Y,
4 – no elements of set X are elements of set Y ($X \cap Y = \emptyset$).

The above degree measurement can be interpreted as the degree measurement of errors, which are made while stating something about the relationship between the elements of sets X and Y, when in reality there is an inclusion $X \subseteq Y$. So, by claiming that "almost all elements of set X are elements of set Y" we make the slightest error, and by claiming that "no elements of set X are elements of set Y" we make the biggest error.

Now the question arises whether it is possible to define an accuracy degree measurement of recognising the relationships between the elements of sets in an automatic way (by means of a machine). In authors' opinion the answer for this question is positive. Theoretical foundations for such an answer can be formulated within the proposed framework of rough set theory (cf. [7], [12]). First, we need to make some generalizations. These will concern the basic definitions of rough set theory.

Rough set theory in the present article is an approach to the discussed solutionof the problem of the definability of sets such that 1) the knowledge of which is incomplete or vague, 2) the knowledge about each set is given independently of the knowledge about any other set, 3) the knowledge about the relationships among the elements is vague or inexact. This theory in itself is complementary to Zadeh's fuzzy set theory [11] and Pawlak's rough set theory [6] and [7]. The authors' considerations on this subject were, however, inspired not only by Pawlak's and Zadeh's approach, but also by the conceptions of Ziarko [12] and Blizard [1] and [2]. The generalization of rough sets proposed in this paper is a result of these considerations.

The paper consists of four sections. In Section 1 we introduce the notion of an approximation space and give some examples of such spaces. In Section 2 we define and characterize the approximating operations, the lower and upper approximations of the defined degree for any set of the universe of a given extended approximation space (called the contextual space). The main notion of a rough set in a given extended approximation space is introduced and cha-

racterized in Section 3. Section 4 contains important definitions and theorems about the membership relations for rough sets.

2 A CONTEXTUAL APPROXIMATION SPACE

We know from experience that when we try to define a set about which our knowledge is incomplete or vague, we have to refer to our knowledge about another set. We then say that the first set is in the *context* with the other set. The less we know about the first set in relation to the other set, the more significant the context is. We say that the first set is in the context with the other set to the degree i, if i is a degree measurement of the *significance level of context.*

Let us now introduce an axiomatically extended notion of an approximation space. Usually, by an approximation space we understand a pair $< U, R >$ (see [6]), where U is a nonempty set and R an equivalence relation in this set, or a pair $< U, C >$, where U is a nonempty set, and C is its covering. Extending the latter interpretation of an approximation space with the context relations, we get a system CAS called the *contextual approximation space*, which we define in the following way:

Definition 1
Let $CAS = < U; (I, \leq, \{\emptyset, \varepsilon\}); \{\subseteq_i\}_{i \in I}; C >$ *be the ordered system, where* U *is a nonempty set, called the universe;* $(I, \leq, \{\emptyset, \varepsilon\})$ *is an ordered system such that:* I *is a set,* $\leq$ *– a linear order relation in the set* I, $\emptyset$ *– the smallest element in* I, ε *– the greatest element in* I; $\{\subseteq_i\}_{i \in I}$ *is a set of relations in the power set* $P(U)$ *of the universe* U; C *is a partition of the universe* U. *The system* CAS *is the contextual approximation space if and only if the following properties hold: for any* $X, Y \subseteq U$ *and for all* $i, j \in I$

(a) $$\exists k \in I(X \subseteq_k Y),$$

(b) $$X \subseteq_\emptyset Y \Leftrightarrow X \subseteq Y,$$

(c) $$X \neq \emptyset \Rightarrow (X \cap Y = \emptyset \Leftrightarrow \forall k \in I(X \subseteq_k Y \Rightarrow k = \varepsilon)),$$

(d) $$X \subseteq_i Y \wedge i \leq j \Rightarrow X \subseteq_j Y,$$

(e) $$X \subseteq_i Y \Leftrightarrow X \subseteq_i X \cap Y.$$

The relations of the set $\{\subseteq_i\}_{i\in I}$ *are called the context relations of a certain degree. The expression* $X \subseteq_i Y$ *is read:* X *is in the context of* Y *to the degree* i.

From conditions (b), (c) and (a) of Definition 1, immediately follows

Proposition 1
In CAS the following conditions are satisfied:

(a) $$x \in X \Rightarrow \{x\} \subseteq_\varnothing X,$$

(b) $$X \neq \emptyset \wedge X \cap Y \neq \emptyset \Rightarrow X \subseteq_\varepsilon Y,$$

(c) $$X \neq \emptyset \Rightarrow X \subseteq_\varepsilon \emptyset.$$

Below we give two examples of contextual spaces.

Example 1
Let U be a nonempty finite set, C – a partition of U, c – a measure of the relative degree of inclusion of two subsets of the universe U defined in the following way (see Ziarko [12]): for any X, $Y \subseteq U$

$$c(X,Y) = \begin{cases} 1 - card(X \cap Y)/card(X) & if\ card(X) \neq 0 \\ 0 & if\ card(X) = 0, \end{cases}$$

where 'card' denotes set cardinality,

$$I = \{i \in \mathcal{R} : 0 \leq i \leq 1\}$$

and $\mathcal{R}$ is the set of all real numbers.

The context relations $\subseteq_i$ are defined as follows:

$$X \subseteq_i Y \Leftrightarrow c(X,Y) \leq i,$$

for any $X, Y \in P(U)$ and for all $i \in I$.

Based on this assumption and by Definition 1, we claim that the following system

$$< U; (I, \leq, \{0,1\}); \{\subseteq_i\}_{i \in I}; C >$$

is a contextual approximation space.

Example 2
Let U be a nonempty finite set, $card(U) = n$, $I = \{i \in \mathcal{N} : 0 \leq i \leq n\}$ and C a partition of the set U. The context relations $\subseteq_i$ are defined as follows: for any $X, Y \in P(U)$ and for each $i \in I$

$$X \subseteq_i Y \Leftrightarrow u(X,Y) \leq i,$$

where

$$u(X,Y) = \begin{cases} card(X \setminus Y) & if \ X = \emptyset \ \ or \ \ X \cap Y \neq \emptyset \\ n & if \ X \neq \emptyset \ \ and \ \ X \cap Y = \emptyset. \end{cases}$$

It is easy to see that the system

$$< U; (I, \leq, \{0,n\}); \{\subseteq_i\}_{i \in I}; C >,$$

where the number 0 is the smallest element in I and the number n is the greatest element in I, is a contextual space.

3 APPROXIMATING OPERATIONS

Let CAS be an arbitrarily given contextual approximation space. The idea of the rough set originated by Z. Pawlak [6] consists of the approximation of sets obtained by means of two operations on sets: the lower approximation and the upper approximation.

In this paper referring to the conception originated by W. Ziarko [12] of the lower approximation and the upper approximation we introduce two operations: the lower approximation $\underline{C}_i$ and the upper approximation $\overline{C}_i$ of the degree $i \in I$. In order to define rough sets, the space CAS is extended to the space $CAS^* =< CAS, \underline{C}_i, \overline{C}_i >$ (cf. [9]), and operations $\underline{C}_i, \overline{C}_i$ are defined as follows:

Definition 2
In CAS, for every $X \subseteq U$ and for any $i \in I$, we define the following sets:

(a) $\underline{C}_i(X) = \bigcup\{E \in C : \exists j \in I(E \subseteq_j X \wedge j \leq i)\};$

(b) $$BN_i(X) =$$
$$= \bigcup\{E \in C : \exists j \in I(E \subseteq_j X \wedge \varepsilon > j > i) \wedge \neg\exists j \in I(E \subseteq_j X \wedge j \leq i)\};$$

(c) $$\overline{C}_i(X) = \underline{C}_i(X) \cup BN_i(X);$$

the set $\underline{C}_i(X)$ is called the lower approximation of the degree i of the set X,
the set $BN_i(X)$ is called the boundary of the degree i of the set X,
the set $\overline{C}_i(X)$ is called the upper approximation of the degree i of the set X.

Definition 2(a) says: the lower approximation of the degree i of the set X is the union of all classes of partition C which are in the context of X to a certain degree less than or equal to i. According to Definition 2(b) the boundary of the degree i of the set X is the union of all such classes of C which are not included in the lower approximation of the degree i of X but are in the context of X to a certain degree that is greater than i and less than ε. According to Definition 2(c) the upper approximation of the degree i of the set X is the union of the lower approximation and the boundary of the degree i of X.

On the basis of Definitions 2(a)-(c), 1(b) and 1(c) the following propositions hold:

Proposition 2
In CAS, for every $X \subseteq U$, if $i = \emptyset$, then*

$$\underline{C}_i(X) = \bigcup\{E \in C : E \subseteq X\},$$

$$\overline{C}_i(X) = \bigcup\{E \in C : E \cap X \neq \emptyset\}.$$

One can see that for $i = \emptyset$, we get equalities defining the lower and upper approximations in Pawlak's sense [6].

Proposition 3
In CAS, for any $X, E \subseteq U$ and for any $i \in I$*

(a) $$\underline{C}_\varepsilon(X) = \overline{C}_\varepsilon(X) = U,$$

(b) $$BN_\varepsilon(X) = \emptyset$$

and

if $i \neq \varepsilon$, *then*

(c) $$E \in C \Rightarrow \underline{C}_i(E) = \overline{C}_i(E) = E,$$

(d) $$\underline{C}_i(\emptyset) = BN_i(\emptyset) = \overline{C}_i(\emptyset) = \emptyset.$$

Proposition 4
In CAS, for every* $X \subseteq U$ *and for any* $i \in I$, *the following conditions are satisfied:*

(a) $$\underline{C}_i(X) \subseteq \overline{C}_i(X),$$

(b) $$\underline{C}_i(X) \cap BN_i(X) = \emptyset,$$

(c) $$\underline{C}_i(\underline{C}_i(X)) = \underline{C}_i(X),$$

(d) $$\overline{C}_i(\underline{C}_i(X)) = \underline{C}_i(X),$$

(e) $$BN_i(\underline{C}_i(X)) = BN_i(\overline{C}_i(X)) = \emptyset.$$

Example 3
Let the system

$$< U; (I, \leq, \{1, 9\}); \{\subseteq_i\}_{i \in I}; C >,$$

where $U = I = \{1, 2, 3, ..., 9\}$, $C = \{E_1, E_2, E_3\}$ and $E_1 = \{1, 4, 7\}$, $E_2 = \{2, 5, 8, \}$, $E_3 = \{3, 6, 9, \}$ be a contextual space from Example 2.
Let $X = \{5, 7\}$.
Then

$$u(E_1, X) = 2, \qquad u(E_2, X) = 2, \qquad u(E_3, X) = 9,$$

$$\underline{C}_1(X) = \emptyset, \qquad BN_1(X) = E_1 \cup E_2, \qquad \overline{C}_1(X) = E_1 \cup E_2,$$
$$\underline{C}_3(X) = E_1 \cup E_2, \qquad BN_3(X) = \emptyset, \qquad \overline{C}_3(X) = \underline{C}_3(X).$$

The following theorem defines sufficient conditions for any two sets A and B to be: A – the lower approximation, B – the upper approximation of a degree of a set.

Theorem 1
In CAS, for all sets* $A, B \subseteq U$, *such that*

$$A = \bigcup\{E \subseteq A : E \in C\},$$

$$B = \bigcup\{E \subseteq B : E \in C\},$$

$$A \subseteq B$$

the following property is satisfied: for any $Z \subseteq B \setminus A$ *and for all* $i \in I$ *such that* $i \neq \varepsilon$, *if for every* $E \in C$ *and* $E \subseteq B \setminus A$ *there exists* $j \in I$ *such that*

$$i < j < \varepsilon \quad \text{and} \quad E \subseteq_j Z,$$

and there is no $j \in I$ *such that*

$$j \leq i \quad \text{and} \quad E \subseteq_j Z,$$

then

$$A = \underline{C}_i(A \cup Z) \quad \text{and} \quad B = \overline{C}_i(A \cup Z).$$

The proof of this theorem is given in the authors' paper [5].

4 CONTEXTUAL ROUGH SETS

In this section we introduce the definition of the generalized notion of the rough set. In CAS^*, for every $X \subseteq U$ and for any $i \in I$, we define a rough set, which will be called a *contextual rough set*, in the following way:

Definition 3
The contextual rough set of the degree i *determined by the set* X *is the family*

$$[X]_i = \{Y \subseteq U : \underline{C}_i(Y) = \underline{C}_i(X) \wedge \overline{C}_i(Y) = \overline{C}_i(X)\}.$$

As a particular case for $i = \emptyset$ we obtain the definition of the rough set introduced by Pawlak [6] (cf. also [7]).

Example 4
Using Definition 3 for the set X from Example 3 we get:

$$[X]_1 = \{\{a, b\} : a \in E_1 \wedge b \in E_2\}.$$

Proposition 5
In CAS^*, *for every* $X \subseteq U$ *and for any* $i \in I$,

(a) $$[\emptyset]_i = \{\emptyset\},$$

(b) $$[X]_\varepsilon = [U]_\varepsilon = P(U)$$

and
if $BN_i(X) = \emptyset$, *then*

(c) $$[X]_i = [\underline{C}_i(X)]_i.$$

From the conditions (d), (e) of Definition 1, Definition 2, the axiom of choice and the propositions given above we infer:

Theorem 2
In CAS, for every* $X \subseteq U$ *and for any* $i \in I$, *there exists a set* $Z \subseteq BN_i(X)$ *such that*

(*) $$[X]_i = [\underline{C}_i(X) \cup Z]_i.$$

According to the above theorem every contextual rough set of the established degree i determined by the set X is determined by the union of the lower approximation of the same degree i of X and a subset of the boundary of the degree i of X. A particular case of this theorem has been given by Wybraniec–Skardowska [10].
Proof.
If $BN_i(X) = \emptyset$, then $Z = \emptyset$ and $[X]_i = [\underline{C}_i(X)]_i$ in view of Proposition 5(c).

If $BN_i(X) \neq \emptyset$, i.e.

$$BN_i(X) = \bigcup\{E \in C : \exists j \in I(E \subseteq_j X \wedge \varepsilon > j > i) \wedge \\ \wedge \neg\exists j \in I(E \subseteq_i X \wedge j \leq i)\} \neq \emptyset,$$

then, from the axiom of choice it follows that there exists the choice set $\{S_j\}_{j\in J}$ for the family

(1) $$\{A \subseteq P(U) : \exists E \in C(E \subseteq BN_i(X) \wedge A = \{S \subseteq E : \\ \exists j \in I(E \subseteq_j S \wedge \varepsilon > j > i) \wedge \neg\exists j \in I(E \subseteq_j S \wedge j \leq i)\})\}.$$

Let us assume that

(2) $$Z = \bigcup\{S_j\}_{j\in J}.$$

It is easy to see that from (1) and (2) it follows that for $j \in J$

(3) $$S_j \subseteq Z \subseteq BN_i(X).$$

We will prove that

$$[X]_i = [\underline{C}_i(X) \cup Z]_i .$$

Since $BN_i(X) \neq \emptyset$, thus $i < \varepsilon$, and for any $E \in C$, from the conditions:

$$E \subseteq \underline{C}_i(Y) \Leftrightarrow E \cap \underline{C}_i(Y) \neq \emptyset, \tag{4}$$

$$E \subseteq \underline{C}_i(Y) \Leftrightarrow E \subseteq_i Y, \tag{5}$$

where $Y \subseteq U$, being simple conclusions from Definitions 1(b),(d), 2(a), and from Definition 1 (b), (d), and also Proposition 4(c) and the condition (5), we obtain:

$$E \subseteq \underline{C}_i(X) \Leftrightarrow E \subseteq \underline{C}_i(\underline{C}_i(X)) \Leftrightarrow E \subseteq_i \underline{C}_i(X). \tag{6}$$

In the further part of the proof we will use the following

Lemma

$$E \subseteq_i \underline{C}_i(X) \Leftrightarrow E \subseteq_i \underline{C}_i(X) \cup Z.$$

We omit the proof of this Lemma. It requires of using the formula (6).
From this Lemma and formula (5) we get the following equivalence:

$$E \subseteq \underline{C}_i(X) \Leftrightarrow E \subseteq \underline{C}_i(\underline{C}_i(X) \cup Z).$$

Therefore, by Definition 2(a), we have:

$$\underline{C}_i(X) = \underline{C}_i(\underline{C}_i(X) \cup Z). \tag{7}$$

The definitions of a partition of a set and the set Z (see (2)) and Definition 2(b) imply that for any $E \in C$

$$E \subseteq BN_i(X) \Leftrightarrow E \cap Z = E \cap (\underline{C}_i(X) \cup Z) \neq \emptyset.$$

Thus, by the definition (2) of the set Z, Definition 1(e) and Definition 2(b), the following equivalences hold: for any $E \in C$

$$E \subseteq BN_i(X) \Leftrightarrow E \cap Z \neq \emptyset \Leftrightarrow$$

$$\Leftrightarrow \exists j \in I(E \subseteq_j E \cap Z \wedge \varepsilon > j > i) \wedge$$

$$\wedge \neg\exists j \in I(E \subseteq_j E \cap Z \wedge j \leq i)) \Leftrightarrow$$

$$\Leftrightarrow \exists j \in I(E \subseteq_j E \cap (\underline{C}_i(X) \cup Z) \wedge \varepsilon > j > i) \wedge$$

$$\wedge \neg\exists j \in I(E \subseteq_j E \cap (\underline{C}_i(X) \cup Z) \wedge j \leq i)) \Leftrightarrow$$

$$\Leftrightarrow \exists j \in I(E \subseteq_j \underline{C}_i(X) \cup Z \wedge \varepsilon > j > i) \wedge$$

$$\wedge \neg\exists j \in I(E \subseteq_j \underline{C}_i(X) \cup Z \wedge j \leq i)) \Leftrightarrow$$
$$\Leftrightarrow E \subseteq BN_i(\underline{C}_i(X) \cup Z).$$

Hence we have:

$$BN_i(X) = BN_i(\underline{C}_i(X) \cup Z). \tag{8}$$

Thus, in view of Definition 2(c) and (7), (8), we get the equality:

$$\overline{C}_i(X) = \overline{C}_i(\underline{C}_i(X) \cup Z) \tag{9}$$

and from the equalities (7) and (9) on the basis of Definition 3, the condition *(*)*. ■

The proved theorem enables assigning the representative of the rough set $[X]_i$ only if the lower approximation and the boundary of the degree i of X is known.

5 CONTEXTUAL ROUGH MEMBERSHIP RELATION

Now we will formulate the definitions of an element of the contextual rough set. These definitions are based on the intuition: a set which is an element of a rough set can be understood as a set of indiscernible objects. The definition of an element of the contextual rough set is analogous to the one introduced by Bryniarski [3].

Definition 4
In CAS^*, for any $X, Y \subseteq U$ and for all $i \in I$, X is an element of the contextual rough set $[Y]_i$, symbolically: $X \in_C [Y]_i$, if and only if

(a)
$$\exists j \in I((X \in C \vee \exists E \in C(\overline{C}_j(X) = E)) \wedge$$
$$\wedge \underline{C}_j(X) \subseteq \underline{C}_i(Y) \wedge \overline{C}_j(X) \subseteq \overline{C}_i(Y))$$

and X is an element of the degree $n \in I$ of the contextual rough set $[Y]_i$, symbolically: $X \in_n [Y]_i$, if and only if

(b)
$$(X \in C \vee \exists E \in C(\overline{C}_n(X) = E)) \wedge$$
$$\wedge \underline{C}_n(X) \subseteq \underline{C}_i(Y) \wedge \overline{C}_n(X) \subseteq \overline{C}_i(Y).$$

The relation $\in_C$ will be called the membership relation and the relation $\in_n$ will be called the membership relation of the degree n.

Proposition 6
In CAS, for any $E, X, Y \subseteq U$, for all $i, j, k \in I$, the following conditions are satisfied:*

$$(a) \qquad X \in_k [Y]_i \Rightarrow X \in_C [Y]_i,$$

$$(b) \qquad X \in_C [Y]_i \Rightarrow \exists n \in I(X \in_n [Y]_i),$$

$$(c) \qquad E \in C \wedge E \in_k [Y]_i \wedge k < \varepsilon \wedge j < \varepsilon \Rightarrow E \in_j [Y]_i,$$

$$(d) \qquad \forall x \in X(\{x\} \in C \Rightarrow \{x\} \in_\emptyset [X]_i),$$

$$(e) \qquad i < \varepsilon \wedge E \in C \Rightarrow (E \subseteq \underline{C}_i(X) \Leftrightarrow E \in_i [X]_i),$$

$$(f) \qquad E \in C \Rightarrow (E \subseteq \underline{C}_i(X) \Leftrightarrow E \in_C [X]_i).$$

Proof.
Implications (a) and (b) follow immediately from Definition 4.

(c) Let $E \in C$, $E \in_k [Y]_i$ and $j < \varepsilon$, $k < \varepsilon$. From Proposition 3(c) we obtain:

$$\underline{C}_k(E) = \overline{C}_k(E) = \underline{C}_j(E) = \overline{C}_j(E).$$

From the above and Definition 4(b), we get: $E \in_j [Y]_i$.

(d) Assuming that $x \in X$ and $\{x\} \in C$, on the basis of Proposition 1(a), we have $\{x\} \subseteq_\emptyset X$. Thus, from Definitions 2(a)-(c) it follows for any $i \in I$ that:

$$\{x\} \subseteq \underline{C}_i(X) \text{ and } \{x\} \subseteq \overline{C}_i(X).$$

From this, Propositions 3(a),(c) and Definition 4(b), we obtain the conclusion: $\{x\} \in_\emptyset [X]_i$.

