Rough Set Based Rule Evaluations and Their Applications

by

Jiye Li

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2007

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Jiye Li

Abstract

Knowledge discovery is an important process in data analysis, data mining and machine learning. Typically knowledge is presented in the form of rules. However, knowledge discovery systems often generate a huge amount of rules. One of the challenges we face is how to automatically discover interesting and meaningful knowledge from such discovered rules. It is infeasible for human beings to select important and interesting rules manually. How to provide a measure to evaluate the qualities of rules in order to facilitate the understanding of data mining results becomes our focus. In this thesis, we present a series of rule evaluation techniques for the purpose of facilitating the knowledge understanding process. These evaluation techniques help not only to reduce the number of rules, but also to extract higher quality rules. Empirical studies on both artificial data sets and real world data sets demonstrate how such techniques can contribute to practical systems such as ones for medical diagnosis and web personalization.

In the first part of this thesis, we discuss several rule evaluation techniques that are proposed towards rule postprocessing. We show how properly defined rule templates can be used as a rule evaluation approach. We propose two rough set based measures, a Rule Importance Measure, and a Rules-As-Attributes Measure, to rank the important and interesting rules. In the second part of this thesis, we show how data preprocessing can help with rule evaluation. Because well preprocessed data is essential for important rule generation, we propose a new approach for processing missing attribute values for enhancing the generated rules. In the third part of this thesis, a rough set based rule evaluation system is demonstrated to show the effectiveness of the measures proposed in this thesis. Furthermore, a new user-centric web personalization system is used as a case study to demonstrate how the proposed evaluation measures can be used in an actual application.

Acknowledgements

I am truly indebted to many people who have contributed to this thesis. First of all, I would like to thank my supervisor Dr. Nick Cercone, who has been the most influential person during the past six and a half years, who has not only supported this work financially, but also given me many insightful advice both on academics and otherwise. My co-supervisor Dr. Robin Cohen has spent considerable time and effort on shaping the final version of this thesis. Dr. Yiyu Yao provided a critical review and has shed light on many future research topics. My thanks also go to Dr. Andrew Wong, Dr. Mohamed Kamel and Dr. Don Cowan, who kindly served on my thesis committee, and Dr. Dale Schuurmans on the thesis proposal committee.

I would like to thank Dr. Guiran Chang from Northeastern University in China, who encouraged me to pursue my graduate studies abroad. The University of Waterloo has provided a great study environment and international atmosphere which has broadened my worldview. The Faculty of Computer Science at Dalhousie University was a source of a cozy and friendly environment during my visit. I received warm friendships and advice from professors, staff members, and students from both University of Waterloo and Dalhousie University. I am sincerely thankful to Dr. Daniel Berry, Dr. Mechelle Gittens, Jane Prime, Margaret Towell, Wendy Rush and Ursula Thoene from University of Waterloo; Dr. Michael Shepherd, Dr. Raza Abidi and Dr. Malcolm Heywood at Dalhousie University. Thank Yingwei Wang, Jiaochao Han, Zhenmei Gu, Vlado Keselj, Aijun An, Jimmy Huang, Xiaohua Tony Hu, Lei Chen, Fuchun Peng, Jinbo Xu and Jenny Liu for their academic advice whenever I needed. During my internship at HP Labs in the Summer 2006, I have also received help and support from the members in the Decision Technology Department, especially from my manager Shailendra Jain, my mentors Dirk Beyer, Rajan Lukose and Jing Zhou, and other interns and friends, Linda, Ernesto, Filippo, Kiran, Yathi, Norman and Minha.

I have received significant amount of friendships during the past six and a half years. I am grateful for all of you that make my life colorful. My thanks go to Ben, Gabriel, Kim and Matt, Jose, Silvia and Ning, Andrei, Hua Wei and Feng, Xinyi, Lijie, Jingwei, Meng, Dennis, Heng Yu, Amir, Stu, Leon, Tony, Wei and Bing, Vincent Park, Sunny, Laurel and Michael from University of Waterloo; Xiaofen, Daan, Lalita, Hathai, Anand, Vivek, Tee,

Carrie and Jason, and Bin Tang from Dalhousie University. Thank the Health Informatics Graduate Students at Dalhousie University (2003, 2004) for being supportive for my first teaching experience. Last but not least, I would like to thank Claude-Guy Quimper for his encouragement and discussions during this thesis work.