(e) Let $i < \varepsilon$ and $E \in C$. First, let us assume that $E \subseteq \underline{C}_i(X)$. Thus, by Proposition 3(c) and Definition 2(c), we have $\underline{C}_i(E) \subseteq \underline{C}_i(X)$ and $\overline{C}_i(E) \subseteq \overline{C}_i(X)$. Thus, by Definition 4(b), we get: $E \in_i [X]_i$.

On the other hand, by assuming that $E \in_i [X]_i$, from Proposition 3(c) and Definition 4(b), we obtain: $E \subseteq \underline{C}_i(X)$.

(f) Let $E \in C$. If $i = \varepsilon$, then on the basis of Proposition 3(a), we have: $E \subseteq \underline{C}_\varepsilon(X) = \overline{C}_\varepsilon(X) = U$ and from Definition 2(a)-(c) it follows that

$\underline{C}_k(E) \subseteq \underline{C}_\varepsilon(X)$ and $\overline{C}_k(E) \subseteq \overline{C}_\varepsilon(X)$, and from this and Definition 4(a) we get: $E \in_C [X]_i$; thus it follows that the equivalence:

$$E \subseteq \underline{C}_i(X) \Leftrightarrow E \in_C [X]_i$$

is true. In the case, if $i \neq \varepsilon$ this proposition follows easily from Propositions 6(e),(c), which were proved above, and Definition 4.■

Definition 5
In CAS we define the binary relation $\subseteq_C$ as follows:*

$$[X]_i \subseteq_C [Y]_j \Leftrightarrow \underline{C}_i(X) \subseteq \underline{C}_j(Y) \wedge \overline{C}_i(X) \subseteq \overline{C}_j(Y) \wedge i \leq j,$$

for any $X, Y \subseteq U$ and for all $i, j \in I$.
The relation $\subseteq_C$ is called the inclusion of contextual rough sets.

Proposition 7
The inclusion $\subseteq_C$ of contextual rough sets is a partial order in the family of all contextual rough sets of the CAS.*

Proposition 8
In CAS, for any $X, Y \subseteq U$ and all $i, j \in I$ the following relationship holds:*

$$[X]_i = [Y]_j \Leftrightarrow [X]_i \subseteq_C [Y]_j \wedge [Y]_j \subseteq_C [X]_i .$$

From introduced definitions and propositions we obtain:

Theorem 3
In CAS, the counterpart of the axiom of extensionality for the membership relation $\in_C$ holds: for any Y, T, X included in U and for every $i \in I$*

$$(X \in_C [Y]_i \Leftrightarrow X \in_C [T]_i) \Rightarrow [Y]_i = [T]_i).$$

In other words, two contextual rough sets of the degree i containing exactly the same elements are equal.
Proof.
In the proof we use the following lemmas:

Lemma 1
In CAS^*, if $X \subseteq U$ and $i \in I$, then

$$\forall E \in C(E \subseteq BN_i(X) \Rightarrow E \cap X \in_C [X]_i).$$

Lemma 2
In CAS^*, if $X, Y \subseteq U$ and $i \in I$, then

$$\forall E \in C((E \in_C [X]_i \Rightarrow E \in_C [Y]_i) \wedge$$

$$\wedge (E \cap X \in_C [X]_i \Rightarrow E \cap X \in_C [Y]_i)) \Rightarrow [X]_i \subseteq_C [Y]_i .$$

In the proof of Lemma 2 Lemma 1 is used. We see that Theorem 3 holds because, by putting first E for X and then $E \cap Y$ for X in the assumption of this theorem, by using Lemma 2 two times and Proposition 8, we obtain:

$$[Y]_i \subseteq_C [T]_i \text{ and } [T]_i \subseteq_C [Y]_i. \blacksquare$$

By analogy, the following theorem can also be proved:

Theorem 4
In CAS^*, the counterpart of the axiom of extensionality for the membership relations $\in_n$ $(n \in I)$ holds: for any Y, T, X included in U and for every $i \in I$

$$(X \in_i [Y]_i \Leftrightarrow X \in_i [T]_i) \Rightarrow [Y]_i = [T]_i).$$

In other words, two contextual rough sets of the degree i containing exactly the same elements of the degree i $(i \in I)$ are equal.

It seems that theorem 4 constitute a start point for elaborating a set-theoretical base for the approach of rough sets proposed in this paper.

FINAL REMARKS

On the basis of the notions introduced in this paper, we can solve not only the problem of gradation of the membership relation (see [8]), but also formulate the problem of defining and examining operations on contextual rough sets analogously to the classical operations on sets. An attempt at solving this

problem has already been presented in the paper [5]. The general solution of this problem will be the subject of future considerations. In authors' opinion definitions, facts, and theorems given in this paper are theoretical basis for formulating algorythms enabling mechanization of recognizing sets and relations among sets, where knowledge about those sets and relations is incomplete or vague.

ACKNOWLEDGEMENT

The authors want to express warm gratitude to an unknown referee for the remarks which were taken into account while writing the paper. This work was supported by grant No 8S 50302106 from the State Committee for Scientific Research (KBN), Warsaw, Poland.

REFERENCES

[1] Blizard, W. D.,"Multiset Theory." *Notre Dame Journal of Formal Logic*, 1989, vol. 30, no. 1, pp. 36-66.

[2] Blizard, W. D., "Real-Valued Multisets and Fuzzy Sets." *Fuzzy Sets and System*, 1989, no. 33, pp. 77-97.

[3] Bryniarski, E., "A Calculus of Rough Sets of the First Order." *Bulletin of the Polish Academy of Sciences*, 1989: Mathematics, no. 37, pp. 71-78.

[4] Bryniarski, E; Wybraniec–Skardowska, U., "On a Generalization of Rough Sets" *Conference Proceedings, CSC'95 Conference Workshops on Rough Sets and Database Computing*, March 2, California State University, San Jose, 1995, pp. 1–7.

[5] Bryniarski, E; Wybraniec–Skardowska, U., "Calculus of Contextual Rough Sets in Contextual Spaces", *Journal of Applied Non–Classical Logics*, 1996, to appear.

[6] Pawlak, Z., "Rough Sets." *International Journal of Computer and Information Systems*, 1982, no. 11, pp. 672-683.

[7] Pawlak, Z., *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1992.

[8] Pawlak, Z., "Hard and Soft Sets." *Warsaw University of Technology, ICS Research Report*, 1994, no. 10, also in: Proceedings of the International Workshop on Rough Sets and Knowledge Discovery, Banff, Alberta, Canada, October 12-15, 1993, pp. 107-110.

[9] Wybraniec–Skardowska, U., "On a Generalization of Approximation Space." *Bulletin of the Polish Academy of Sciences, Mathematics*, 1989, vol. 37, no 1-6, pp. 51-62.

[10] Wybraniec–Skardowska, U., "Status of Rough Information and the Problem of Vagueness" (in Polish). In: M Omyla (ed.), *Nauka i Jezyk, Biblioteka Mysli Semiotycznej* (J. Pelc, ed.) Warszawa, 1994, pp. 409-431.

[11] Zadeh, L. A., "Fuzzy Sets". *Information and Control*, 1965, vol. 8, pp. 338-353.

[12] Ziarko, W., "Variable Precision Rough Set Model." *Journal of Computer and System Sciences*, 1993, vol. 46, no. 1, pp. 39-59.

18

MAINTENANCE OF REDUCTS IN THE VARIABLE PRECISION ROUGH SET MODEL

Marzena Kryszkiewicz

Institute of Computer Science,
Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: mkr@ii.pw.edu.pl

ABSTRACT

Definitions of a reduct for a single object, decision class and all objects of decision table for the variable precision rough set model are introduced. The definitions have a property that the set of prime implicants of minimal disjunctive normal form of a discernibility function is equal to the set of reducts. Thus the problem of reducts maintenance in dynamically extended information systems is equivalent to the problem of discernibility function maintenance. We prove that the problem can be stated in the form of a Boolean equation: $g \wedge h = f \wedge k$, where f, h and k are given monotonic Boolean functions and g is a function to be determined in minimal disjunctive normal form. An incremental algorithm finding the solution of the above equation is proposed.

1 INTRODUCTION

The rough set model (RS) [1] has been conceived as a tool to conceptualize, organize and analyze various types of data, in particular, to deal with inexact, uncertain or vague knowledge in applications related to Artificial Intelligence.

A basic problem related to practical applications of the rough set based knowledge representation systems (shortly RSKRS) is whether the whole set of attributes is really necessary to represent a given partition of the knowledge, and if not, how to determine the simplified and still sufficient knowledge representation equivalent to the original. Significant results in this area have been achieved in [2], where the methods of reducing the knowledge were studied. The

problem of reducing the RSKRS (known as finding reducts) is transformable to the NP-hard problem of finding minimal disjunctive normal form of a monotonic Boolean function.

The variable precision rough set model (VPRS) [3] is an extension of the rough set model. The model was proposed to analyse and identify data patterns which represent statistical trends rather than functional. The main idea of VPRS model is to allow objects to be classified with an error smaller than some predefined level.

In the paper we introduce definitions of a reduct for a single object, for a decision class and for all objects of a decision table in the VPRS model. The definitions have a property that, like in rough set model, the set of prime implicants of minimal disjunctive normal form (*mdnf*-form) of a discernibility function, which is given in conjunctive normal form (*cnf*-form) initially, is equal to the set of reducts. Thus the problem of reducts maintenance in dynamically extended information systems is equivalent to the problem of discernibility function maintenance. We claim that the problem can be stated in the form of a Boolean equation: $g \wedge h = f \wedge k$, where f, h and k are given monotonic Boolean functions and g is a function to be determined in *mdnf*-form. We propose an incremental algorithm finding the solution of the above equation.

2 BASIC CONCEPTS OF INFORMATION SYSTEMS

Information system is a pair $= (\mathcal{O}, U)$, where U is a non-empty finite set of *attributes* and $\mathcal{O}$ is a non-empty finite set of *objects* described by the set of attributes U. V_a will denote the *domain* of an attribute a. Function $v_a : \mathcal{O} \rightarrow V_a$, assigns a value of an attribute $a \in A$ to every object from $\mathcal{O}$.

Decision table is an information system $S = (\mathcal{O}, C \cup D)$, in which the sets of conditional and decisional attributes C and D respectively, such that $C \cup D = U$ and $C \cup D = \emptyset$, are distinguished.

Each subset of attributes $A \subseteq U$ determines a binary *indiscernibility relation* $IND(A)$, as follows:

$$IND(A) = \{(x, y) \in \mathcal{O} \times \mathcal{O} | \forall a \in A, v_a(x) = v_a(y)\}.$$

The family of all equivalence classes $[x]_A$, $x \in \mathcal{O}$, of the relation $IND(A)$, $A \subseteq U$, constitutes a partition of $\mathcal{O}$, which we will denote by A^*.

In the sequel D_i, $D_i \in D^*$, will denote the set of objects with the decision value i.
$\overline{X}$ will denote the complement of X in $\mathcal{O}$: $\overline{X} = \mathcal{O} \backslash X$.
$\underline{A}X$ is *lower approximation* of X, $X \subseteq \mathcal{O}$, (in RS model) iff

$$\underline{A}X = \bigcup \{E \in A^* | E \subseteq X\} \text{ , where } A \subseteq U.$$

$\underline{A}_\beta X$ is *β-lower approximation* of X, $X \subseteq \mathcal{O}$, (in VPRS model) iff

$$\underline{A}_\beta X = \bigcup \{E \in A^* | card(E \cap \overline{X})/card(E) \leq \beta\}, \text{ where } A \subseteq U \text{ and } 0 \leq \beta < 0.5.$$

The coefficient β determines the admissible degree of classification inaccuracy of X with regard to $IND(A)$. One can easily note that $\underline{A}_\beta X = \underline{A}X$ for $\beta = 0$ and $\underline{A}(\underline{A}_\beta X) = \underline{A}_\beta X$ for $0 \leq \beta < 0.5$.

Property 2.1.

$$\forall D_i, D_j \in D^*, i \neq j, \underline{A}_\beta D_i \cap \underline{A}_\beta D_j = \emptyset; \bigcup_{D_i \in D^*} \underline{A}_\beta D_i \subseteq \mathcal{O}.$$

Property 2.1 is a simple consequence of the assumption that $0 \leq \beta < 0.5$.

Example 2.1. Determine $\underline{C}D_1$ and $\underline{C}_\beta D_1$, $\beta = 1/3$, for the decision table from Table 2.1, where $C = \{a, b, c, d\}$ and $D = \{e\}$.

U	a	b	c	d	e
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	2	1	1
4	1	1	3	1	1
5	2	2	2	2	1
6	1	1	1	1	2
7	2	2	2	1	2
8	1	1	1	2	2
9	1	1	3	1	2

Table 2.1 Decision table.

Solution:

$D_1 = \{1,2,3,4,5\}, \overline{D_1} = \{6,7,8,9\}$ and $C^* = \{E_1, E_2, E_3, E_4, E_5, E_6\}$, where

$E_1 = \{1,2,6\}; \quad E_2 = \{3\}; \quad E_3 = \{4,9\};$
$E_4 = \{5\}; \quad E_5 = \{7\}; \quad E_6 = \{8\};$

$card(E_1 \cap \overline{D_1})/card(E_1) = 1/3 = \beta;$
$card(E_2 \cap \overline{D_1}/card(E_2) = 0/1 < \beta;$
$card(E_3 \cap \overline{D_1}/card(E_3) = 1/2 > \beta;$
$card(E_4 \cap \overline{D_1})/card(E_4) = 0/1 < \beta;$
$card(E_5 \cap \overline{D_1})/card(E_5) = 1/1 > \beta;$
$card(E_6 \cap \overline{D_1})/card(E_6) = 1/1 > \beta;$

Hence:

$\underline{C}D_1 = E_2 \cup E_4 = \{3,5\}$ and $\underline{C}_\beta D_1 = E_1 \cup E_2 \cup E_4 = \{1,2,3,5,6\}$.

□

Example 2.2. Determine $\underline{B}(\underline{C}_\beta D_1), \beta = 1/3$, for the decision table from Table 2.1 if: a) $B = \{a,b,c,d,\}$, b) $B = \{a,b,c\}$, c) $B = \{b,c,d\}$, d) $B = \{b,c\}$, e) $B = \{b,d\}$, f) $B = \{c,d\}$.

Solution:
From Example 2.1 we have: $\underline{C}_\beta D_1 = \{1,2,3,5,6\}$.

a) $B^* = \{\{1,2,6\},\{3\},\{4,9\},\{5\},\{7\},\{8\}\}$;
$\underline{B}(\underline{C}_\beta D_1) = \{1,2,3,5,6\} = \underline{C}_\beta D_1$;

b) $B^* = \{\{1,2,6,8\},\{3\},\{4,9\},\{5,7\}\}$;
$\underline{B}(\underline{C}_\beta D_1) = \{3\} \neq \underline{C}_\beta D_1$;

c) $B^* = \{\{1,2,6\},\{3\},\{4,9\},\{5\},\{7\},\{8\}\}$;
$\underline{B}(\underline{C}_\beta D_1) = \{1,2,3,5,6,\} = \underline{C}_\beta D_1$;

d) $B^* = \{\{1,2,6,8\},\{3\},\{4,9\},\{5,7\}\}$;
$\underline{B}(\underline{C}_\beta D_1) = \{3\} \neq \underline{C}_\beta D_1$;

e) $B^* = \{\{1,2,3,4,6,9\},\{5\},\{7\},\{8\}\}$;
$\underline{B}(\underline{C}_\beta D_1) = \{5\} \neq \underline{C}_\beta D_1$;

f) $B^* = \{\{1,2,6\},\{3,7\},\{4,9\},\{5\},\{8\}\}$;
$\underline{B}(\underline{C}_\beta D_1) = \{1,2,5,6,\} \neq \underline{C}_\beta D_1$.

□

3 REDUCTS OF DECISION TABLE IN VPRS MODEL

Given decision table $S = (\mathcal{O}, C \cup D)$ we are often interested in transfomation of S into a reduced system $S' = (\mathcal{O}, A \cup D)$ such that $A \subseteq C$ and the classification abilities of S' for a given set of objects are the same as those of S. We can say informally that a minimal set of attributes A preserving the classification abilities is called a *reduct.* Here we introduce definitions of reducts in VPRS model that preserve classification abilities of the original decision table for all objects from $POS_{\underline{\beta}}(C) = \bigcup_{D_1 \in D^*} \underline{C}_\beta D_i$. (Extended classification of reducts in RS and VPRS models can be found in [4]).

A $\underline{\beta}$ *-reduct for a decision table* we call any subset B, $B \subseteq C$, such that:

$$\forall D_i \in D^*, \underline{B}(\underline{C}_\beta D_i) = \underline{C}_\beta D_i \text{ and } \forall A \subset B, \exists D_i \in D^*, \underline{A}(\underline{C}_\beta D_i) \neq \underline{C}_\beta D_i.$$

A $\underline{\beta}$ *-reduct for a decision class* D_i we call any subset B, $B \subseteq C$, such that:

$$\underline{B}(\underline{C}_\beta D_i) = \underline{C}_\beta D_i \text{ and } \forall A \subset B, \underline{A}(\underline{C}_\beta D_i) \neq \underline{C}_\beta D_i.$$

A $\underline{\beta}$ *-reduct for an object* x we call any subset B, $B \subseteq C$, such that:

$$x \in \underline{B}(\underline{C}_\beta D_i) \Leftrightarrow x \in \underline{C}_\beta D_i \text{ and } \forall A \subset B, x \notin \underline{A}(\underline{C}_\beta D_i).$$

From Example 2.2 we can conclude that $\{b,c,d\}$ is a $\underline{\beta}$ -reduct for $D_1 = \{1,2,3,4,5\}$ in the decision table presented in Table 2.1.

We have assumed that $\underline{\beta}$ -reducts preserve classification abilities of the original decision table for all objects from $POS_{\underline{\beta}}(C)$. However, the formal trial to compute $\underline{\beta}$-reduct for an object from $\mathcal{O} \backslash POS_{\underline{\beta}}(C)$ by the definition above returns $\emptyset$ as the only $\underline{\beta}$-reduct.

4 DISCERNIBILITY FUNCTIONS

It was proved in [2], [5] that the problem of computing reducts in RS model is transformable to the problem of finding prime implicants of a monotonic Boolean function called *discernibility function.* Following this approach to determination of reducts we define monotonic Boolean functions of this property for $\underline{\beta}$-reducts.

Let $\sum \delta_A(x,y)$ be a Boolean expression which is equal to 1, if $v_a(x) = v_a(y)$ for each $a \in A$. Otherwise, let $\sum \delta_A(x,y)$ be a disjunction of variables corresponding to attributes $a \in A$, such that $v_a(x) \neq v_a(y)$.

$\Delta_{\underline{\beta}}$ is a *discernibility function for a decision table* S iff

$$\Delta_{\underline{\beta}} = \prod_{(x,y)\in \underline{C}_\beta D_i \times \overline{\underline{C}_\beta D_i}} \sum \delta_C(x,y) \text{ for all } D_i \in D^*.$$

$\Delta_{\underline{\beta}}(D_i)$ is a *discernibility function for a decision class* D_i in S iff

$$\Delta_{\underline{\beta}}(D_i) = \prod_{(x,y)\in \underline{C}_\beta D_i \times \overline{\underline{C}_\beta D_i}} \sum \delta_C(x,y).$$

$\Delta_{\underline{\beta}}(x)$ is a *discernibility function for an object* x in S iff

$$\text{If } x \in \underline{C}_\beta D_i, i = v_D(x), \text{ then } \Delta_{\underline{\beta}}(x) = \prod_{y\in \overline{\underline{C}_\beta D_i}} \delta_C(x,y) \text{ else } \Delta_{\underline{\beta}}(x) = 1.$$

Property 4.1. (Composition of discernibility functions)

$$\Delta_{\underline{\beta}} = \prod_{D_i \in D^*} \Delta_{\underline{\beta}}(D_i); \Delta_{\underline{\beta}}(D_i) = \prod_{x\in \underline{C}_\beta D_i} \Delta_{\underline{\beta}}(x).$$

Example 4.1. Determine all types of $\underline{\beta}$-reducts for decision table described by Table 2.1.

Solution:
First we construct Table 4.1 (discernibility matrix) in which we place values $\delta_C(x,y)$ for all pairs $(x,y) \in \mathcal{O} \times \mathcal{O}$. Then we construct appropriate discernibility functions and determine their implicants. We use Property 4.1 to simplify computations.

$x\backslash y$	1	2	3	4	5	6	7	8	9
1			c	c	$abcd$		abc	d	c
2			c	c	$abcd$		abc	d	c
3	c	c		c	abd	c	ab	cd	c
4	c	c	c		$abcd$	c	d	cd	
5	$abcd$	$abcd$	abd	$abcd$		$abcd$	d	abc	$abcd$
6			c	c	$abcd$		abc	d	c
7	abc	abc	ab	d	d	abc		$abcd$	abc
8	d	d	cd	cd	abc	d	$abcd$		cd
9	c	c	c		$abcd$	c	abc	cd	

Table 4.1 Discernibility matrix.

From Table 2.1 we have:

$D^* = \{D_1, D_2\}$, where $D_1 = \{1,2,3,4,5\}$ and $D_2 = \{6,7,8,9\}$;

$C^* = \{\{1,2,6\},\{3\},\{4,9\},\{5\},\{7\},\{8\}\}$.

Hence,

$\underline{C}_\beta D_1 = \{1,2,3,5,6\}$; $\overline{\underline{C}_\beta D_1} = \{4,7,8,9\}$;

$\underline{C}_\beta D_2 = \{7,8\}$; $\overline{\underline{C}_\beta D_2} = \{1,2,3,4,5,6,9\}$;

$\Delta_{\underline{\beta}}(1) = \Delta_{\underline{\beta}}(2) = \Delta_{\underline{\beta}}(6) = c(a \vee b \vee c)d = cd$;

$\Delta_{\underline{\beta}}(3) = c(a \vee b)(c \vee d) = c(a \vee b) = ac \vee bc$;

$\Delta_{\underline{\beta}}(4) = \Delta_{\underline{\beta}}(9) = 1$;

$\Delta_{\underline{\beta}}(5) = (a \vee b \vee c \vee d)d(a \vee b \vee c) = d(a \vee b \vee c) = ad \vee bd \vee cd$;

$\Delta_{\underline{\beta}}(7) = (a \vee b \vee c)(a \vee b)d = (a \vee b)d = ad \vee bd$;

$\Delta_{\underline{\beta}}(8) = d(c \vee d)(a \vee b \vee c) = d(a \vee b \vee c) = ad \vee bd \vee cd$;

$\Delta_{\underline{\beta}}(D_1) = \Delta_{\underline{\beta}}(1) \wedge \Delta_{\underline{\beta}}(2) \wedge \Delta_{\underline{\beta}}(3) \wedge \Delta_{\underline{\beta}}(5) \wedge \Delta_{\underline{\beta}}(6) = acd \vee bcd$;

$\Delta_{\underline{\beta}}(D_2) = \Delta_{\underline{\beta}}(7) \wedge \Delta_{\underline{\beta}}(8) = ad \vee bd$;

$\Delta_{\underline{\beta}} = \Delta_{\underline{\beta}}(D_1) \wedge \Delta_{\underline{\beta}}(D_2) = acd \vee bcd$.

□

5 INCREMENTAL COMPUTATION OF $\underline{\beta}$-REDUCTS

Let us consider two decision tables: $S = (\mathcal{O}, C \cup D)$ and $S' = (\mathcal{O}', C \cup D)$, $\mathcal{O}' = \mathcal{O} \cup \{z\}$. Our task is to compute $\underline{\beta}$-reducts of S' (i.e. prime implicants of discernibility functions of S') using the knowledge about $\underline{\beta}$-reducts of S (i.e. prime implicants of discernibility functions of S).

We restrict our considerations to determining $\underline{\beta}$-reducts for objects. $\underline{\beta}$-reducts for decision classes and for the whole decision table may be computed in analogous way with the use of Property 4.1 (of composing discernibility functions).

In the sequel k will stand for the decision value of a new object z (i.e. $k = v_D(z)$) and all notations referring to S' will be followed by the prime sign (').

The essential property, we shall exploit to work out an inceremental way of computing $\underline{\beta}$-reducts, is that for each family $E \in C^*$ there is at most one $\underline{\beta}$-lower approximation $\underline{C}_\beta D_i$ $(i \in V_D)$ such that $E \subseteq \underline{C}_\beta D_i$. We will specify only those discernibility functions for S' which differ from the corresponding functions in S. To simplify the problem of incremental computation of $\underline{\beta}$-reducts we will analyse two simpler subcases:

1. $card([z]'_c) = 1$:

$$\Delta'_{\underline{\beta}}(z) = \prod_{y \in \underline{C}_\beta D_k} \sum \delta_C(z, y);$$

$$\Delta'_{\underline{\beta}}(x) = \Delta_{\underline{\beta}}(x) \wedge \sum \delta_C(x, z) \text{ for } x \in POS_{\underline{\beta}}(C) \backslash \underline{C}_\beta D_k.$$

2. $card([z]'_C) > 1$:

a) If there is no $\underline{C}_\beta D_i$ such that $(\mathcal{O} \cap [z]'_C) \subseteq \underline{C}_\beta D_i$ and $z \notin \underline{C}_\beta D'_k$ then

$$\Delta'_{\underline{\beta}}(z) = 1;$$

b) If there is $\underline{C}_\beta D_i$ such that $(\mathcal{O} \cap [z]'_C) \subseteq \underline{C}_\beta D_i$ and $z \in \underline{C}_\beta D'_i$ then

$$\Delta'_{\underline{\beta}}(z) = \Delta_{\underline{\beta}}(x), \text{ where } x \text{ is an object from } [z]'_C;$$

c) If there is $\underline{C}_\beta D_i$ such that $(\mathcal{O} \cap [z]'_C) \subseteq \underline{C}_\beta D_i$ and $z \notin \underline{C}_\beta D'_i$ and $z \notin \underline{C}_\beta D'_k$ then

$$\Delta'_{\underline{\beta}}(x) = \Delta_{\underline{\beta}}(x) \wedge \sum \delta_C(x,z) \text{ for } x \in \underline{C}_\beta D'_i;$$

$$\Delta'_{\underline{\beta}}(z) = \Delta'_{\underline{\beta}}(x) = 1 \text{ for } x \in [z]'_C;$$

d) If there is $\underline{C}_\beta D_i$ such that $(\mathcal{O} \cap [z]'_C) \subseteq \underline{C}_\beta D_i$ and $z \notin \underline{C}_\beta D'_i$ and $z \in \underline{C}_\beta D'_k$ then $\neg((\mathcal{O} \cap [z]'_C) \subseteq \underline{C}_\beta D_k), [z]'_C \subseteq \underline{C}_\beta D'_k, \neg([z]'_C \subseteq \underline{C}_\beta D'_i), \underline{C}_\beta D'_i = \underline{C}_\beta D_i \backslash [z]'_C$. Hence:

$$\Delta'_{\underline{\beta}}(x) = \Delta_{\underline{\beta}}(x) \wedge \sum \delta_C(x,z) \text{ for } x \in \underline{C}_\beta D'_i;$$

$$\Delta'_{\underline{\beta}}(x) \wedge \prod_{y \in \underline{C}_\beta D_k} \sum \delta_C(x,y) = \Delta_{\underline{\beta}}(x) \wedge \prod_{y \in \underline{C}_\beta D'_i} \sum \delta_C(x,y) \text{ for } x \in [z]'_C; \quad (5.1)$$

$$\Delta'_{\underline{\beta}}(z) = \Delta_{\underline{\beta}}(x), \text{ where } x \text{ is an object from } [z]'_C;$$

$$\Delta'_{\underline{\beta}}(x) \wedge \sum \delta_C(x,z) = \Delta_{\underline{\beta}}(x) \text{ for } x \in \underline{C}_\beta D_k; \quad (5.2)$$

e) If there is no $\underline{C}_\beta D_i$ such that $(\mathcal{O} \cap [z]'_C) \subseteq \underline{C}_\beta D_i$ and $z \in \underline{C}_\beta D'_k$ then

$$\Delta'_{\underline{\beta}}(z) = \prod_{y \in \overline{\underline{C}_\beta D_k}} \sum \delta_C(x,y);$$

$$\Delta'_{\underline{\beta}}(x) = \Delta'_{\underline{\beta}}(z) \text{ for } x \in [z]'_C;$$

$\Delta'_{\underline{\beta}}(x)$ is defined by Eq. (5.2) for $x \in \underline{C}_\beta D_k$.

□

Equations (5.1-5.2) do not determine $\Delta'_{\underline{\beta}}(x)$ uniquely. Beneath we present equations that express the dependency between $\Delta'_{\underline{\beta}}$ and $\Delta_{\underline{\beta}}(x)$ in a unique way. To this end, we need to define the operator $CABSREL(C_1, C_2)$. Its arguments are Boolean expressions in *cnf*-form. $CABSREL$ returns *cnf*-expression constructed from those implicates from C_1 which are not absorbable by any implicate from C_2.

Ad. Eq. (5.1) For $x \in (\mathcal{O} \cap [z]'_C)$

$$\begin{aligned}\Delta_{\underline{\beta}}(x) &= \prod_{y \in \overline{\underline{C}_\beta D_i}} \sum \delta_C(x,y) = \\ &= \prod_{y \in \overline{\underline{C}_\beta D_i} \setminus \underline{C}_\beta D_k} \sum \delta_C(x,y) \wedge \prod_{y \in \underline{C}_\beta D_k} \sum \delta_C(x,y);\end{aligned}$$

Hence Eq. (5.1) for $x \in [z]'_C$ can be rewritten as follows:

$$\Delta'_{\underline{\beta}}(x) = \prod_{y \in \overline{\underline{C}_\beta D_i} \setminus \underline{C}_\beta D_k} \sum \delta_C(x,y) \wedge \prod_{y \in \underline{C}_\beta D'_i} \sum \delta_C(x,y) \qquad (5.3)$$

where *mdnf*-form of the expression $\prod_{y \in \overline{\underline{C}_\beta D_i} \setminus \underline{C}_\beta D_k} \sum \delta_C(x,y)$ can be computed from the equation below:

$$\begin{aligned}MDNF(\Delta_{\underline{\beta}}(x)) &= MDNF\left(\prod_{y \in \overline{\underline{C}_\beta D_i} \setminus \underline{C}_\beta D_k} \sum \delta_C(x,y) \wedge \right. \\ &\wedge \; CABSREL\left(\prod_{y \in \underline{C}_\beta D_k} \sum \delta_c(x,y),\right. \\ &\qquad \left.\left. \prod_{y \in \overline{\underline{C}_\beta D_i} \setminus \underline{C}_\beta D_k} \sum \delta_c(x,y)\right)\right) \qquad (5.4)\end{aligned}$$

□

Ad. Eq. (5.2)

For $x \in \underline{C}_\beta D_k$

$$\begin{aligned}\Delta_{\underline{\beta}}(x) &= \prod_{y \in \overline{\underline{C}_\beta D_k}} \sum \delta_c(x,y) = \\ &= \prod_{y \in \overline{\underline{C}_\beta D_k} \setminus [z]'_C} \sum \delta_C(x,y,) \wedge \prod_{y \in [z]'_C} \sum \delta_C(x,y) = \\ &= \prod_{y \in \overline{\underline{C}_\beta D_k} \setminus [z]'_C} \sum \delta_C(x,y) \wedge \sum \delta_C(x,z);\end{aligned}$$

Hence Eq. (5.2) for $x \in \underline{C}_\beta D_k$ can be rewritten in the form:

$$\Delta'_{\underline{\beta}}(x) = \prod_{y \in \overline{\underline{C}_\beta D_k} \setminus [z]'_C} \sum \delta_C(x,y) \tag{5.5}$$

where *mdnf*-form of the expression $\prod_{y \in \overline{\underline{C}_\beta D_k} \setminus [z]'_C} \sum \delta_C(x,y)$ is defined by the equation:

$$\begin{aligned} MDNF(\Delta_{\underline{\beta}}(x)) \;=\; & MDNF\left(\prod_{y \in \overline{\underline{C}_\beta D_k} \setminus [z]'_C} \sum \delta_C(x,y) \wedge \right. \\ \wedge \; & CABSREL\left(\sum \delta_C(x,z), \right. \\ & \left. \left. \prod_{y \in \overline{\underline{C}_\beta D_k} \setminus [z]'_C} \sum \delta_C(x,y) \right) \right) \end{aligned} \tag{5.6}$$

□

6 FINDING PRIME IMPLICANTS OF BOOLEAN SUBFUNCTION

In the previous section we encountered the problem of determining *mdnf*-form of a Boolean subfunction (see Eq. 5.4, 5.6). Formally, the problem considered is as follows:

Find prime implicants of function g such that:

$$f = g \wedge h (6.1)$$

and

6.2 *mdnf*-form of function f (discernibility function) and *cnf*-forms of functions g and h are given.[1]

[1]It is possible to construct algorithms which determine prime implicants of g when only *mdnf*-form of f and *cnf*-form of h are given [6]. However, the time complexity of these algoritms is at least n-times, where n is a sum of the numbers of variables occuring in all implicates of $MCNF(h)$, greater than the complexity of the algorithm to be presented.

6.3 Implicates of *mcnf*-form of functions h are not absorbable by implicates occuring in *cnf*-form of function g and implicates of *mcnf*-form of function g are not absorbable by implicates occuring in *cnf*-form of function h.

Given *cnf*-forms of functions g and h we can obtain their *mcnf*-forms applying the absorption law. In the sequel, we assume that *mcnf*-forms of functions g and h are known.

The problem of determining subfunction g from the equation above may be solved iteratively. Each time *mdnf*- and *mcnf*-form of function g_i, $i = 0 \ldots n$, where n is the number of implicates in $MCNF(h)$, such that:

$$\begin{cases} g_0 = f \\ g_{i-1} = g_i \wedge h_i \quad \text{for } i = 1 \ldots n, \end{cases}$$

where h_i is the i-th implicate of $MCNF(h)$,

should be determined. (*mcnf*-form of function g_i is being obtained in a very simple way - by deleting h_i from $MCNF(g_{i-1})$.)

As $f = g_0 = h_1 \wedge g_1 = h_1 \wedge h_2 \wedge g_2 = h_1 \wedge h_2 \wedge \ldots \wedge h_n \wedge g_n$ and $f = g \wedge h = h \wedge g$ thus $g = g_n$.

So we have simplified the initial task of finding the *mdnf*-form of the function g_i. Let us formulate this problem as follows:

Given single implicate expression h, *mdnf*-form of function f and *mcnf*-form of function g, determine *mdnf*-form of subfunction g satisfying Eq. (6.1) and Condition (6.3).

Property 6.1.

Each implicant of $MDNF(f)$ contains at least one variable from h.

Proof: Property 6.1 is a simple consequence of the fact that each prime implicant of a monotonic Boolean function has at least one common variable with any implicate in a *cnf*-form of this function. This fact was proved in [2].

□

Property 6.2.

If a is a variable occuring both in implicate h and in some implicant f_i of $MDNF(f)$, then
a) f_i is a prime implicant of g, if there exists an implicate in $MCNF(g)$ that contains a,
b) f_i/a is a prime implicant of g, otherwise.

Proof: A variable appears in an implicate of *mcnf*-form of a Boolean function iff there is an implicant which contains this variable in *mdnf*-form of this function.
Ad. a) If there is an implicate in $MCNF(g)$ that contains a then $MDNF(g)$ will contain prime implicants with this variable. Let us denote this set of implicants by $G(a)$. If h also contains a then the set of prime implicants in $MDNF(f)$ which contain a will be equal to $G(a)$.
Ad. b) It is trivial that f_i/a is a prime implicant of g if f_i is a prime implicant of g and a does not appear in h.

□

Property 6.3.

If a is a variable occuring both in implicate h and in an implicant f_i of $MDNF(f)$, and no implicate in $MCNF(g)$ contains a, then f_i does not contain any of the remaining variables from h.

Proof: By contradiction: Let a appears both in implicate h and in an implicant f_i of $MDNF(f)$, but it does not appear in any implicate of $MCNF(g)$. Let f_i contains also variable b from h, where b is a different variable from a. From Property 6.2 b) we have that f_i/a is a prime implicant of g. We can also conclude that f_i/a contains b since f_i contains b. This means that $MCNF(g)$

contains b. However, applying Property 6.2 a) we induce that f_i/a is a prime implicant of f, which leads to contradiction with the initial assumption.

□

Properties 6.1-6.3 allow us to construct Algorithm 6.1, which solves the simplified problem:

Algorithm 6.1

begin
  $MDNF\_G = 0$;
  determine set A containing those variables from H
  which occur also in $MCNF\_G$;
  determine set $\overline{A}$ containing those variables from H
  which does not belong to A;
  forall implicant Y **in** $MDNF\_F$; /* Loop 6.1 */
    if there is a variable from A in Y **then**
      $MDNF\_G = MDNF\_G \vee Y$; /* Property 6.2.a */
      delete Y from $MDNF\_F$;
    endif;
  endfor;
  forall implicant Y **in** $MDNF\_F$; /* Loop 6.2 */
    delete variables $\overline{A}$ from Y;
    /* there will be deleted only one variable from Y
      according to Property 6.3 */
    $MDNF\_G = MDNF\_G \vee Y$; /* Property 6.2.b */
    delete implicants absorbable by Y from $MDNF\_F$;
  endfor;
  return $MDNF\_G$;
end;

Algorithm 6.1 has three arguments $MDNF\_F, MCNF\_G$ and H representing *mdnf*-form of function f, *mcnf*-form of function g and implicate h respectively. The resulting *mdnf*-form of g is returned by variable $MDNF\_G$. Initially, $MDNF\_G$ does not contain any implicant.

Set A is constructed from those variables from H which occur also in $MCNF\_G$. The time complexity of this operation is $O(n)$, where n is the number of all implicates in $MCNF\_G$. Set $\overline{A}$ will contain those variables from H which do not belong to A. This operation is performed in one step.

According to Property 6.2.a, implicants from $MDNF\_F$ that contain some variables from A should belong to $MDNF\_G$. Thus they are being deleted from $MDNF\_F$ and placed in $MDNF\_G$. The time complexity of this operation (Loop 6.1) is $O(m)$, where m is the number of all implicants in $MDNF\_F$. Having finished Loop 6.1, $MDNF\_F$ does not have implicants containing variables from A.

In accordance with Property 6.2.b, $MDNF\_G$ should contain also all those implicants Y/a, such that Y is an implicant of $MDNF\_F$ and $a \in \overline{A}$. As a result of deleting variables $\overline{A}$ from implicants of $MDNF\_F$ before placing them in $MDNF\_G$, there may be created implicants which already exist in $MDNF\_G$. To avoid this, there are being deleted all such implicants from $MDNF\_F$ that are absorbable by consequtively determined implicants Y/a. Let l means the number of implicants in $MDNF\_F$ before Loop 6.2 has been performed. The pessimistic time complexity of the absorption operation is $O(l^2)$.[2]

Example 6.1. Illustrate the execution of Algorithm 6.1 for $MDNF\_F = ac \vee bcd \vee cde$, $MCNF\_G = c(a \vee d)$ and $H = a \vee b \vee e$.

Solution:
First, we determine sets A and $\overline{A}$: $A = \{a\}$, $\overline{A} = \{b, e\}$. Next we perform Loop 6.1. As a result we achieve: $MDNF\_G = ac$ and $MDNF\_F = bcd \vee cde$. While perfoming Loop 6.2 we add implicant cd to $MDNF\_G$, thus obtaining the final form of $MDNF\_G$: $MDNF\_G = ac \vee cd$.

□

Among the operations performed to determine prime implicants of subfunction g the most time consuming ones are: computation of *mcnf*-forms of functions g and h from their *cnf*-forms, and testing, whether consequtive implicants being created from $MDNF\_F$ in Loop 6.2 of Algorithm 6.1 exist already in $MDNF\_G$. Pessimistic time complexity of these operations is $O(l^2)$, where l

[2]If $card(\overline{A}) = 1$, then consequtively determined implicants Y/a are unique. Thus, there is no need to perform absorption operation.

is either the number of implicates of $MCNF(g)$ or $MCNF(h)$ respectively or the number of implicants in $MDNF\_F$. Using tree structures described in [7], it is possible to lower the time complexity to $O(l \times n)$, where n is the number of all different variables occuring in respective *mcnf*-, *mdnf*-Boolean expressions.

Let us note that when maintaining reducts in dynamic environment, one may keep the information only on a few the most interesting prime implicants of a discernibility function. This may considerably speed up the process of reducts reparation.

7 CONCLUSIONS

The problem of reducts maintenance in VPRS is equivalent to the problem of discernibility functions maintenance. If a new object creates a new class with regard to classifying attributes than a new discernibility function is created only for this object and the remaining discernibility functions are modified in a strightforward way by multiplying the old *mdnf*-form by an disjunction of literals. Otherwise, we define the problem in the form of a Boolean equation: $\Delta'_{\underline{\beta}} \wedge h = \Delta_{\underline{\beta}} \wedge k$, where $\Delta'_{\underline{\beta}}$- is a discernibility function of an extension S' of an information system S, to be determined in *mdnf*-form, $\Delta_{\underline{\beta}}$- is in *mdnf*-form, which represents $\underline{\beta}$-reducts in S, h and k are Boolean expressions in *cnf*-form. Occurrence of expressions h and k in the equation is caused by changes in at most two β-lower approximations after S has been extended. Modification of reducts is limited to computations only for the objects belonging to these changed β-lower approximations. We also offered a general method of how to modify these reducts.

Problem of reducts (as defined in [2]) maintenance in RS, as a special case of reducts maintenance in VPRS, may be formulated in a simpler form. Reducts of extended system can be determined from the equation: $\Delta'_{\underline{\beta}} = \Delta_{\underline{\beta}} \wedge k$ [8], [?]. On the other hand, it can be proved that the equation $\Delta'_{\underline{\beta}} \wedge h = \Delta_{\underline{\beta}} \wedge k$ is applicable in case of VPRS with Asymmetric Bounds [10], which is a generalization of VPRS and for types of reducts proposed in [4], [11].

Acknowledgements

This work has been supported by grant: No. 8 T11C 038 08 from the State Committee for Scientific Research I am grateful to Dr. Wojciech Ziarko and Dr. Jack David Katzberg for inspiration. Many thanks are owned to Dr. Henryk Rybinski for several helpful comments.

REFERENCES

[1] Pawlak, Z., *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publisher, 1991.

[2] Skowron, A., Rauszer, C., "The Discernibility Matrices and Functions in Information Systems". In *Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory*, R. Slowinski, ed., Kluwer Academic Publishers, pp. 331-362, 1992.

[3] Ziarko, W., "Analysis of Uncertain Information in the Framework of Variable Precision Rough Sets", *Foundations of Computing and Decision Sciences*, Vol. 18, No. 3-4, pp. 381-396, 1993.

[4] Kryszkiewicz, M., *Knowledge Reduction in Information Systems*, Ph.D. Thesis, Warsaw University of Technology, 1994.

[5] Pawlak, Z., Skowron, A., "A Rough Set Approach to Decision Rules Generation", ICS Research Report 23/93, Warsaw University of Technology, 1993.

[6] Kryszkiewicz, M., "Maintenance of Reducts in the Variable Precision Rough Sets Model", ICS Research Report 31/94, Warsaw University of Technology, June 1994.

[7] de Kleer, J., "An Improved Incremental Algorithm for Generating Prime Implicates", *Representation and Reasoning: Tractability, AAAI*, pp. 780-785, 1992.

[8] Orlowska, E., Orlowski, M., "Maintenance of Knowledge in Dynamic Information Systems". In *Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory*, R. Slowinski, ed., Kluwer Academic Publishers, pp. 315-330, 1992.

[9] Shan, N., Ziarko, W., "An Incremental Learning Algorithm for Constructing Decision Rules". In *Rough Sets, Fuzzy Sets and Knowledge Discovery (RSKD '93)*, W. Ziarko, ed., Springer-Verlag, pp. 326-334, 1994.

[10] Katzberg, J.D., Ziarko, W., "Variable Precision Rough Sets with Asymmetric Bounds". In *Rough Sets, Fuzzy Sets and Knowledge Discovery (RSKD '93)*, W. Ziarko, ed., Springer-Verlag, pp. 167-177, 1994.

[11] Skowron, A., Stepaniuk, J., "Intelligent Systems Based on Rough Set Approach", *Foundations of Computing and Decision Sciences*, Vol. 18, No. 3-4, pp. 343-360, 1993.

19

PROBABILISTIC ROUGH CLASSIFIERS WITH MIXTURES OF DISCRETE AND CONTINUOUS ATTRIBUTES

Andrzej Lenarcik and Zdzisław Piasta

Mathematics Department,
Kielce University of Technology,
Al. 1000-lecia, 25-314 Kielce, Poland,
e-mail: {ztpal,matzp}@eden.tu.kielce.pl

ABSTRACT

A procedure for construction of a rule inductive classifier is presented. A form of the classifier corresponds to the decision algorithms originated from the concepts of the rough set theory. The kernel-based and frequency-based estimators are used to approximate the probabilistic structure of the data. The procedure directly incorporates attributes of any mixed type. It also accepts missing values in the data. A minimization of misclassification cost is used as the criterion of classifier generation. The procedure is illustrated on two datasets from the credit assessment domain.

1 INTRODUCTION

The rough set theory of Z.Pawlak [1] [2] inspired mamy of researchers to develope learning algorithms appropriated for classifying new objects (see papers in [3] [4] [5], for example). However, since the problem of classification exceeds the basic concepts of the rough set theory, the algorithms incorporate same statistical techniques, such as k-fold cross-validation or bootstrapping at the stage of verification. The procedure for classifier construction presented in this paper refers to statistical methods also at the stage of generation. The procedure directly incorporates the arbitrary mixtures of continuous and discrete condition attributes, and accepts missing values in the data. The resulting decision algorithm may be used not only to classify new objects but also to explain hidden relationships between the attributes and decisions.

The classifier construction is based on the information about the finite set $U = \{u_1, \ldots, u_n\}$ of learning objects. The set U is also called a learning sample. The information is expressed in terms of condition attributes $a_1, \ldots, a_m$ and one decision attribute d. We denote by $V^{(q)}$ the set of values for the attribute a_q $(q = 1, \ldots, m)$, and by V_d the finite set of values (decisions) for the decision attribute. The set $V^{(q)}$ is finite for the discrete a_q attribute and it is an interval of real numbers $\mathbf{R}$ when the a_q attribute is continuous. If the set $V^{(q)}$ is treated as ordered then the attribute a_q is ordinal. The condition attribute space is the product $\mathcal{V} = V^{(1)} \times \ldots \times V^{(m)}$. By a feasible subset in V_q we mean the set $\Delta^{(q)} \subset V_q$ which is an interval, when the attribute a_q is ordinal, or an arbitrary subset, otherwise. The set $\Delta \subset \mathcal{V}$ we call a feasible subset in $\mathcal{V}$ if it has the form $\Delta = \Delta^{(1)} \times \ldots \times \Delta^{(m)}$, where $\Delta^{(q)}$ are the feasible subsets in V_q for $q = 1, \ldots, m$. By the rough classifier [8] in $\mathcal{V}$ we mean a triple $\mathcal{K} = (\mathcal{S}, V_d, \kappa)$, where $\mathcal{S} = \{\Delta_1, \ldots, \Delta_k\}$ is the partition of $\mathcal{V}$ into the feasible subsets, V_d is the set of decisions, and $\kappa : \mathcal{S} \dashrightarrow V_d$ is a partial mapping which assigns a particular decision to each element of the partition. The mapping κ is called a classification algorithm or a decision algorithm of the rough classifier. The algorithm consists of the following rules

$$a^{(1)} \in \Delta_i^{(1)} \textbf{ and } a^{(2)} \in \Delta_i^{(2)} \textbf{ and } \ldots \textbf{ and } a^{(m)} \in \Delta_i^{(m)} \textbf{ then } d = \kappa(\Delta_i)$$

The rough classifier is complete when $\mathrm{Dom}(\kappa) = \mathcal{S}$.

2 OPTIMIZATION CRITERION

The classifier construction is based on the minimization of misclassification cost of unseen objects. In many practical situations it is more costly to make one kind of classification error than the other. For example, it is rather more costly to determine that a credit applicant will be paying debts back when in reality he is a defaulter, than to establish that a client with a good credit record will default on payments.

Without a loss of generality we can state that $V_d = \{1, 2, \ldots, l\}$. Let $\mathcal{S} = \{\Delta_1, \ldots, \Delta_k\}$ be the partition of the condition attribute space into the feasible subsets. By p_{ij} we denote the probability of an object with the j–th decision occurring in the region Δ_i $(i = 1, \ldots, k;\ j = 1, \ldots, l)$. The probability that the new object is associated with the j–th decision we denote by π_j. Clearly, $\pi_j = p_{1j} + p_{2j} + \ldots + p_{kj}$ $(j = 1, \ldots, l)$.

Let the unit cost of misclassifying the object with the j-th decision as the object with the j' -th decision be equal $C(j \to j')$ $(j \neq j')$. In the case of a correct

classification the negative costs $C(j \to j)$ can be used. The minimization of the error rate can be obtained as a particular case of the misclassification cost minimization with $C(j \to j') = 1$, if $j \neq j'$, and $C(j \to j') = 0$, otherwise.

If new objects occuring in the region Δ_i will be related all the time with the same decision $j \in V_d$, then the mean unit cost will be equal

$$C(\Delta_i \to j) = C(1 \to j)p_{i1} + C(2 \to j)p_{i2} + \ldots + C(l \to j)p_{il} \,. \tag{1.1}$$

The optimal decision $j \in V_d$ minimizes the cost (1.1). The minimal value of the cost associated with the region Δ_i is equal

$$C(\Delta_i) = \min_{j=1,\ldots,l} C(\Delta_i \to j) \,. \tag{1.2}$$

If $C(\Delta_i \to j) = C(\Delta_i)$ then the decision $j \in V_d$ is called permissible for Δ_i. If $\kappa(\Delta_i) \in V_d$ is the permissible decision for each element of the partition $\mathcal{S} = \{\Delta_1, \ldots, \Delta_k\}$ then the complete classifier $\mathcal{K} = (\mathcal{S}, V_d, \kappa)$ is called permissible for $\mathcal{S}$.

The minimal value of the cost associated with the partition $S = \{\Delta_1, \ldots, \Delta_k\}$ is given by

$$C(\mathcal{S}) = C(\Delta_1) + \ldots + C(\Delta_k) \,. \tag{1.3}$$

3 ESTIMATION OF PROBABILITIES

In practice, the probabilities p_{ij} defined in the previous section are unknown and should be estimated. The simplest estimator of p_{ij} is of the form

$$\hat{p}_{ij} = \frac{n_{ij}}{N} \,, \tag{1.4}$$

where n_{ij} is the number of objects with the j–th decision in the region Δ_i, and N is the number of all learning objects. Using this estimator makes sense when the frequencies with which the learning objects with different decisions occur are close to the same frequencies for unseen objects. If this condition is not satisfied then the prior probabilities π_j of each decision have to be given. In this case the estimator of p_{ij} is of the form

$$\hat{p}_{ij} = \pi_j \frac{n_{ij}}{n_j} \,, \tag{1.5}$$

where n_j is the number of learning objects with the j–th decision ($j = 1, \ldots, l$).

The alternative procedure of estimating the probabilities p_{ij} can be based on the estimation of unknown probability density functions in the space $\mathcal{V}$. Let us denote by $f_j : \mathcal{V} \to \mathbf{R}$ the probability density functions for the j–th decision in the space $\mathcal{V}$. Then

$$p_{ij} = \pi_j \int_{\Delta_i} f_j(x)dx \ , \tag{1.6}$$

where dx is the product of counting and Lebesgue measures taken in the same order as the order of discrete and continuous attributes in the product $\mathcal{V} = V^{(1)} \times \ldots \times V^{(m)}$. It is known [6] that the optimal decision for a new object occuring at $x \in \mathcal{V}$ is equal to the value $j \in V_d$ which minimizes the

$$C(1 \to j)f_1(x)\pi_1 + \ldots + C(l \to j)f_l(x)\pi_l \ . \tag{1.7}$$

In practice, the densities $f_j(x)$ have to be replaced by their estimators $\hat{f}_j(x)$.

The values of the attributes $a_1, \ldots, a_m, d$ for $u_\nu \in U$ we denote by (x_ν, d_ν) where $x_\nu = (x_\nu^{(1)}, \ldots, x_\nu^{(m)})$. The kernel method [7] is used to estimate the densities $f_j(x)$. The estimators of $f_j(x)$ are given by

$$\hat{f}_j(x) = \frac{1}{n_j} \sum_{\substack{\nu = 1 \\ d_\nu = j}}^{n} \prod_{q=1}^{m} K_\nu^{(q)}(x^{(q)}, h_j^{(q)}) \ , \tag{1.8}$$

where $K_\nu^{(q)}(x, h)$ ($x \in V^{(q)}$, $h > 0$) is the specified kernel function for the attribute a_q with the center in $x_\nu^{(q)}$ and with the smoothing parameter h. Using this approach the condition attributes with missing values are accepted. If the condition attribute a_q is continuous then

$$K_\nu^{(q)}(x, h) = \begin{cases} n(x, \bar{x}_j^{(q)}, [s_j^{(q)}]^2) & \text{for missing } x_\nu^{(q)} \\ n(x, x_\nu^{(q)}, h) & \text{otherwise} \end{cases} \ , \tag{1.9}$$

where $n(x, m, h) = (2\pi h)^{-\frac{1}{2}} \exp[-\frac{(x-m)^2}{2h}]$ is the probability density function for a normal distribution (with mean m and variance h), and $\bar{x}_j^{(q)}$, $s_j^{(q)}$ are, respectively, the mean value and the standard deviation of these values of the attribute a_q which are related with the j-th decision.

The problem of selecting the kernels $K_\nu^{(q)}$ for discrete attributes is more complex [7]. A diffusion approach is used to solve this problem. The values of the set $V^{(q)}$ can be identified with small air-tight boxes, which are connected by the

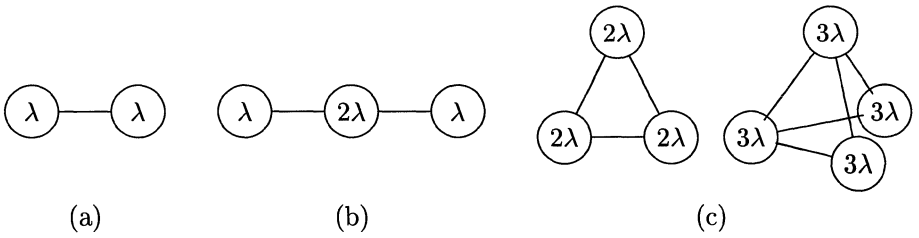


Figure 1 Illustration of the kernel construction for (a) binary, (b) ordinal, (c) nominal discrete condition attributes.

pipes of the same diameter. In the moment $h = 0$ the unit amount of gas is placed in the box corresponding to the value $y \in V^{(q)}$. For $h \geq 0$ the value of $K_q(x, y, h)$ is defined as the amount of gas in the box which corresponds to the value $x \in V^{(q)}$. Evidently

$$K_q(x, y, 0) = \begin{cases} 1 & \text{for } x \neq y \\ 0 & \text{otherwise} \end{cases} \tag{1.10}$$

The idea of the pipe-line construction for the discrete condition attributes with the values expressed in different scales is illustrated in Figure 1. A specific value of the parameter λ depends on the pipe diameter. In computations we assume $\lambda = 1$. The inverse of the number given in the net mesh is equal to the mean value of the time intervals during which a gas particle is staying in the box. The gas diffusion process in the net is described by the system of differential equations $\frac{d}{dh}K = AK$, where the matrix A is

$$\begin{pmatrix} -\lambda & \lambda \\ \lambda & -\lambda \end{pmatrix}, \quad \begin{pmatrix} -\lambda & \lambda & 0 & \dots & 0 \\ \lambda & -2\lambda & \lambda & \dots & 0 \\ 0 & \lambda & -2\lambda & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -\lambda \end{pmatrix}, \tag{1.11}$$

$$\begin{pmatrix} (1-p)\lambda & \lambda & \dots & \lambda \\ \lambda & (1-p)\lambda & \dots & \lambda \\ \dots & \dots & \dots & \dots \\ \lambda & \lambda & \dots & (1-p)\lambda \end{pmatrix}, \tag{1.12}$$

for, binary, ordinal or nominal condition attributes, respectively. The parameter p in the matrix (1.12)is equal to the number of levels of the nominal

condition attribute. The kernel $K_\nu^{(q)}(x,h)$ is defined by the formula

$$K_\nu^{(q)}(x,h) = \begin{cases} \frac{n_j^{(q)}(x)}{n_j^{(q)}} & \text{for missing } x_\nu^{(q)} \\ K_q(x, x_\nu^{(q)}, h) & \text{otherwise} \end{cases}, \tag{1.13}$$

where $n_j^{(q)}(x)$ is the cardinality of the set $\{u_\nu \in U : x_\nu^{(q)} = x, d_\nu = j\}$, and $n_j^{(q)}$ is the number of objects which are related with the j–th decision and have no missing values for the a_q attribute. In the most general case one could optimize the smoothing parameters $h_j^{(q)}$ in (1.8) with regard to all dimensions $q = 1,\ldots,m$, and for all decisions $j = 1,\ldots,l$. In practice, however, it is sufficient to consider only a few number of parameters [9]. The optimization is limited to two parameters h_c and h_d, such that $h_j^{(q)} = [s_j^{(q)}]^2 h_c$ for continuous attributes and $h_j^{(q)} = h_d$ for all discrete condition attributes.

The mean cost of decision-making with an arbitrary classifier may be estimated using unseen objects from the given test sample. Let $n(j \to j')$ be the number of test objects with the j–th decision classified as the objects with the j'-th decision. If the frequencies $\frac{n_j}{N}$ in the test sample are approximately equal to the prior probabilities π_j then the estimate of the mean cost is

$$\hat{C} = \frac{1}{N}\left\{\sum_{j,j'} C(j \to j') n(j \to j')\right\}, \tag{1.14}$$

otherwise

$$\hat{C} = \sum_j \pi_j \left\{\sum_{j'} C(j \to j') \frac{n(j \to j')}{n_j}\right\}. \tag{1.15}$$

The mean cost can also be estimated using the leave-one-out method with the learning sample only. The values of the smoothing parameters h_c and h_d leading to the minimal costs $\hat{C}$ can be found using this method. When $h_c \to 0$ and $h_d = 0$ then the kernel-based estimators (1.6) of p_{ij} with $f_j(x)$ estimated by (1.8), are equivalent to the frequency-based estimators (1.5).

4 DECISION RULE GENERATION

The process of the rule generation has to be proceeded by a secondary coding of the condition attributes. The coding of a_q is performed by selecting a certain

partition $\mathcal{F}^{(q)} = \{\Delta_1^{(q)}, \ldots, \Delta_{n_q}^{(q)}\}$ of the set $V^{(q)}$ into the feasible sets. The collection $\mathcal{F} = (\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(m)})$ of the condition attributes partition generates the partition $\mathcal{S}(\mathcal{F})$ of the space $\mathcal{V}$ into the feasible subsets

$$\Delta_{i_1}^{(1)} \times \ldots \times \Delta_{i_m}^{(m)}, \qquad 1 \leq i_1 \leq n_1, \ldots, 1 \leq i_m \leq n_m . \tag{1.16}$$

Hence, because of (1.3), the cost $C(\mathcal{F})$ can be defined as $C(\mathcal{S}(\mathcal{F}))$. The process of the secondary coding consists in constructing the sequence of the collections $\mathcal{F}_0$, $\mathcal{F}_1$, ..., $\mathcal{F}_r$, such that $\mathcal{F}_0 = (\{V^{(1)}\}, \ldots, \{V^{(m)}\})$, and if

$$\mathcal{F}_i = \{\{\Delta_1^{(1)}, \ldots, \Delta_{n_1}^{(1)}\}, \ldots, \{\Delta_1^{(m)}, \ldots, \Delta_{n_m}^{(m)}\}\},$$

then in the successive step of the algorithm we find the set $\Delta_i^{(q)}$ and its partition $\Delta_i^{(q)} = \tilde{\Delta}_i^{(q)} \cup \bar{\Delta}_i^{(q)}$ into the feasible subsets that for the collection

$$\mathcal{F}_{i+1} = \{\{\Delta_1^{(1)}, \ldots, \Delta_{n_1}^{(1)}\}, \ldots, \{\Delta_1^{(q)}, \ldots, \tilde{\Delta}_i^{(q)}, \bar{\Delta}_i^{(q)}, \ldots \Delta_{n_q}^{(q)}\}, \ldots, \{\Delta_1^{(m)}, \ldots, \Delta_{n_m}^{(m)}\}\}$$

the maximal cost fall $C(\mathcal{F}_i) - C(\mathcal{F}_{i+1})$ is observed. The number of the algorithm steps is given in advance or is determined using some additional information obtained, for example, from the test sample.

In the second stage of classifier generation the partitions of the form $\mathcal{S} = \{\Delta_1, \ldots, \Delta_k\}$ are determined under the following conditions:

1. each element Δ_i of $\mathcal{S}$ is the feasible subset in $\mathcal{V}$,
2. each element Δ_i of $\mathcal{S}$ is the sum of the partition elements from $\mathcal{S}(\mathcal{F}_r)$ for which a common permissible decision exists,
3. the number of the elements in the partition S is minimal under the two previous conditions.

In the third stage of classification algorithm construction, the permissible classifiers for the partitions $\mathcal{S}$ obtained in the second stage are selected.

In computations, the set of the feasible intervals for the continuous attribute a_q have to be finite. The endpoints of these intervals have been selected using the method of intermediate values generation presented in [8].

5 ILLUSTRATIVE EXAMPLES

In this section we illustrate the method of classifier generation in the mixed case of discrete and continuous condition attributes by two case studies from the credit assessment domain. The *credit* dataset was used by J.R.Quinlan [10]. The *German* dataset was created by H.Hofmann from the University of Hamburg. Both datasets were donated to the repository of machine learning databases at the University of California in Irvine. In both datasets credit applicants (objects) are described by the sets of condition attributes characterizing the applicants. With each object one of two decisions is associated, good or bad credit applicant. The values of the condition discrete attributes are expressed in different scales: binary, e.g., whether the applicant is a foreign worker, ordinal, e.g., a number of existing credits at the bank, or nominal, e.g., personal status and sex. Also a number of continuous condition attributes, for example, credit amount, is used to characterize the applicants.

number r of steps	error rate	
	"frequency"	"kernel"
0	0.4450	0.4450
1	0.1532*	0.1532*
2	0.1763	0.1590
3	0.1763	0.1705
4	0.1734	0.1676
5	0.1734	0.1676
6	0.1763	0.1763

Table 1 Results obtained during the first stage of classifier construction for the *credit* dataset.

The learning sample for the *credit* dataset has been composed of 154 randomly selected objects with the decision "+" and 192 randomly selected objects with the decision "−". The test sample has been composed of the left objects: 153 with the decision "+" and 191 with the decision "−". The following optimal values of the coefficients determining the smoothing parameters have been found: $h_c = 0.23$, $h_d = 0.03$. The prior probabilities $\pi_- = 0.555$ and $\pi_+ = 0.445$ are equal to the observed frequencies of the decisions in the dataset. The misclassification cost is defined by $C(\text{“}+\text{”} \rightarrow \text{“}-\text{”}) = 1$, $C(\text{“}-\text{”} \rightarrow \text{“}+\text{”}) = 1$, $C(\text{“}-\text{”} \rightarrow \text{“}-\text{”}) = 0$, $C(\text{“}+\text{”} \rightarrow \text{“}+\text{”}) = 0$, which corresponds to the minimization of the error rate.

Table 1 contains the results obtained during the classifier construction for the frequency-based and kernel-based estimators. For each number of steps $r = 1, \ldots, 6$, the error rates have been estimated on the test sample using the formula (1.15). The case $r = 0$ corresponds to the trivial classifier when all the objects are related with the same decision "−". The minimal error rate is obtained for $r = 1$, which corresponds to the classification algorithm with two classification rules. The algorithm assigns the decision "−" to the objects with $a_9 = f$, and the decision "+" to the objects with $a_9 = t$.

The learning sample for the *German* dataset has been composed of 250 randomly selected objects with the decision *good* and 250 randomly selected objects with the decision *bad*. The test sample has been composed of the left objects: 450 with the decision *good* and 50 with the decision *bad*. The following optimal values of the coefficients determining the smoothing parameters have been found: $h_c = 0.6$, $h_d = 0.5$. The prior probabilities should reflect the proportions of good and bad credit applicants in the population of all bank clients. The values $\pi_{good} = 0.9$ and $\pi_{bad} = 0.1$ have been assumed in computations. The misclassification costs have been defined by the donator of the dataset: $C(good \to bad) = 1$, $C(bad \to good) = 5$, $C(bad \to bad) = 0$, $C(good \to good) = 0$. Table 2 contains the results obtained during the classi-

number r of steps	misclassification cost	
	"frequency"	"kernel"
0	0.5000	0.5000
1	0.4300	0.4920
2	0.4060	0.4920
3	0.3980*	0.4940
4	0.4740	0.4840
5	0.4700	0.4720*
6	0.4620	0.4860

Table 2 Results obtained during the first stage of classifier construction for the *German* dataset.

fier construction for the frequency-based and kernel-based estimators. For each number of steps $r = 1, \ldots, 6$, the misclassification costs have been estimated on the test sample using the formula (1.15). The case $r = 0$ corresponds to the trivial classifier which associates the decision *good* with each credit applicant. The minimal cost of decision-making is obtained for $r = 3$ using the

"frequency" method. It corresponds to the classification algorithm with the following four classification rules:

1. **if** $a_5 \geq 8918$ **then** $d = bad$,
2. **if** $a_5 < 8918$ **and** $a_2 < 11.5$ **then** $d = good$,
3. **if** $a_5 < 8918$ **and** $a_2 \geq 11.5$ **and** $a_1 = 1$ **then** $d = bad$,
4. **if** $a_5 < 8918$ **and** $a_2 \geq 11.5$ **and** $a_1 \in \{2, 3, 4\}$ **then** $d = good$.

Table 3 includes characterization of each classification rule by the estimate of its probability and by the cost of decisions.

Rule	$P(\Delta_i)$	$C(\Delta_i \rightarrow good)$	$C(\Delta_i \rightarrow bad)$
1	0.0324	0.0540	0.0216
2	0.2284	0.0440	0.2196
3	0.1544	0.1960	0.1152
4	0.5848	0.2060	0.5436

Table 3 Characterization of the classification rules for the *German* dataset.

SUMMARY

The presented classifiers combine features of rule inductive systems based on the rough set theory with statistical approximation of datasets probabilistic structure. The datasets structure is approximated from the learning sample using the nonparametric estimators of unknown probabilities. The final decision algorithm minimizes the cost of misclassification. The proposed procedure of classifier construction accepts both continuous and discrete attributes. Continuous attributes are directly incorporated by the procedure. The method handles nominal attribures with unordered values. Missing values in the data are also accepted. The resulting decision rules include only those of the primary attributes which possess good classification properties. A particular form of the final decision algorithm depends on an *a priori* definition of the unit costs of wrong classification and the unit profits of correct classification. The strength of each decision rule is characterized by the estimate of its probability and by the estimate of the decision costs. The method leaves room for non-determinism in assignment of classes to partition blocks of the underlying

attribute space, which is important when several classes with very close values of the cost compete for assignment.

Acknowledgements

This study was supported by the State Committee for Scientific Research (KBN) under grant no. 8 S503 033 06. Thanks to anonymous reviewiers for helpful comments.

REFERENCES

[1] Pawlak, Z., "Rough sets," International Journal of Computer and Information Science, Vol.11, 1982, pp. 341–356.

[2] Pawlak, Z., "Rough Sets: Theoretical Aspects of Reasoning about Data," Kluwer Academic Publishers, Dordrecht, 1991.

[3] Slowinski, R. (ed.), "Intelligent Decision Support: Handbook of Applications and Advances of the Rough Set Theory," Kluwer, Dordrecht, 1992.

[4] Ziarko, W. (ed.), "Rough Sets, Fuzzy Sets and Knowledge Discovery," Springer-Verlag, London, 1994.

[5] Lin, T.Y. and Wildberger A.M. (eds), "Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discowery," Simulation Councils, Inc., San Diego, 1995.

[6] Lachenbruch, P.A., "Discriminant Analysis," Hafner Press, New York, 1975.

[7] Hand, D.J., "Kernel Discriminant Analysis," Research Studies Press, New York, 1982.

[8] Lenarcik, A. and Piasta, Z., "Rough classifiers," in Ziarko, W. (ed.), "Rough Sets, Fuzzy Sets and Knowledge Discovery," Springer-Verlag, London, 1994, pp. 298–316.

[9] Titterington, O.M., "A comparative study of kernel-based density estimates for categorical data," Technometrics, Vol. 22, 1980, pp. 259-268.

[10] Quinlan, J.R., "C4.5: Programs for Machine Learning," Kaufmann, San Mateo, 1993.

20

ALGEBRAIC FORMULATION OF MACHINE LEARNING METHODS BASED ON ROUGH SETS, MATROID THEORY, AND COMBINATORIAL GEOMETRY

Shusaku Tsumoto and Hiroshi Tanaka

Department of Information Medicine,
Medical Research Institute,
Tokyo Medical and Dental University
1-5-45 Yushima, Bunkyo-city Tokyo 113
Japan

ABSTRACT

In order to acquire knowledge from databases, there have been proposed several methods of inductive learning, such as ID3 family and AQ family. These methods are applied to discover meaningful knowledge from large databases, which shows they are useful. However, since there has been no formal approach proposed to treat these methods, efficiency of each method is only compared empirically. In this paper, we introduce matroid theory and rough sets to construct a common framework for empirical machine learning methods which induce the combination of attribute-value pairs from databases. Combination of the concepts of rough sets and matroid theory gives us an excellent set-theoretical framework and enables us to understand the differences and the similarities between these methods clearly. In this paper, we compare three classical methods, AQ, Pawlak's Consistent Rules and ID3. The results show that there exist the differences in algebraic structure between the former two and the latter and that this causes the differences between AQ and ID3.

1 INTRODUCTION

1.1 Motivation

In order to acquire knowledge from databases, there have been proposed several methods of inductive learning, such as ID3 family[2, 3, 16] and AQ family[1, 12, 13]. These methods are applied to discover meaningful knowledge from large databases, which shows that these methods are useful. However, since there has been no formal approach proposed to treat these methods, efficiency of each method is compared by using real-world databases[1, 2, 13, 16], such as medical databases. These results suggest some differences between these methods. However, since sometimes these differences may depend on applied domains, general discussion is left unsolved.

In this paper, we introduce matroid theory[20, 21, 22] and rough sets[15] to construct a common framework for empirical machine learning methods which induce knowledge from attribute-value pattern databases. Combination of the concepts of rough sets and matroid theory gives us an excellent set-theoretical framework and enables us to understand the differences of these methods clearly. Using this framework, we compare three classical methods: AQ, Pawlak's Consistent Rules[15] and ID3 and we obtain seven interesting conclusions from our approach. First, while AQ and Pawlak's method are equivalent to the greedy algorithm for finding the bases of a matroid from the space spanned by attribute-value pairs, ID3 method calculates ordered greedoids, which are defined by weaker axioms than matroids. Second, a matroid defined by AQ method(AQ matroid) is a dual matroid of one defined by Pawlak's method[15](Pawlak's matroid). Third, according to the computational complexity of the greedy algorithm, the efficiency of both methods depends on the number of attributes, especially, independent variables. Fourth, the induced results are optimal to the training samples if and only if the conditions on independence are hold. Thus, if the addition of new examples makes independent attributes change their nature into dependent ones, then the condition of deriving optimal solution is violated. Fifth, in addition to the fourth conclusion, because a greedoid of ID3 has weaker structure than the other two methods, ID3 method is the most sensitive to training samples, although its computational complexity is the lowest. Sixth, if we get a well-defined weight function for predictive accuracy, then the solutions derived by the greedy algorithm would be optimal to prediction. Finally, seventh, we can also apply knowledge on combinatorial geometry, because matroid theory is closely related with combinatorial geometry.

The paper is organized as follows: section in Section 2 and 3, we briefly discuss original rough set model and AQ method respectively. In Section 4, the elementary concepts of matroid theory are introduced, and several characteristics are discussed. Section 5 presents comparison between AQ matroid and Pawlak's matroid. Section 6 gives ID3 greedoid, which is algebraic structure of decision tree induction. In Section 7, we discuss optimal solutions given by the greedy algorithm. Section 8 presents discussion of pruning and truncation in terms of rough sets and matroid theory. In Section 9, we briefly discuss the relationship between matroid theory and combinatorial geometry. Finally, in Section 10, we conclude the results of this paper.

1.2 Notations and Some Assumptions

In this paper, we focus on algebraic specification of domain-independent aspects of classical empirical learning methods, AQ, Pawlak's method and ID3. Thus, we do not consider about constructive generalization[12], since this method explicitly needs domain-specific knowledge. And, moreover, we omit the proofs of the theorems, since all the proofs in Section 4 are originated from matroid theory, which readers could refer to [20, 21], and since the proofs of most of the theorem in Section 5 and Section 6 are trivial. For further information on rough sets and matroid theory, readers could refer to [15, 20, 21].

Below in this subsection, we mention about the following four notations used in this paper. First, for simplicity, we deal with classification of two classes, one of which are supported by a set of positive examples, denoted by D and the other of which are by a set of negative examples, $U - D$. Moreover, we assume that D is decomposed into several disjoint subsets, denoted by $D = \cup_j D_j$. Second, we regard an attribute-value pair as an **elementary equivalence relation** as defined in rough sets[15]. That is, the combination of attribute-value pairs, which is called *the complex of selectors* in terms of AQ theory, is denoted by an equivalence relation, R. A set of elements which supports R, which is called a *partial star* in AQ, is referred to as an **indiscernible set**, denoted by $[x]_R$. For example, let $\{1, 2, 3\}$ be a set of samples which supports an equivalence relation R. Then, we say that a partial star of R is equal to $\{1, 2, 3\}$ in terms of AQ. This notion can be represented as $[x]_R = \{1, 2, 3\}$ in terms of rough sets. Third, when we describe a conjunctive formula, we use the ordinary logical notation. Furthermore, when an equivalence relation is described as an attribute-value pair, denoted by $[attribute = value]$. For example, if an equivalence relation R means "a=1 and b=0", then we write it as $R = [a = 1] \bigwedge [b = 0]$. Finally, we define partial order of equivalence as follows:

Definition 1 *Let $A(R_i)$ denote the set whose elements are the attribute-value pairs included in R_i. If $A(R_i) \subseteq A(R_j)$, then we represent this relation as:*

$$R_i \preceq R_j.$$

For example, let R_i represent a conjunctive formula, such as $a \wedge b \wedge c$, where a, b, c are elementary equivalence relations. Then $A(R_i)$ is equal to $\{a, b, c\}$. If we use the notation of Michalski's APC(Annotated Predicate Calculus)[12], R_i can be represented as, say $[a = 1]\&[b = 1]\&[c = 1]$, then $A(R_i)$ is equal to a set of selectors, $\{[a = 1], [b = 1], [c = 1]\}$.

2 ROUGH SET THEORY

2.1 Elementary Concepts of Rough Set Theory

Rough set theory is one of the most important approaches, which characterizes classification from the viewpoint of set theory, developed and rigorously formulated by Pawlak[15]. This theory can be used to acquire certain sets of attributes for classification and can also evaluate how precisely the attributes of database are able to classify data. In this paper, we only mention what we need in relation to our reasoning strategy, because including whole discussion of rough sets is too lengthy and redundant. For further information, readers could refer to [15].

Table 1 is a small example of database which collects the patients who complain of headache. First, let us consider how an attribute "location" classifies the headache patients' set of the table. The set whose value of the attribute "location" is equal to "whole" is {2,8,10}(In the following, the numbers represent each record number). This set means that we cannot classify {2,8,10} further solely by using the constraint $R = [loc = whole]$. Thus, we refer to this set as the indiscernible set over relation R, denoted by $[x]_R$=U/R= {2,8,10} (U denotes the total set of database). In this set, {2,10} suffer from muscle contraction headache("m.c.h."), {8} suffers from intracranial mass lesion("i.m.l."). Hence other additional attributes are needed to classify this set of patients as to their disease. Using this concept, we can evaluate the classificatory power of each attribute. For example, $[prod = 1]$ is specific to the case of classic migraine ("classic"). It is notable that this notion can be

Table 1 A Small Example of Databases

No.	loc	nat	his	prod	jolt	nau	M1	M2	class
1	occular	per	per	0	0	0	1	1	m.c.h.
2	whole	per	per	0	0	0	1	1	m.c.h.
3	lateral	thr	par	0	1	1	0	0	common.
4	lateral	thr	par	1	1	1	0	0	classic.
5	occular	per	per	0	0	0	1	1	psycho.
6	occular	per	subacute	0	1	1	0	0	i.m.l.
7	occular	per	acute	0	1	1	0	0	psycho.
8	whole	per	chronic	0	0	0	0	0	i.m.l.
9	lateral	thr	per	0	1	1	0	0	common.
10	whole	per	per	0	0	0	1	1	m.c.h.

Definition. loc: location, nat: nature, his:history,
Definition. prod: prodrome, nau: nausea, jolt: Jolt headache,
M1, M2: tenderness of M1 and M2, 1: Yes, 0: No, per: persistent,
thr: throbbing, par: paroxysmal, m.c.h.: muscle contraction headache,
psycho.: psychogenic pain, i.m.l.: intracranial mass lesion, common.:
common migraine, and classic.: classical migraine.

extended to multivariate cases, such as $[x]_{[loc=whole]\wedge[M2=1]} = \{2, 10\}$. Moreover, an attribute itself can be also taken as a relation. For example, $[x]_{loc} = \{[x]_{[loc=whole]}, [x]_{[loc=lateral]}, [x]_{[loc=occular]}\} = \{\{2, 8, 10\}, \{3, 4, 9\}, \{1, 5, 6, 7\}\}$. Using these basic concepts, several specific sets and measures for these sets are defined as follows: [1]

Definition 2 *Let R be an equivalence relation and X be the subset of U.*

$$\begin{array}{ll} \textit{positive region} & Posi_R(X) = \bigcup\{Y \in U/R : Y \subseteq X\} \\ \textit{possible region} & Poss_R(X) = \bigcup\{Y \in U/R : Y \bigcap X \neq \phi\} \\ \textit{boundary region} & Bound_R(X) = Poss_R(X) - Posi_R(X) \\ \textit{accuracy measure} & \alpha_R(X) = \dfrac{card\ \ Posi_R(X)}{card\ \ Poss_R(X)}. \end{array}$$

□

For example, the set whose class is "m.c.h." is composed of {1,2,5,10}. Let us take this set as X. Then the following relations R_i are obtained such that

[1] In [15], Pawlak does not use the above term "possible region", but he refers to the above possible region as *upper* approximation of X. Compared with the word "positive", we use "possible" which reflects the intuitional meaning of an *upper* approximation.

$U/R_i \bigcap X \neq \phi$:

$$\begin{aligned} R_1 &= [loc = occular] \wedge [nat = per] \wedge [his = per] \\ &\quad \wedge [prod = 0] \wedge [jolt = 0] \wedge [nau = 0] \wedge [M1 = 1] \wedge [M2 = 1], \text{ and} \\ R_2 &= [loc = whole] \wedge [nat = per] \wedge [his = per] \\ &\quad \wedge [prod = 0] \wedge [jolt = 0] \wedge [nau = 0] \wedge [M1 = 1] \wedge [M2 = 1]. \end{aligned}$$

Let R denote $R_1 \cup R_2$. That is, $U/R = \{[x]_{R_1}, [x]_{R_2}\} = \{\{1,5\}, \{2,10\}\}$. The positive region of "m.c.h." over the relation R is $U/R - \{1,5\} = \{2,10\}$. And then its possible region is {1,2,5,10}, which includes one case for "psycho": {5}. Furthermore, we can derive {1,5} as the boundary region, and the accuracy measure is 2/4.

2.2 Pawlak's Consistent Rules

Based on the concepts of rough sets, Pawlak[15] introduces *Reduction of Knowledge*, which is a method to examine the independencies of the attributes iteratively and extract the minimum indispensable part of equivalence relations. Here we only mention about the definition of *consistent rules* and their knowledge reduction. For further details, readers could refer to [15].

Definition 3 *Let R_j be an equivalence relation and D be a set of samples which belongs to a target concept. $R_j \Rightarrow D$ is called a consistent rule when $Posi_{R_j}(D)$ is given by:*

$$Posi_{R_j}(D) = D_k = [x]_{R_j} \subseteq D,$$

where $Posi_{R_j}(D)$ denotes the positive region of D in terms of R_j. □

Definition 4 *Let R_0 be equal to $R \bigwedge [a = v]$, where $[a = v]$ denotes a certain attribute-value pair. If an attribute-value pair $[a = v]$ is satisfied with the following equation:*

$$Posi_{R_0}(D) = Posi_{R \bigwedge [a=v]}(D) = Posi_R(D),$$

then we say that $[a = v]$ is dispensable in R_0, and can be deleted from R. □

Intuitively, reduction procedure removes redundant variables which do not contribute to classification of a class. For the above example, because $Posi_{R_2}(X) = \{2, 10\}$, $R_2 \Rightarrow X$ is a consistent rule of X. Here, let us focus on an attribute "prod". That is, let R_2 be decomposed into $R_3 \wedge [prod = 0]$, where R_3 is equal to $[loc = whole] \wedge [nat = per] \wedge [his = per] \wedge [jolt = 0] \wedge [nau = 0] \wedge [M1 = 1] \wedge [M2 = 1]$. Then, because $[x]_{R_3} = [x]_{R_2}$, or $Posi_{R_3 \wedge [prod=0]}(X) = Posi_{R_3}(X)$, this attribute can be deleted. Applying this method iteratively, the following minimum equivalent relations are obtained: $[loc = whole] \wedge [M1 = 1]$ and $[loc = whole] \wedge [M2 = 1]$.

If we use some weight function for efficiency, this algorithm can be viewed as the greedy algorithm which finds independent variables. However, while AQ is based on incremental addition of equivalence relations, Pawlak's method is based on incremental removal of dependent equivalence relations. This characteristic is also discussed in Section 5.

3 AQ METHOD

3.1 Bounded Stars as Positive Regions

AQ is an inductive learning system based on incremental STAR algorithm[12]. This algorithm selects one "seed" from positive examples and starts from one "selector"(attribute-value pair) contained in this "seed" example. It adds selectors incrementally until the "complexes" (conjunction of attributes) explain only positive examples, called a **bounded star**. In general, many complexes can satisfy these positive examples. Thus, AQ finds the most preferred ones, according to a flexible extra-logical criterion.

It would be worth noting that the complexes supported only by positive examples corresponds to the lower approximation, or the positive region in rough set theory. That is, the rules induced by AQ is equivalent to consistent rules defined by Pawlak when constructive generalization rules[12] are not used. In fact, AQ's star algorithm without constructive generalization can be reformulated by the concepts of rough sets. For example, a bounded star denoted by $G(e|U - D, m)$ in Michalski's notation is equal to $G = \{R_i | [x]_{R_i} = D_j\}$, such that $|G| = m$ where $|G|$ denotes the cardinality of G. This star is composed of many complexes, which is ordered by LEF_i, lexicographic evaluation functional, which is defined as the following pair:$< (-negcov, \tau_1), (poscov, \tau_2) >$ where $negcov$ and $poscov$ are numbers of negative and positive examples, respectively, covered by

an expression in the star, and where τ_1 and τ_2 are tolerance thresholds for criterion $poscov, negcov$ ($\tau \in [0..100\%]$). This algorithm shows that AQ method is a kind of greedy algorithm which finds independent variables using selectors which are equivalent to equivalence relations in terms of rough sets. We will discuss this characteristic later in Section 5.

3.2 INDUCE method of AQ algorithm

An algorithm to derive a bounded star is called INDUCE method in AQ algorithm [12]. Here we illustrate how this INDUCE method works. For example, let us consider the above example of database shown in Table 1, which collects the patients who complain of headache. If the second sample, whose record number is 2, is selected as a seed, then an attribute value pair, $[loc = whole]$ can be regarded as a *selector*. Then a partial star is obtained, which includes this seed and supports $[loc = whole]$, as {2,8,10}.

This is not a bounded star for a class "m.c.h." (muscle contraction headache), because the class of "8" is "i.m.l." (intracranial mass lesion), that is, this star includes a *negative example* as to "m.c.h". Thus, additional selectors are required to remove "8" from a star. This means that some selectors are needed in order to get a bounded star. In AQ method, a selector is chosen from the selectors which is supported by the seed, sample "2". For example, if a selector $[his = per]$ is chosen, then a star of $[loc = whole]\&[his = per]$ is equal to {2,10}, which only consists of positive samples as to "m.c.h."

Therefore $[loc = whole] \wedge [his = per]$ can be regarded as a premise of a rule for classification of "m.c.h.", that is, if a sample satisfies $[loc = whole] \wedge [his = per]$, then a class of this sample is "m.c.h.". It is also notable that $[loc = whole] \wedge [M1 = 1]$ and $[loc = whole] \wedge [M2 = 1]$ generate a bounded star whose elements are also {2,10}. In order to choose a suitable selector from possible selectors, some extra-logical criterion should be applied, such as aforementioned LEF criterion, which includes domain-specific knowledge. That is, in AQ algorithm, domain knowledge is applied in order to select attribute-value pairs, or premises of rules from possible combination of those pairs, which are suitable to describe the structure of domain-knowledge.

4 MATROID THEORY

4.1 Definition of Matroids

Matroid theory abstracts the important characteristics of matrix theory and graph theory, firstly developed by Whitney[22] in the thirties of this century. The advantages of introducing matroid theory are the following: 1) Because matroid theory abstracts graphical structure, this shows the characteristics of formal structure in graph clearly. 2) A matroid is defined by the axioms of independent sets. Therefore, it makes the definition of independent structure clear. 3) Duality is one of the most important structures in matroid theory, which enables us to treat relations between dependency and independency rigorously. 4) The greedy algorithm is one of the algorithms for acquiring an optimal base of a matroid. Since this algorithm has been studied in detail, well-established results can be applied to our problem.

Although there are many interesting and attractive characteristics of matroid theory, we only discuss about duality, and the greedy algorithm, both of which are enough for our algebraic specification. For further information on matroid theory, readers might refer to [20].

First, we begin with the definition of a matroid. A matroid is defined as an independent space which satisfies the following axioms:

Definition 5 *The pair* $M(E, \mathcal{J})$ *is called a matroid, if*

1) E *is a finite set,*
2) $\emptyset \in \mathcal{J} \subset 2^E$,
3) $X \in \mathcal{J}, Y \subset X \Rightarrow Y \in \mathcal{J}$,
4) $X, Y \in \mathcal{J}, card(X) = card(Y) + 1 \Rightarrow (\exists a \in X - Y)(Y \cup \{a\}) \in \mathcal{J}$.

If $X \in \mathcal{J}$, *it is called* **independent**, *otherwise* X *is called* **dependent**. □

One of the most important characteristic of matroid theory is that this theory refers to the notion of independence using the set-theoretical scheme. As shown in [15], since rough set theory also considers the independence of the attributes from the viewpoint of set theory, it is expected that our definition of independence with respect to learning methods can be discussed by combination of rough set theory with matroid theory.

4.2 Duality

Another important characteristic is duality. While this concept was firstly introduced in graph theory, a deeper understanding of the notion of the duality in graph theory can be obtained by examining matroid structure. Definition of duality is as follows:

Definition 6 *If $M = (E, \mathcal{J})$, is a matroid with a set of bases β, then the matroid with a set of elements E, and a set of bases $\beta^* = \{E - B | B \in \beta\}$ is termed the* **dual** *of M and is denoted by M^*.* □

From this definition, it can be easily shown that $(M^*)^* = M$, and M and thus M^* are referred to a **dual matroid pair**. And we have the following theorem:

Theorem 1 *If M is a matroid, then M^* is a matroid.* □

4.3 The Greedy Algorithm

Since it is important to calculate a base of a matroid in practice, several methods are proposed. In these methods, we focus on the greedy algorithm. This algorithm can be formulated as follows:

Definition 7 *Let B be a variable to store the calculated base of a matroid, and E denote the whole set of attributes. We define the Greedy Algorithm to calculate a base of a matroid as follows:*

1. $B \leftarrow \phi$.
2. *Calculate "priority queue" Q using weight function of E.*
3. *If B is a base of $M(E, \mathcal{J})$ then stop. Else go to 4.*
4. $e \leftarrow first(Q)$, *which has a minimum weight in Q.*
5. *If $B \cup \{e\} \in \mathcal{J}$ then $B \leftarrow B \cup \{e\}$. goto 2.* □

This algorithm searches one solution which is optimal in terms of one weight function. Note that a matroid may have many bases. The bases derived by the greedy algorithm are optimal to some **predefined** weight function. Hence if we cannot derive a suitable weight function we cannot get such an optimal base. In the following, we assume that we can define a good weight function for

the greedy algorithm. For example, we can use *information gain* as defined in [2, 16] for such function. When information gain is used as a weight function, the greedy algorithm with this weight function gives a solution optimal to apparent accuracy. since this gain is closely related with apparent accuracy or apparent accuracy. In other words, the solution is optimal to apparent rate, that is, in the language of statistics, the algorithm calculates the best class allocation of training samples. Under this assumption, this algorithm has the following characteristics:

Theorem 2 *The complexity of the greedy algorithm is*

$$\mathcal{O}(mf(\rho(M)) + m\log m),$$

where $\rho(M)$ is equal to a rank of matroid M, m is equal to the number of the elements in the matroid, $|E|$, f represents a function of computational complexity of an independent test, which is the procedure to test whether the obtained set is independent, and is called independent test oracle. □

Theorem 3 *The optimal solution is derived by this algorithm if and only if a subset of the attributes satisfies the axioms of the matroid.* □

This theorem is very important when we discuss the optimal solutions of learning algorithms. This point is discussed in Section 7.

4.4 Unions and Intersections of Matroids

Because matroid theory is based on set-theoretical framework, we can define unions and intersections of matroids. [2].

First, we define the union of matroids as follows.

Definition 8 *Let $M_1, M_2, \cdots, M_m$ be matroids on S. Let*

$$\mathcal{J} = \{X : X = X_1 \cup X_2 \cup \cdots \cup X_m; X_i \in \mathcal{J}(M_i)\ (1 \leq i \leq m)\}.$$

[2]Unfortunately, intersections of matroids do not always satisfy the axioms of a matroid in general [20] However, in this paper, we deal with only special class of a matroid, called *simple matroids*, whose intersections always satisfy the axioms of a matroid.

Then $\mathcal{J}$ is the collection of independent sets of a matroid on S, which satisfies the axioms of independent sets. We refer to the matroid M whose independent sets are equal to $\mathcal{J}$ as:

$$M = M_1 \vee M_2 \vee \cdots \vee M_m,$$

called the union of matroids. □

Therefore, it guarantees that the problem can be decomposed into disjoint subproblems and that the total solution can be obtained by taking unions of the solutions of sub-problems. That is, let the whole problem denote S. We first try to decompose S into $M_1 \vee M_2$. Then we will calculate both bases, and finally take the union of both bases. We use this result in the subsequent sections.

In the same way, the intersection of a matroid is also defined:

Definition 9 *Let $M_1, M_2, \cdots, M_m$ be matroids on S. Let*

$$\mathcal{J} = \{X : X = X_1 \cap X_2 \cap \cdots \cap X_m; X_i \in \mathcal{J}(M_i)\ (1 \leq i \leq m)\}.$$

Then $\mathcal{J}$ is the collection of independent sets of a matroid on S, which satisfies the axioms of independent sets. We refer to the matroid M whose independent sets are equal to $\mathcal{J}$ as:

$$M = M_1 \wedge M_2 \wedge \cdots \wedge M_m,$$

called the intersection of matroids. □

It is notable that the intersection of matroids exactly corresponds to the *core* of reducts [15]. We illustrate this notion in the next section.

5 AQ MATROIDS AND PAWLAK'S MATROIDS

Here we show that our "rough sets" formalization of AQ algorithm is equivalent to the greedy algorithm for calculating bases of a matroid and that the derived bases are dual to those derived by Pawlak's reduction method.

5.1 AQ matroids

Under the above assumption we can constitute a matroid of AQ method, which we call *AQ matroid* as follows:

Theorem 4 *Let B denote the base of a matroid such that $[x]_B = D_k$. If we define an independent set $\mathcal{J}(D_k)$ as $\{A(R_j)\}$ which satisfies the following conditions:*

1) $R_j \preceq B$,
2) $[x]_B \subseteq [x]_{R_j}$,
3) $\forall R_i$ *s.t.* $R_i \prec R_j \preceq B$, $D_j = [x]_B \subseteq [x]_{R_j} \subset [x]_{R_i}$,

where the equality holds only if $R_j = B$. Then this set satisfies the definition of a matroid. We call this type of matroid, $M(E, \mathcal{J}(D_k))$, a AQ matroid. □

The first condition means that a base is a maximal independent set and each relation forms a subset of this base. And the second condition is the characteristic which satisfies all of these equivalence relations. Finally, the third condition denotes the relationship between the equivalence relations: Any relation R_i which forms a subset of $A(R_j)$ must satisfy $[x]_{R_j} \subset [x]_{R_i}$. Note that these conditions reflect the conditional part of AQ algorithm. For example, let us consider the example shown in Table 1. Let us take two equivalence relations, $[loc = whole]$ and $[M1 = 1]$. $[x]_{[loc=whole]}$ and $[x]_{[M1=1]}$ are equal to {2,8,10}, and {1,2,5,10}. Because these two sets are supersets of $D = [x]_{[loc=whole] \wedge [M1=1]} = \{2, 10\}$, which is a positive region of class "m.c.h.", we derive the following relations: $D \subset [x]_{[loc=whole]}$ and $D \subset [x]_{[M1=1]}$. Therefore, $\{[loc = whole]\}$, {[M1=1]}, and $\{[loc = whole] \wedge [M1 = 1]\}$ belong to the independent sets of the target concept, classification of a class "m.c.h.".

Note that D has the other two bases, $\{[loc = whole] \wedge [M2 = 1]\}$ and $\{[loc = whole] \wedge [his = per]\}$, Since D has totally three disjoint bases, the base of a matroid is derived as the union of three bases, $\{[loc = whole], [M1 = 1]\} \cup \{[loc = whole], [M2 = 1]\} \cup \{[loc = whole], [his = per]\}$. [3]

We can also derive the intersection of three bases, which corresponds to the core of reducts as: $\{[loc = whole]\}$. [4]

[3] These three bases correspond to *reducts* defined in rough set theory [15].

[4] These discussions suggest that the nature of AQ algorithm should be captured by the concepts of rough set theory.

It is also notable that each D_k has exactly one independent set $J(D_k)$. Therefore the whole AQ algorithm is equivalent to the greedy algorithm for acquiring a set of bases of AQ matroid, denoted by $\{\mathcal{J}(D_k)\}$. Furthermore, since the independent test depends on the calculus of indiscernible sets, is less than $\mathcal{O}(\rho(M) * n^2)$ where n denotes a sample size, the computational complexity is given as follows:

Theorem 5 *Assume that we do not use constructive generalization. Then the complexity of AQ algorithm is less than*

$$\mathcal{O}(mn^2\rho(M)) + m\log m),$$

where $\rho(M)$ is equal to a rank of matroid M, m is equal to the number of the elements in the matroid, $|E|$. □

Hence the computational complexity of AQ depends mainly on the number of the elements of a matroid, since it increases exponentially as the number of the attribute-value pairs grows large.

5.2 Pawlak's Matroids

On the other hand, since $\rho(M)$ is the number of independent variables, $m - \rho(M)$ is equal to the number of dependent variables. From the concepts of the matroid theory, if we define a dependent set $\mathcal{I}$ as shown below, then $M(E,\mathcal{I})$ satisfies the condition of the dual matroid of $M(E,\mathcal{J})$.

Theorem 6 *Let B denote the base of a matroid such that $[x]_B = D_k$. If we define an independent set $\mathcal{I}(D_k)$ as $\{A(R_j)\}$ which satisfies the following conditions:*

1) $B \prec R_j$,
2) $[x]_B = [x]_{R_j}$,
3) $\forall R_i$ s.t. $B \prec R_i \preceq R_j$, $D_k = [x]_B = [x]_{R_j} = [x]_{R_i}$,

then $M(E,\mathcal{I}(D_k))$ is a dual matroid of $M(E,\mathcal{J}(D_k))$, and we call $M(E,\mathcal{I}(D_k))$ a Pawlak's matroid. □

The first condition means that a base is a maximal independent set and each relation forms a superset of this base. And the second condition is the char-

acteristic which satisfies all of these equivalence relations. Finally, the third condition denotes the relationship between the equivalence relations: Any relation R_i which forms a subset of $A(R_j)$ must satisfy $[x]_{R_i} \subset [x]_{R_j}$. Note that these conditions reflect the conditional part of reduction method. For example, let us take R_2 in Section 2.1 as an example. In this case, R_2 is equal to a positive region of a class "m.c.h.". If we describe R_2 as $R_2 = R_3 \wedge [jolt = 0] \wedge [nau = 0]$, where R_3 is equal to $[loc = whole] \wedge [nat = per] \wedge [his = per] \wedge [prod = 0] \wedge [M1 = 1] \wedge [M2 = 1]$, $[jolt = 0]$, and $[nau = 0]$, $R_2 = R_3 \wedge [jolt = 0] \wedge [nau = 0]$, then we get the following result: $\{2, 10\} = [x]_{R_2} = [x]_{R_3 \wedge [jolt=0]} = [x]_{R_3}$. Therefore $[jolt = 0]$, $[nau = 0]$, and $[jolt = 0] \wedge [nau = 0]$ are the elements of a Pawlak's matroid.

As shown above, the algorithm of Pawlak's method is formally equivalent to the algorithm for the dual matroid of AQ matroid, and the computational complexity of Pawlak's method is less than $\mathcal{O}((p - \rho(M)) * (n^2 + 2n) + m \log m)$. Hence, we get the following theorem.

Theorem 7 *The complexity of the Pawlak's method is less than*

$$\mathcal{O}(mn^2(p - \rho(M))) + m \log m),$$

where p is a total number of attributes, $\rho(M)$ is equal to a rank of matroid M, and m is equal to the number of the elements in the matroid, $|E|$. □

From these consideration, if $\rho(M)$ is small, AQ algorithm performs better than Pawlak's one under our assumption.

6 ID3 GREEDOIDS

6.1 Induction of Decision Trees and Greedoids

Induction of decision trees, such as CART[2] and ID3[16] is another inductive learning method based on the ordering of variables using information entropy measure or other similar measures. This method splits training samples into smaller ones in a top-down manner until it cannot split the samples, and then prunes the overfitting leaves.

As to pruning methods, we discuss independently later, so here we briefly illustrate splitting procedures. For simplicity, let us consider classification of "m.c.h." in the example shown in Table 1. Then positive samples consist of {1,2,10}. For each attribute, the splitting procedure calculates information gain, which is defined as the difference between the value of entropy measure (or other similar measures) before splitting and the averaged value after splitting. And the procedure selects the attribute which gives the maximum information gain.

For example, since positive examples consist of three elements, the root entropy measure is equal to: $-\frac{3}{10}\log_2\frac{3}{10} = 0.5211$. In the case of an attribute "M1", the number of positive examples in a sample which satisfy $[M1 = 1]$ is three of four, and that of positive examples in a sample which satisfies $[M1 = 0]$ is zero of six. So, the expected entropy measure is equal to: $-\frac{4}{10}(\frac{3}{4}\log_2\frac{3}{4}) - \frac{6}{10}(0\log_2 0) = 0.1245$. Because $0\log_2 0$ is defined as 0, information gain is derived as: $\frac{3}{10}\log_2\frac{3}{10} - \frac{4}{10}(\frac{3}{4}\log_2\frac{3}{4}) = 0.5211 - 0.1245 = 0.3966$. On the other hand, in the case of an attribute "location", since the excepted entropy measure is equal to: $-\frac{4}{10}(\frac{1}{4}\log_2\frac{1}{4}) - \frac{3}{10}(\frac{2}{3}\log_2\frac{2}{3}) - \frac{3}{10}(0\log_2 0) = 0.3170$, information gain of this attribute is equal to: $0.5211 - 0.3170 = 0.2041$. Therefore, "M1" is better for classification at the root. In fact, "M1" is the best attribute for information gain, and the training samples are split into two subsamples, one of which satisfies "M1=1" and the other of which satisfies "M1=0". Then these processes are recursively applied to subsamples. In this case, we get the following small tree for classification of "m.c.h.":

$$\begin{cases} \text{[M1=1]}(m.c.h. : 3 \quad non - m.c.h. : 1) \\ \qquad \begin{cases} \text{[loc=whole]}\cdots\cdots(m.c.h. : 2) \\ \text{[loc=occular]}\cdots\cdots(m.c.h. : 1 \quad non - m.c.h. : 1) \end{cases} \\ \\ \text{[M1=0]}(m.c.h. : 0 \quad non - m.c.h. : 6) \end{cases}$$

"Non-m.c.h." denotes the negative samples with respect to "m.c.h." For each node, (m.c.h.: p non-m.c.h.: n) denotes the number of elements of positive examples and that of negative examples. In other words, p "m.c.h." samples belong to that node, while n negative samples are also included. Note that, in this induction, $[his = per]$ and$[M2 = 1]$ are never derived without using *surrogate split* [2], while both attribute-value pairs are obtained by AQ method.

The main characteristic of the bases derived by ID3 is the following. First, the attribute-value pairs are totally ordered, and in each branch, which corresponds to each base for D_j, subsets of each branch have to preserve this order. For example, let a base be composed of binary attributes, say, $\{a, b, c\}$, in which ID3 the algorithm chooses these attributes from the left to the right. Then

the allowable subsets are: $\{a\}$, $\{a, b\}$, and $\{a, b, c\}$. Second, each base has the common attribute at least in the first element. For example, if one base is composed of $\{a, b, c\}$, then another base is like $\{a, b, \bar{c}\}$, or $\{\bar{a}, d, c\}$, where $\bar{a}$ denotes a complement of a.

Although these global constraints, especially the first one, decreases the search space spanned by attribute-value pairs, those make a family of subsets lose the characteristics of a matroid. In fact, a set of the subsets derived by ID3 method does not satisfy the axiom of a matroid. It satisfies the axiom of a greedoid[21], which is a weaker form of a matroid, defined as follows.

Definition 10 *The pair $M(E, \mathcal{J})$ is called a greedoid, if*

1) E is a finite set,
2) $\emptyset \in \mathcal{F} \subset 2^E$,
3) $X \in \mathcal{F}$, there is an $x \in X$ such that $X - x \in \mathcal{F}$,
4) $X, Y \in \mathcal{F}, card(X) = card(Y) + 1 \Rightarrow (\exists a \in X - Y)(Y \cup \{a\}) \in \mathcal{F}$.

If $X \in \mathcal{J}$, it is called **feasible**, *otherwise X is called* **infeasible**. □

Note that the third condition becomes a weaker form, which allows for the total ordering of elements. Because of this weakness, some important characteristics of matroids, such as duality, are no longer preserved. Hence ID3 has no dual method like AQ and Pawlak's method. However, since the optimality of the greedy algorithm is preserved, so we can discuss these characteristics in the same way. Using the above formulation, the search space for ID3 is defined as an ordered greedoid in the following.

Definition 11 *Let B denote the base of a matroid such that $[x]_B = D_k$. If we define a feasible set $\mathcal{K}(D_k)$ as: $\{A(R_j)\}$ which satisfies the following conditions:*

1) $R_j \preceq B$,
2) $[x]_B \subseteq [x]_{R_j}$,
3) $\forall R_i$ s.t. $R_i \preceq R_j \preceq B$, $D_j = [x]_B \subseteq [x]_{R_j} \subset [x]_{R_i}$,

where the equality holds only if $R_j = B$, and if we demand that the each $K(D_k)$ should satisfy the following conditions:

(1) for all R_i and R_l, $R_i \preceq R_l$ or $R_l \preceq R_j$ holds ,

(2) $\forall$ $\mathcal{K}(D_q)$ and $\mathcal{K}(D_p)$, For all $R_j \in \mathcal{K}(D_q)$ and $R_i \in \mathcal{K}(D_p)$, if $[x]_{R_i} \cap [x]_{R_j} \neq \phi$, then $R_j \preceq R_i$,

then this set satisfies the definition of a greedoid. We call this type of greedoid, $G(E, \mathcal{K}(D_k))$*, a ID3 greedoid.* □

Note that each D_k has exactly one feasible set $\mathcal{K}(D_k)$. For the above example where $D = \{2, 10\}$ as shown in Section 5, $calK(D) = \{[M1 = 1], [M1 = 1] \wedge [loc = whole]\}$.

Therefore the whole ID3 algorithm is equivalent to the greedy algorithm for acquiring a set of bases of ID3 greedoid, denoted by $\{\mathcal{K}(D_k)\}$.

6.2 Computational Complexity of ID3

As shown above, ID3 algorithm is also the greedy algorithm for deriving a base of a greedoid. However, the main feature of this algorithm is that two constraints to independent sets are given. This reduces the search space of independent sets, because the sets which satisfy the above two constraints are not so many. Here, we obtain the following theorem:

Theorem 8 *The complexity of ID3 algorithm is less than*

$$\mathcal{O}(mn^2\rho(M)) + m\log m),$$

where $\rho(M)$ *is equal to a rank of greedoid* M*,*m *is equal to the number of the elements in the greedoid,* $|F|$*.* □

The difference in computational complexity between AQ and ID3 is the value of m. This difference is illustrated as follows. Let all attributes be binary and the total number of attributes be p. Then, for AQ, since the search space is spanned by the whole combination of attribute-value pairs, $|E|$ is almost equal to 2^p. On the other hand, the search space for ID3 is equal to $2^{\rho(M)+1} - 1$. Therefore, if $\rho(M) \ll p$, then the computational complexity of ID3 is much lower than AQ.

Hence, in many cases, this ID3 method is faster than the other two methods. However, it does not mean that ID3 performs well, because some optimal solutions will never be found by the two constraints to independent sets. They will not appear in the space of ID3 greedoid. This phenomenon sometimes

makes ID3 performs worse. For example, when training samples do not reflect the importance of variable, that is, when some less important variables are given more weight than important ones, some relations between important ones can be never found. Therefore this fact explains one aspect that ID3 is more sensitive to training samples.

7 OPTIMAL SOLUTIONS

As discussed in the above section, when we adopt a weight function which is described as a monotonic function of apparent error rate, we obtain an optimal solution which is the best for apparent error rate. Thus, in this case, Theorem 3 tells us that an optimal solution is obtained only when relations between training samples and attributes-value pairs satisfy the conditions of AQ matroid.

However, this assumption is very strict, since apparent error rate depends on only given training samples. In practice, it is often violated by new additional training samples. For example, when in the old training samples, $R_i \prec R_j$ implies $[x]_{R_j} \subset [x]_{R_i}$, additional samples cause the latter relation to be $[x]_{R_j} = [x]_{R_i}$. In other words, additional samples cause independent variables to be dependent. In this case, the former derived solution is no longer optimal to this weight function. This problem is also discussed from the viewpoint of predictive accuracy $\hat{\alpha}_{R_i}(D)$ defined in the following equation:

$$\begin{aligned} \hat{\alpha}_{R_i}(D) &= \frac{card\ \{([x]_{R_i} \bigcap D) \bigcup ([x]^c_{R_i} \bigcap D^c)\}}{card\ \{[x]_{R_i} \bigcup [x]^c_{R_i}\}} \\ &= \varepsilon_{R_i} \alpha_{R_i}(D) + (1 - \varepsilon_{R_i}) \alpha^c_{R_i}(D^c), \end{aligned}$$

where ε_{R_i} denotes the ratio of training samples to total population, $\alpha_{R_i}(D)$ denotes an apparent accuracy, and $\alpha^c_{R_i}(D)$ denotes the accuracy of classification for unobserved cases, $[x]^c_{R_i}$ and D^c.

Therefore the value of ε_{R_i} determines whether $\alpha_{R_i}(D)$ is suitable to predictive classification or not. On one hand, if ε_{R_i} is near to 0, then $\hat{\alpha}_{R_i}(D)$ may be quite different from $\alpha_{R_i}(D)$. So, in this case, an optimal solution based on apparent accuracy is less reliable. On the other hand, if ε_{R_i} is near to 1, then $\hat{\alpha}_{R_i}(D)$ may be equal to $\alpha_{R_i}(D)$. Thus, in this case, an optimal solution based on apparent accuracy is much reliable. As shown in the above formula, since ε_{R_i} is dependent on sampling from total population, predictivity depends on

sampling from total population. Hence it is a very important factor whether sampling is good or not.

The above formula also suggests that, if we have a weight function which is a monotonic function of predictive error rate, then we derive a base optimal to it. Unfortunately, it is impossible to derive such function, since we can only estimate predictive error rate.

Two approaches discuss these functions: one is MDL function [18], and the other is Bayesian model [4], whose usefulness is ensured in their papers. Since MDL function can be viewed as one kind of Bayesian model, we focus on the latter model in this paper. Cestnik and Bratko discuss predictive accuracy insightfully [4] and they obtain the Bayesian formula as shown in Section 6. The above formula is rewritten as:

$$\hat{\alpha}_{R_i}(D) = \frac{card\ [x]_{R_i}}{card\ [x]_{R_i} + m}\alpha_{R_i}(D) + \frac{m}{card\ [x]_{R_i} + m}p_a.$$

Therefore m corresponds to $card\ [x]^c_{R_i}$ and p_a corresponds to $\alpha^c_{R_i}(D^c)$, and Assistant Professional induces a tree optimal to this predictive accuracy.

This result suggests that Assistant Professional calculates the best tree for predictive classification if suitable m and p_a are provided. In other words, this system is the best way in noisy domain. Interestingly, the report on MONK's problems supports this analysis [19]. In this report, it is reported that Assistant gains the best accuracy for deterministic DNF domain and noisy DNF domain. It is also notable that Assistant gets a good result even in non-DNF domain, the Monk's second problem.

8 PRUNING, TRUNCATION AS GENERALIZATION

Overspecialization, or overfitting consists of two factors. One is that the induced results perform worse when applied to future examples, and the other is that the induced result only covers small examples. As to the former factor, we discuss one possible solution in the above section: when we get a good estimator to predictive accuracy, the greedy algorithm calculates solutions optimal to predictive accuracy. However, this solution does not solve the latter prob-

lem: to derive more general formulae which cover as many training samples as possible, such that the number of the formulae is as small as possible. [5]

From the viewpoint of covering, solutions derived by the above greedy algorithms do not always satisfy the condition of *minimal covering* problem. Formally, let us consider a case when the training samples S consist of the union of $[x]_{R_i}(i \in I)$, where R_i denotes the induced rule from training samples. Then, the minimal covering problem is to find a subcover $[x]_{R_j}(j \in J)$, where $J \subseteq I$, which has the property that $|J|$ is a minimum.

Unfortunately, we can not determine whether the number of the induced covering, say k is minimal, since we do not know the cardinality of I, $|I|$.

This problem is also described in our matroid framework. In the above definition, since $A(R_i)$ corresponds to a base of a matroid, the minimal covering problem is to find a subset L of $\cup_{i \in I} 2^{A(R_i)}$. Because $2^{|A(R_i)|}$ is exponential to the number attributes, the computation will be intractable when $|A(R_i)|$ grows large, even if we confine the search space to the space spanned by independent sets. [6]

Since it is well known that the computational complexity of the minimal covering problem is $NP - complete$ [11], the best way is one of the two methods: either we perform the exhaustive search for possible covering or we apply heuristic methods to get approximate solutions.

9 MATROID THEORY AND COMBINATORIAL GEOMETRY

It is well studied that matroid theory is closely related with combinatorial geometry. For example, nine different geometries can be constructed on a five-element set [21]. These geometries can be classified with respect to rank, which is equal to the cardinality of the base. Therefore, the algebraic structure of these geometries corresponds to that of matroids.

[5] This problem is closely related with constructive generalization. However, we do not discuss this problem in this paper. Readers could refer to [14] for further discussion of constructive generalization.

[6] If we deal with the whole attribute, the size of search space will be $2^{|A|}$, where $|A|$ denotes the number of total attribute-value pairs.

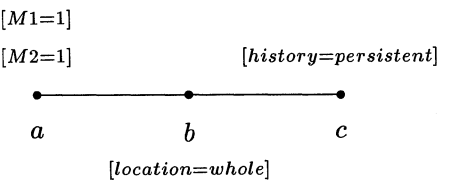
This means that we can also apply the above correspondence to AQ and Pawlak's matroids defined in Section 5. Let $B_{1j}, B_{2j}, \cdots, B_{nj}$ be bases of a AQ matroid which satisfies $[x]_{B_{ij}} = D_j$. Then, $B = B_{1j} \cup B_{2j} \cup \cdots \cup B_{nj}$ is a union of the bases. If we regard a set B as a set of geometric points, then we can construct a geometry G_B, based on the characteristics of these matroids, such as rank. Furthermore, if $C = B_{1j} \cap B_{2j} \cap \cdots \cap B_{nj}$, corresponding to *core*, exists, then this core can be viewed as a constraint on G_B.

For example, let us consider the database shown in Section 2. In this case, B for *m.c.h.* is equal to $\{[loc = whole], [his = per], [M1 = 1], [M2 = 1]\}$. However, $[M1 = 1]$ and $[M2 = 1]$ should be regarded as the same point, because $[x]_{[M2=1]} = [x]_{[M1=1]}$. Thus, B is set to $\{[loc = whole], [his = per], [M1 = 1]\}$, and we can construct a simple geometry on these three points, as shown in Fig. 1(a). Furthermore, we have one constraint, because C is equal to $\{[loc = whole]\}$.

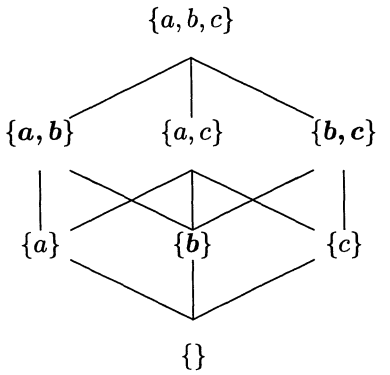
This information on the constraint will be easy to see when a geometry structure is transformed into a geometric lattice shown in Fig.1(b). In this lattice, the bold characters denote partial sets which include C. Thus, those sets make a partial structure in the geometric lattice.

Although the above example has only trivial structure because of rank-2, we have more interesting structure when rank ≥ 3. For example, there are four possibilities when we consider a rank-3 geometry with five points, as shown in Fig.2. In this figure, any three-element subset, say $\{a, b, c\}$, can be a base of each matroid, when no constraint is imposed on these five-point geometries. In the same way as the above example, learning methods can check whether any constraint is available or not.

In summary, rough classification and empirical learning methods can be viewed as a kind of method to construct finite point geometry. Furthermore, both AQ and Pawlak's method do not only calculate a geometric structure, but also calculate constraints on this geometry. It would be our future work to study precisely relations between AQ and Pawlak's matroids and combinatorial geometry.



(a) Geometric Structure



(b) Geometric Lattice

Figure 1 Geometric Structure of the Small Example (Table 1)

10 CONCLUSION

In this paper, we integrate the concepts of matroid theory with those of rough sets, which give us an excellent framework and enables us to understand the differences between AQ, Pawlak's method and ID3 clearly. Using this frame-

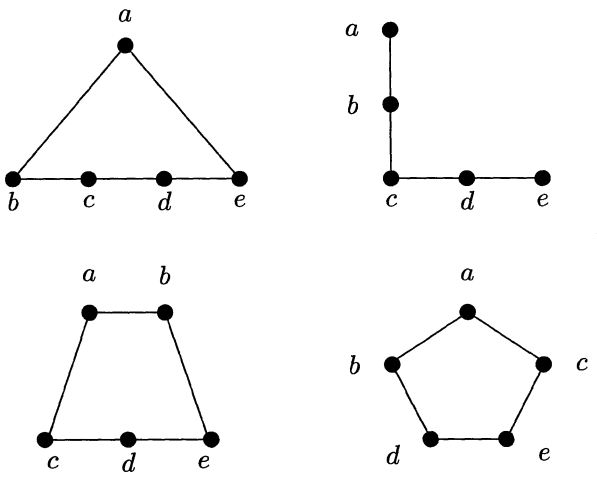


Figure 2 Rank-3 Geometries of Five Points

work, we obtain seven interesting conclusions from our approach. First, while AQ and Pawlak's method are equivalent to the greedy algorithm for finding bases of matroids, ID3 method calculates those of ordered greedoids. Second, AQ matroids are dual to Pawlak matroids. Third, the efficiency of AQ and Pawlak's method depends on the number of attributes, especially, independent variables. Fourth, the induced results are optimal to the training samples with respect to apparent accuracy. Fifth, in addition to the fourth conclusion, since a greedoid of ID3 has weaker structure than the other two methods, ID3 method is the most sensitive to training samples, although its computational complexity is the lowest. Sixth, if we get a well-defined weight function for predictive accuracy, then the solutions derived by the greedy algorithm would be optimal to prediction. Finally, seventh, we can also apply knowledge on combinatorial geometry, because matroid theory is closely related with combinatorial geometry.

Although these results were observed by some experimental results[1, 2, 13, 16], they have not yet been explained by formal theory. We feel that the extension of these approach can be applied to the extension of the above three original methods, such as POSEIDON(AQ16)[1], VPRS[24] and C4[17]. Thus, it would

be our future work to formalize these methods and to analyze the relationship between these existing algorithms by using our framework.

Acknowledgements

This research is supported by Grants-in-Aid for Scientific Research from the Ministry of Education, Science and Culture, Japan.

REFERENCES

[1] Bergadano, F., Matwin, S., Michalski, R.S. and Zhang, J. Learning Two-Tiered Descriptions of Flexible Concepts: The POSEIDON System, *Machine Learning*, **8**, 5-43, 1992.

[2] Breiman, L., Freidman, J., Olshen, R. and Stone, C. *Classification And Regression Trees.* Belmont, CA: Wadsworth International Group, 1984.

[3] Cestnik, B., Kononenko, I., Bratko, I. Assistant 86: A knowledge elicitation tool for sophisticated users. *Proceedings of the Second European Working Session on Learning*, pp.31-45, Sigma Press, 1987.

[4] Cestnik, B., Bratko, I. On Estimating Probabilities in Tree Pruning. *Proceedings of EWSL-91*, 1991.

[5] Garey, M.R. and Johnson, D.S. *Computers and Intractability*, W.H. Freeman and Company, New York, 1979.

[6] Michalski, R.S. A Theory and Methodology of Machine Learning. Michalski, R.S., Carbonell, J.G. and Mitchell, T.M., *Machine Learning - An Artificial Intelligence Approach*, 83-134, Morgan Kaufmann, CA, 1983.

[7] Michalski, R.S., et al. The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains, *Proc. of AAAI-86*, 1041-1045, Morgan Kaufmann, CA, 1986.

[8] Michalski, R.S., and Tecuci, G.(eds) Machine Learning vol.4 - A Multistrategy Approach -, Morgan Kaufmann, CA, 1994.

[9] Pawlak, Z. *Rough Sets*, Kluwer Academic Publishers, Dordrecht, 1991.

[10] Quinlan, J.R. Induction of decision trees, *Machine Learning*, **1**, 81-106, 1986. On Estimating Probabilities in Tree Pruning. *Proceedings of EWSL-91*, 1991.

[11] Garey, M.R. and Johnson, D.S. *Computers and Intractability*, W.H. Freeman and Company, New York, 1979.

[12] Michalski, R.S. A Theory and Methodology of Machine Learning. Michalski, R.S., Carbonell, J.G. and Mitchell, T.M., *Machine Learning - An Artificial Intelligence Approach*, 83-134, Morgan Kaufmann, CA, 1983.

[13] Michalski, R.S., et al. The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains, *Proc. of AAAI-86*, 1041-1045, Morgan Kaufmann, CA, 1986.

[14] Michalski, R.S., and Tecuci, G.(eds) Machine Learning vol.4 - A Multistrategy Approach -, Morgan Kaufmann, CA, 1994.

[15] Pawlak, Z. *Rough Sets*, Kluwer Academic Publishers, Dordrecht, 1991.

[16] Quinlan, J.R. Induction of decision trees, *Machine Learning*, **1**, 81-106, 1986.

[17] Quinlan, J.R. Simplifying Decision Trees. *International Journal of Man-Machine Studies*, **27**, 221-234, 1987.

[18] Quinlan, J.R. and Rivest, R.L. Inferring Decision Trees Using the Minimum Description Length Principle, *Information and Computation*, **80**, 227-248, 1989.

[19] Thrun, S. et al. The MONK's Problems - A Performance Comparison of Different Learning Algorithms. CMU-CS-91-197, 1991.

[20] Welsh, D.J.A. *Matroid Theory*, Academic Press, London, 1976.

[21] White, N.(ed.) *Matroid Applications*, Cambridge University Press, 1991.

[22] Whitney, H. On the abstract properties of linear dependence, *Am. J. Math.*, **57**, 509-533, 1935.

[23] Ziarko, W. The Discovery, Analysis, and Representation of Data Dependencies in Databases, in:*Knowledge Discovery in Database*, Morgan Kaufmann, 1991.

[24] Ziarko, W. Variable Precision Rough Set Model, *Journal of Computer and System Sciences*, **46**, 39-59, 1993.

21

TOPOLOGICAL ROUGH ALGEBRAS

Anita Wasilewska[1]

Department of Computer Science,
State University of New York,
Stony Brook, NY, USA 11794

ABSTRACT

It is known ([15]) that the propositional aspect of rough set theory is adequately captured by the modal system S5. A Kripke model gives the approximation space (A,R) in which well formed formulas are interpreted as rough sets. Banejee and Chakraborty ([1]) introduced a new binary connective in S5, the intended interpretation of which was the notion of rough equality, defined by Pawlak in 1982. They called the resulting Lindenbaum-Tarski like algebra a rough algebra. We show here that their rough algebra is a particular case of a quasi-Boolean algebra (as introduced in [4]). It also leads to a definition of a new classes of algebras, called topological quasi-Boolean algebras[2] and topological rough algebras. We introduce, following Rasiowa and Białynicki-Birula's representation theorem for the quasi-Boolean algebras ([4], [20]), a notion of quasi field of sets and generalize it to a new notion of a *topological quasi field of sets.* We use it to give the representation theorems for the topological quasi-Boolean algebras and topological rough algebras, and hence to provide a mathematical characterization of the rough algebra.

[1]This paper was initiated in November 1993 during the author's discussions with M. Banejee who also visited the Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland. The research was supported by a Fulbright grant no. 93-68818 (1993-1994).

[2]These observations and definition were introduced by A. Wasilewska in November 1993 and hastily reported in [1], [2], and [3].

1 INTRODUCTION

We use here algebraic logic techniques to give a deeper mathematical meaning to the *rough sets theory* in general, and to the notion of *rough equality*, introduced by Pawlak in ([16]), in particular.

It is difficult to establish who was the first to use the algebraic methods. The investigations in logic of Boole himself led to the notion which we now call Boolean algebra, but one of the turning points in the algebraic study of logic was the introduction by Lindenbaum,[3] and in a slight different form by Tarski (in [23]) of the method of treating formulas, or equivalence classes of formulas as elements of an abstract algebra, called now the *Lindenbaum-Tarski algebra.*

The main use of the Lindenbaum-Tarski algebra is to show the correspondance between logic and abstract algebras. The algebra corresponding to classical logic is, of course, Boolean algebra. Other non-classical logic algebraic studies were initiated by Stone (in [21]), Tarski (in [24]), and McKinsey and Tarski (in [14]), followed by Henkin ([5]) and Rasiowa and Sikorski ([17]). The investigations of Tarski, McKinsey, Henkin, and Rasiowa led to what is now called algebraic models for intuitionistic, modal S4 and S5 logics, as opposed to Kripke models which were invented some 20 years later and applied to the intuitionistic and a variety of modal logics in [8].

Boolean and other algebras are fairly abstract structures. Their deeper mathematical meaning is established by a proper *representation theorem.* The existence of the representation theorem is always the first question one asks about a newly created algebra. The first representation theorem for Boolean algebras was established by Stone (in [22]) and stated the following:

Every Boolean algebra A is isomorphic to a field of sets. More exactly, A is isomorphic with a field of subsets of its Stone space S, which is a compact, totally disconnected Hausdorf space.

The representation theorem, hence establishes a relationship between logic, algebra, set theory and topology.

[3] A. Lindenbaum was a prominent Polish mathematician who was killed by the Nazis during the Second World War, and whose various results were not published. It was McKinsey, a close collaborator of Tarski, who first (in [13]) called the method of treating the set of all formulas of propositional logic "an unpublished method of Lindenbaum explained to me by Professor Tarski".

In our work we use the algebraic logic techniques to link rough set theory, logic, abstract algebras and topology. In particular, we show that the notion of *rough equality* of sets leads, via logic and a Lindenbaum-Tarski like construction of an algebra of formulas, to a definition of new classes of algebras, called here *topological quasi-Boolean algebras* and *topological rough algebras.* We provide here the representation theorems for these new classes of algebras. In order to do so we proceed as follows:

In section 2 we give a short overview of the work by Banerjee and Chakraborty ([1]) leading to their definition of a *rough algebra.* Then we show that their rough algebra is a particular case of a quasi-Boolean algebra. The quasi-Boolean algebras are a slight generalization of de Morgan lattices, which have been introduced and examined in [9]. The quasi-Boolean algebras do not define a logic. They are a part of the theory of quasi-pseudo-Boolean algebras that are models for a constructive logic with strong negation ([10], [12], [26], [25]).

In section 3 we introduce, justify, and discuss the notions of a topological quasi-Boolean algebra and of a topological rough algebra. We show that Białynicki-Birula and Rasiowa's proof of a representation theorem for quasi-Boolean algebras ([4]) can be generalized, via MacKinsey and Tarski's proof for the topological Boolean algebras ([14]), to our case. From that we get as a corollary the following mathematical characterization of the rough algebra.

The rough algebra of Banerjee and Chakraborty is isomorphic with a topological rough-field of sets.

2 ROUGH ALGEBRA AS A TOPOLOGICAL QUASI-BOOLEAN ALGEBRA

Our work is based on Pawlak's definition of a *rough equality* ([16]), the work of Orlowska ([15]) and of Banerjee and Chakraborty ([1]).

To make our paper self contained we review here some here basic definitions and explain the main points of Banerjee and Chakraborty's construction of the rough algebra.

Approximation space Let U be a non-empty set called a universe, and let R be an equivalence relation on U. The triple $(U, \emptyset, R)$ is called an approximation space.

Lower, upper approximations Let $(U, \emptyset, R)$ and $A \subset U$. Denote by $[u]$ the equivalence class of R. The sets

$$IA = \bigcup\{[u] \in A/R : \ [u] \subset A\},$$

$$CA = \bigcup\{[u] \in A/R : \ [u] \cap A \neq \emptyset\}$$

are called *lower* and *upper* approximations of A, respectively. We use here a topological notation for the lower and upper approximation because of their topological interpretation and future considerations.

Rough equality Given an approximation space $(U, \emptyset, R)$ and any $A, B \subset U$. We say that the sets A and B are *roughly equal* and denote it by $A \sim_R B$ if and only if $IA = IB$ and $CA = CB$.

Boolean algebra An abstract algebra $(A,\ 1,\ 0,\ \Rightarrow,\ \cap,\ \cup, \neg)$ is said to be a Boolean algebra if it is a distributive lattice and every element $a \in A$ has a complement $\neg a \in A$.

Topological Boolean algebra By a topological Boolean algebra we mean an abstract algebra $(A,\ 1,\ 0,\ \Rightarrow,\ \cap,\ \cup, \neg,\ I)$ where $(A,\ 1,\ 0,\ \Rightarrow,\ \cap,\ \cup, \neg)$ is a Boolean algebra and, moreover, the following conditions hold: $I(a \cap b) = Ia \cap Ib$, $Ia \cap a = Ia$, $IIa = Ia$, and $I1 = 1$, for any $a, b \in A$.

The element Ia is called a *interior of a*. The element $\neg I \neg a$ is called a *closure of a* and will be denoted by Ca. Thus the operations I and C are such that $Ca = \neg I \neg a$ and $Ia = \neg C \neg a$. The element a is said to be *open* (*closed*) if $a = Ia$ $(a = Ca)$.

Orlowska has shown in [15] that propositional aspects of rough set theory are adequately captured by the modal system S5. In this case a Kripke model gives the approximation space $(A, \emptyset, R)$ in which the well formed formulas are interpreted as *rough sets.*

Following Orlowska result, Banerjee and Chakraborty introduced in [1] a new binary connective $\sim$ in S5, the intended interpretation of which is the notion of the *rough equality.* I.e., they added to the standard set $\{\cup, \cap, \rightarrow, \Leftrightarrow, \neg, \Box, \Diamond\}$ of propositional modal connectives a new binary connective $\sim$ defined in terms

of standard connectives as follows: for any formulas A, B (of the modal S5 language), we write $A \sim B$ for the formula $((\Box A \Leftrightarrow \Box B) \cap (\Diamond A \Leftrightarrow \Diamond B))$. In the next step they have used this new connective to define a construction similar to the construction of Lindenbaum-Tarski algebra on the set of all formulas of $S5$ with additional connective $\sim$. Before describing their construction leading to the definition of the rough algebra, we include below a description of a standard construction of a Lindenbaum-Tarski algebra for a given logic.

Lindenbaum-Tarski construction Given a propositional logic with a set $\mathcal{F}$ of formulas. We define first two binary relations $\leq$ and $\approx$ in the algebra $\mathcal{F}$ of formulas of the given logic as follows. For any $A, B \in F$,

$A \leq B$ if and only if $\vdash (A \Rightarrow B)$, and

$A \approx B$ if and only if $\vdash (A \Rightarrow B)$ and $\vdash (B \Rightarrow A)$.

Then we use the set of axioms and rules of inference of the given logic to prove all facts listed below.

The relation $\leq$ is a quasi-ordering in F.

The relation $\approx$ is an equivalence relation in F. We denote the equivalence class containing a formula A by $[A]$.

The quasi ordering $\leq$ on F induces an ordering relation on $F/\approx$ defined as follows: $[A] \leq [B]$ if and only if $A \leq B$.

The equivalence relation $\approx$ on F is a congruence with respect to all logical connectives.

The resulting algebra with universe $F/\approx$ is called a Lindenbaum-Tarski algebra.

EXAMPLE 1 *The Lindenbaum-Tarski algebra for* classical propositional logic *with the set of connectives* $\{\cup, \cap, \Rightarrow, \neg\}$ *is the following.*

$$\mathcal{LT} = (F/\approx, \cup, \cap, \Rightarrow, \neg),$$

where the operations $\cup, \cap, \Rightarrow$ *and* $\neg$ *are determined by the congruence relation* $\approx$ *i.e.* $[A] \cup [B] = [(A \cup B)]$, $[A] \cap [B] = [(A \cap B)]$, $[A] \Rightarrow [B] = [(A \Rightarrow B)]$, $\neg[A] = [\neg A]$.

We prove, in this case (see [17]) that the Lindenbaum-Tarski algebra is a Boolean algebra with a unit element V. Moreover, for any formula A, $\vdash A$ if and only if $[A] = V$.

EXAMPLE 2 *The Lindenbaum-Tarski algebra for modal logic $S4$ or $S5$ with the set of connectives $\{\cup, \cap, \Rightarrow, \neg, \Box, \Diamond\}$ is the following.*

$$\mathcal{LT} = (F/\approx, \cup, \cap, \Rightarrow, \neg, I, C),$$

where the operations $\cup, \cap, \Rightarrow$ and $\neg, I, C$ are determined by the congruence relation $\approx$ i.e. $[A] \cup [B] = [(A \cup B)]$, $[A] \cap [B] = [(A \cap B)]$, $[A] \Rightarrow [B] = [(A \Rightarrow B)]$, $\neg[A] = [\neg A], IA = [\Box A]$, and $CA = [\Diamond A]$.

In the case of modal logic $S4$ the Lindenbaum-Tarski algebra (see [13], [14], [17]) is *a topological Boolean algebra* and in the case of $S5$ it is topological Boolean algebra such that every open element is closed and every closed element is open. Moreover, in both cases, for any formula A, $\vdash A$ if and only if $[A] = V$.

Banerjee, Chakraborty construction We define a new binary relation $\approx$ on the set F of formulas of the modal S5 logic as follows. For any $A, B \in F$,

$A \approx B$ if and only if $A \sim B$, i.e.

$A \approx B$ if and only if $\vdash ((\Box A \Leftrightarrow \Box B) \cap (\Diamond A \Leftrightarrow \Diamond B))$.

We prove that the above relation $\approx$, corresponding to the notion of rough equality is *an equivalence relation* on the set F of formulas of $S5$.

We define a binary relation $\leq$ on $F/\approx$ as follows.
$[A] \leq [B]$ if and only if $\vdash ((\Box A \Rightarrow \Box B) \cap (\Diamond A \Rightarrow \Diamond B))$.

We prove that $\leq$ is an order relation on $F/\approx$ with the greatest element $1 = [A]$, for any formula A, such that $\vdash A$, and with the least element $0 = [B]$, such that $\vdash \neg B$.

We prove that $\approx$ *is a congruence relation* with respect to the logical connectives $\neg, \Box, \Diamond$, but *is not a congruence relation* with respect to $\Rightarrow, \cap$ and $\cup$.

We introduce two new operations $\sqcup$ and $\sqcap$ in $F/\approx$ as follows
$[A] \sqcap [B] = [(A \cap B) \cup (A \cap \Diamond A \cap \Diamond B \cap \neg\Diamond(A \cap B))]$,
$[A] \sqcup [B] = [(A \cup B) \cap (A \cup \Box A \cup \Box B \cup \neg\Box(A \cup B))]$.

We call the resulting structure a *rough algebra.*

The formal definition of the rough algebra is hence the following.

Rough algebra An abstract algebra

$$\mathcal{R} = (F/\approx,\ \sqcup,\ \sqcap, \neg,\ I,\ C,\ 0,\ 1),$$

such that the operations $\sqcup$, $\sqcap$ are defined above and the operations $\neg, I, C$ are induced, as in the Lindenbaum-Tarski algebra, by the relation $\approx$, i.e. $\neg[A] = [\neg A]$, $IA = [\Box A]$, and $CA = [\Diamond A]$ is called a rough algebra.

Properties of the rough algebra In [3] many important properties of the rough algebra were were proved. They were also reported in [1]. We cite here only those which are relevant to our future investigations.

P1 $(F/\approx,\ \leq, \sqcup,\ \sqcap, 0,\ 1)$ is a distributive lattice with 0 and 1.

P2 For any $[A], [B] \in F/\approx$, $\neg([A] \sqcup [B]) = (\neg[A] \sqcap \neg[B])$,

P3 For any $[A] \in F/\approx$, $\neg\neg[A] = [A]$.

P4 The rough algebra is not a Boolean algebra, i.e. there is a formula A of a modal logic $S5$, such that $\neg[A] \sqcap [A] \neq 0$ and $\neg[A] \sqcup [A] \neq 1$.

P5 For any $[A], [B] \in F/\approx$, $I([A] \sqcap [B]) = (I[A] \sqcap I[B])$, $I[A] \leq [A]$, $II[A] = I[A]$, $I1 = 1, and\ CI[A] = I[A]$, where $C[A] = \neg I \neg [A]$.

2.1 Questions and observations

The above, and other properties of the rough algebra proved in [3] lead to some natural questions and observations.

Q1 By the property **P4**, the rough algebra's complement operation ($\neg$) is not a Boolean complement. Let's call it *a rough complement.* We can see that the rough complement is pretty close to the Boolean complement because the other de Morgan law $\neg([A] \sqcap [B]) = (\neg[A] \sqcap \neg[B])$ holds in the rough algebra, as well as the very Boolean laws $\neg 1 = 0$ and $\neg 1 = 0$. So what kind of a complement is the rough complement?

Q2 The rough algebra *is not*, by **P4**, a Boolean algebra, so which kind of algebra is it?

Q3 Has such an algebra been discovered and investigated before?

OBSERVATION 1 A complement operation with similar properties to the *rough complement* was introduced in 1935 by Moisil and lead to a definition of a notion of *de Morgan Lattices*. De Morgan lattices are distributive lattices satisfying the conditions **P2** and **P3**. They were investigated in [9], [7], [6].

OBSERVATION 2 In 1957 Białynicki-Birula and Rasiowa have used the de Morgan lattices to introduce a notion of *a quasi-Boolean algebra*. They defined (in [4]) the quasi-Boolean algebras as de Morgan lattices with *unit* element 1.

The formal definition of the quasi-Boolean algebras is the following.

Quasi-Boolean algebra (Białynicki-Birula, Rasiowa, 1957) An abstract algebra $\mathcal{A} = (A, \cup, \cap, \sim, 1)$ is called a *quasi-Boolean algebra* if $(A, \cup, \cap, 1)$ is a distributive lattice with unit element 1 and for any $a, b \in A$, $\sim (a \cup b) = (\sim a \cap \sim b)$ and $\sim\sim a = a$.

One can easily prove that in every quasi-Boolean algebra the *zero (0)* element exists. From that and properties **P1 - P4** we get the following fact.

FACT 1 *The rough algebra is not a Boolean algebra, but is a quasi-Boolean algebra.*

OBSERVATION 3 The property **P5** tells that the operations I and C of the rough algebra $(F/\approx, \sqcup, \sqcap, \neg, I, C, 0, 1)$ fulfill the axioms of the topological Boolean algebra.

DEFINITION 1 (Topological quasi-Boolean algebra)

An algebra $(A, \cap, \cup, \sim, 1, I)$ *is called* a topological quasi-Boolean algebra *if* $(A, \cup, \cap, \sim, 1)$ *is a quasi-Boolean algebra and for any* $a, b \in A$, $I(a \cap b) = Ia \cap Ib$, $Ia \cap a = Ia$, $IIa = Ia$, *and* $I1 = 1$.

The element Ia is called a *quasi-interior of a*. The element $\sim I \sim a$ is called *quasi-closure of a*. It allows us to define in A an unary operation C such

that $Ca = \sim I \sim a$. We can hence represent the topological quasi-Boolean algebra as an algebra $(A, \cap, \cup, \sim, I, C, 0, 1)$ *similar* to the rough algebra $(F/\approx, \sqcup, \sqcap, \neg, I, C, 0, 1)$. From **P4** we immediately get the following.

FACT 2 *A rough algebra* $\mathcal{R} = (F/\approx, \sqcup, \sqcap, \neg, I, C, 0, 1)$ *is a* topological quasi-Boolean algebra.

3 TOPOLOGICAL ROUGH ALGEBRAS

The property **P5** of the rough algebra tells us also that the operations I and C fulfill an additional property: for any $[A] \in F/\approx$, $CI[A] = I[A]$. This justifies the following definition.

DEFINITION 2 (Topological rough algebra) *A topological quasi-Boolean algebra* $(A, \cap, \cup, \sim, I, C, 0, 1)$ *such that for any* $a \in A$, $CIa = Ia$, *is called a* topological rough algebra.

Note that the class of all topological quasi-Boolean algebras, and the class of all topological rough algebras are *equationally definable.*

Directly from above we get the following answer to the question **Q3**.

FACT 3 *The rough algebra* $(F/\approx, \sqcup, \sqcap, \neg, I, C, 0, 1)$ *is a* topological rough- algebra.

As we have said in the Introduction, one of the first questions one asks about a new algebra, or classes of algebras, is the existence and form of the representation theorem. This is a question about a deeper mathematical meaning of the newly created abstract algebras. We are going to introduce here all notions and steps which lead to the understanding of meaning, complexity and beauty of the representation theorem (and of its proof). We are not including the proof here. It is quite long and technical and will be published separately. Since the first question (**Q1**) we have asked here about the rough algebra was to provide some characterization of its complement operation, which we have called a *rough complement*, we will start with the mathematical characterization of this notion.

The OBSERVATION 1 and FACT 1 provided already some characterization of the rough complement, namely that it is a *quasi-Boolean complement.* We use and extend here the 1957 work of Białynicki-Birula and Rasiowa from [4] to characterize further the rough complement, the topological quasi-Boolean algebras, and the rough algebras.

Let A be a non-empty set. We define after [4] the following notion.

Involution Any mapping $g : A \longrightarrow A$ such that for all $a \in A$, $g(g(a)) = a$ is called an *involution.* Clearly, every involution is a one-one mapping from A onto A.

Let $Q(A)$ be a non-empty class of subsets of A, containing A and closed under set-theoretical union and intersection, and under the operation $\sim$ defined as follows: *for any* $X \in Q(A)$, $\sim X = A - g(X)$. It is proved in [4] that $(Q(A), A, \cup, \cap, \sim)$ is an example of a quasi-Boolean algebra. The quasi-Boolean algebra is a particular case (when the topology is given by the identity operation, i.e. when for any $a \in A$, $Ia = a$) of the topological quasi-Boolean algebra.

This justifies the following definition.

DEFINITION 3 (Rough complementation) *Given a non empty set A and an involution g on A. We call the operation $\sim X = A - g(X)$ a* rough complementation of sets.

DEFINITION 4 (Quasi-field of sets) *Given a non empty set A and the rough complementation $\sim$ in A, we call a structure $(Q(A), A, \cup, \cap, \sim)$ a* quasi-field of subsets of A.

The representation theorem for the quasi-Boolean algebras says that quasi-fields of sets are typical examples of quasi-Boolean algebras (see [4], [20]), i.e. that the following holds.

Representation theorem *Every quasi-Boolean algebra is isomorphic to a quasi-field of certain open subsets of a topological, compact T_0 space.*

We follow here basic notions of the algebraic logic, where the topological space is defined as follows.

Topological space A set A is said to be a *topological space* if with every $X \subset A$ there is associated a set $IX \subset A$ such that the following conditions are satisfied: for any $X, Y \subset A$, $I(X \cap Y) = IX \cap IY$, $IX \subset X$, $IIX = X$, $IA = A$. The operation I is called the interior operation. The topological space is often written as (A, I).

Given a topological space (A, I), then $(Q(A), A, \cup, \cap, \sim, I)$ where $\sim$ is rough complement is a topological quasi-Boolean algebra. Clearly, every subalgebra of this algebra is also a topological quasi-Boolean algebra. This justifies the following definitions.

DEFINITION 5 (Topological quasi-field of sets) *Given a topological space* (A, I) *and a quasi-field of sets* $(Q(A), A, \cup, \cap, \sim)$. *We define the* closure operation C *as* $CX = \sim I \sim X$ *and call a structure* $(Q(A), A, \cup, \cap, \sim, I, C)$ *a* topological quasi-field of sets *or, more precisely, a topological quasi-field of subsets of* A.

DEFINITION 6 (Topological rough-field of sets)

A topological quasi-field of sets $(Q(A), A, \cup, \cap, \sim, I, C)$ *is called a* topological rough-field of sets *if additionally, for any set* $X \in Q(A)$, $CIA = IA$.

The most important notion leading to the proof of the representation theorem for any algebra are the notions of a filter and ideal (see [21], [20]).

Filter A non-empty set ∇ of elements of universe A of an algebra with two binary operations $\cap$ and $\cup$ is said to be *a filter* in A provided that, for any elements $a, b \in A$, $a \cap b \in \nabla$ if and only if $a \in \nabla$ and $b \in \nabla$.

Ideal A non-empty set Δ of elements of A is said to be *an ideal* in A provided that, for any elements $a, b \in A$, $a \cup b \in \Delta$ if and only if $a \in \Delta$ and $b \in \Delta$.

A filter (ideal) is called *maximal* in A if it is proper and is not any proper subset of a proper filter (ideal) in A.

We adopt Rasiowa's definition of a I-filter ([20]) to our purposes, i.e. we adopt the following definition.

DEFINITION 7 (Rough filter) *A filter* ∇ *(ideal* Δ*) in a topological quasi-Boolean algebra* $(A, \cap, \cup, \sim, I, C, 0, 1)$ *is called a* rough filter *(rough ideal) provided*

$$a \in \nabla \quad \textit{implies that} \quad Ia \in \nabla \quad \textit{for every} \quad a \in A.$$

Then we show that all major properties of Rasiowa's I-filters hold for the rough-filters, in particular we show the following.

FACT 4 (Maximal rough filter) *For every element* $a \neq 0$ *in* A *there exists a maximal rough filter* ∇ *in* A*, such that* $a \in \nabla$.

Given a topological quasi-Boolean algebra $(A, \cap, \cup, \sim, I, C, 0, 1)$, let now put, for any set $S \subset A$,

$$\widetilde{S} = \{\sim a \in A : \quad a \in S\}.$$

We prove the following duality property.

Duality If ∇ is a rough maximal filter in a topological quasi-Boolean algebra, the the set $\widetilde{\Delta}$ is a rough maximal ideal.

Then we combine the techniques of Stone ([21]), Rasiowa ([19]) and Białynicki-Birula and Rasiowa ([4]) and prove that the following holds.

THEOREM 1 (Representation theorem) *For every topological quasi-Boolean algebra (topological rough algebra)* $\mathcal{A}$ *there exists a monomorphism* h *from* $\mathcal{A}$ *into a topological quasi-field (rough-field) of sets of a topological space* A.

FACT 5 *The rough algebra* $\mathcal{R} = (F/\approx, \sqcup, \sqcap, \neg, I, C, 0, 1)$ *is isomorphic with a topological rough-field of sets.*

4 SUMMARY

We have shown here that Pawlak's notion of a rough equality of sets leads via work of Banerjee and Chakraborty to the definition of two classes of abstract algebras. These algebras generalize McKinsey and Tarski notion of a closure algebra (named here after [11], [19], [20], a topological Boolean algebra) and Białynicki-Birula and Rasiowa notion of a quasi-Boolean algebra. We have also shown that it is possible to formulate and prove proper representation theorems for those classes of algebras. We have also obtained, as a particular case of those general results, a deeper mathematical characterization of the notion of rough equality.

REFERENCES

[1] M. Banerjee, M.K. Chakraborty, "Rough Consequence and Rough Algebra", Rough Sets, Fuzzy Sets and Knowledge Discovery, Of the International Workshop on Procedings of Rough Sets and Knowledge Discovery, (RSKD'93), Banf, Alberta, Canada, 1993, W.P. Ziarko, (Ed.), Springer-Verlag, London, (1994), pp. 196-207.

[2] M. Banerjee, M.K. Chakraborty, "Rough algebra", Bull. Polish Acad. Sc. (Math.), vol.41, No.4, 1993, pp. 299 - 297.

[3] M. Banerjee, "A Cathegorical Approach to the Algebra and Logic of the indiscernible", Ph.D dissertation, Mathematics Department, University of Calcutta, India.

[4] Białynicki-Birula, A., Rasiowa, H., "On the representation of quasi-Boolean algebras", Bull. Ac. Pol. Sci. Cl. III, 5 (1957), pp. 259-261.

[5] Henkin, L., "An algebraic characterization of Quantifiers", Fundamenta Mathematicae 37 (1950), pp. 63-74.

[6] Henkin, L., "A class of non-normal models for classical sentential logic", The Journal of Symbolic Logic 28 (1963), p. 300.

[7] Kalman, J. A., "Lattices with involution", Trans.Amer. Math. Soc. 87 (1958), pp. 485 - 491.

[8] Kripke, S., "Semantics analysis of intuitionistic logic", Proc. of the Eight Logic Colloqium, Oxford 1963, edited by J.N. Crossley & M.N.E. Dummet, pp. 92-130, North Holland Publishing C. (1965).

[9] Moisil, G. C., "Recherches sur l'algebre de la logique", Annales Sc. de l'Univerite de Jassy 22 (1935), pp. 1 -117.

[10] Nelson, D., "Constructible falsity", The Journal of Symbolic Logic 14 (1949), pp. 16-26.

[11] Nöbeling, G., "Grundlagen der analitishen Topologie", Berlin. Götingen, Heilderberg, 1954.

[12] Markov, A.A., " Konstriktivnaja logika", Usp. Mat. Nauk 5 (1950), pp. 187 -188.

[13] McKinsey, J. C.C., "A solution of the decision problem for the Lewis systems S.2 and S.4 with an application to topology", The Journal of Symbolic Logic 6 (1941), pp. 117- 188.

[14] McKinsey, J. C.C., Tarski, A., "The algebra of topology", Annals of Mathematics 45 (1944), pp. 141-191.

[15] E. Orłowska, "Semantics of vague concepts", In (Dorn,G., Weingartner, P. (eds)) Foundations of Logic and Linguistics, Selected Contributions to the 7th International Congress of Logic, Methodology and Philosophy of Science, Saltzburg 1983, Plenum Press, pp. 465-482.

[16] Z. Pawlak, "Rough Sets", Int. J. Comp. Inf. Sci., 11 (1982), pp. 341- 356.

[17] Rasiowa, H., Sikorski, R., "A proof of completeness theorem of Gódel", Fundamenta Mathematicae 37 (1950), pp. 193-200.

[18] Rasiowa, H., "Algebraic treatement of the functional calculi of Heyting and Lewis", Fundamenta Mathematicae, 38 (1951), pp. 99 -126.

[19] Rasiowa, H., Sikorski, R., "The Mathematics of Metamathematics", PWN, Warszawa, 1963.

[20] Rasiowa, H., "An Algebraic Approach to Non-Classical Logics", Studies in Logic and the Foundations of Mathematics, Volume 78, North-Holland Publishing Company, Amsterdam, London - PWN, Warsaw, (1974).

[21] Stone, M. H., "Boolean algebras and their relation to topology", Proc. Nat. Ac. Sci. 20 (1934), pp. 197-202.

[22] Stone, M. H., "Topological representation of distributive lattices and Brouwerian logics", Čas.Mat. Fys. 67 (1937), pp. 1 -25.

[23] Tarski, A., "Grundzüge des Syntemenkalküls". Ester Teil. Fundamenta Mathematicae 25 (1935), pp. 503-526.

[24] Tarski, A., "Der Aussagencalcul und die Topologie", Fundamenta Mathematicae 31 (1938), pp. 103-134.

[25] Thomason , R. H., "A semantical study of constructible falsity", Zeitschr. für Math. Logik und Grundl. der Math. 15 (1969), pp. 247-257.

[26] Vorobiev, N.N., "Konstructivnoje isčislenie vyskasivanij s silnym otrizaniem", Dokl. Akad. Nauk SSSR 85 (1952), pp. 456-468.

INDEX