This thesis is dedicated to my parents Professor Guangtian Li and Mrs. Xiuling Fu, my sister Jihua and her husband Dezhong, for always being there for me.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Thesis Statement

In order to automatically discover meaningful and important knowledge from huge amounts of rules generated in a knowledge discovery system, we propose several rule evaluation measures to facilitate the knowledge understanding process. Automatic rule evaluation measures are proposed to extract and rank important knowledge. Empirical studies on artificial data sets and real world data sets demonstrate how such techniques can contribute to practical systems such as ones for medical diagnosis and web personalization systems.

1.2 Motivation

Knowledge discovery in databases is a process of discovering previously unknown, valid, novel, potentially useful and understandable patterns in large data sets [23]. Data mining is one of the activities in this interactive process. Data mining encompasses many different techniques and algorithms, including clustering, classification, association rule algorithms and so on.

A sample knowledge discovery system is illustrated in Figure 1.1. The system first performs data preprocessing, a step in which the inconsistent data and the data instances containing missing attribute values are processed. Then the generation of rules is con-

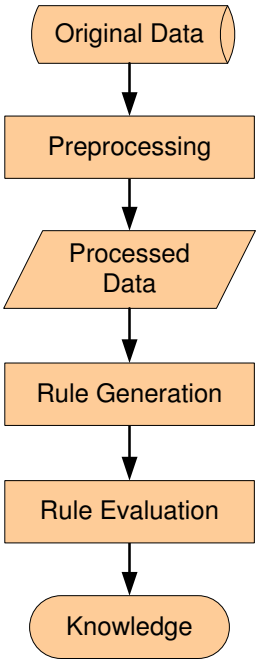

Figure 1.1: A Knowledge Discovery System

ducted on the processed data by certain algorithms. After the rules are generated, rule evaluations are performed. Redundant or not important rules are removed, and useful rules are presented as knowledge as the output of the system. Rule generation is one of the important processes in the knowledge discovery system. For example, a rule such as "Japanese cars with manual transmission and light weight usually have higher mileage", can be learned by a classification algorithm from a data set of cars which contain mileage of the cars and features such as the manufacturer, the model, the transmission, the weight and so on [39]. Such rules are used for making predictions.

A challenging problem in rule generation is that an extensive number of rules are extracted by data mining algorithms over large data sets, and it is infeasible for human beings to select important, useful, and interesting rules manually. How to develop measures to automatically extract and evaluate interesting, relevant, and novel rules becomes an urgent and practical topic in this area.

Many existing methods such as rule interestingness measures and rule quality measures

from statistics and information theory areas were reported in [17, 35, 85]. Comparisons of different measures are reported for general purpose applications. We say a measure is a subjective measure if it is defined based on a domain expert's opinions towards the particular application [35]. A measure is an objective measure if it measures the data itself without any predefined opinions. Subjective measures that use real human evaluators are the optimal measure to evaluate rules, although they are sometimes infeasible and expensive, because they may require humans to look at a large number of rules. Previous studies on rule evaluations focus mostly on objective measures [35], which do not contain any knowledge from the domain of the data. Therefore such objective measures may not sufficiently evaluate whether a rule is indeed interesting for a certain domain.

In this thesis, we study and propose several rule evaluation measures for the purpose of facilitating the knowledge understanding process. Our motivation is to design automatic rule evaluation measures that can bring both domain related knowledge (such as what are the important attributes and what are the expected results) and the objective measures together into the rule evaluations. Such measures are proposed to help to extract and rank important rules from a large number of rules generated by a learning algorithm.

1.3 Objectives

We elaborate on our goals in this thesis as follows.

- Domain-based Rule Templates. As explained earlier, rule evaluation measures can be categorized into two groups, objective measures and subjective measures. The objective measures on evaluating rules provide a straightforward way to evaluate. However, they cannot provide a sufficient evaluation for specific application purposes. Consider association rules generation [3] as an example. An association rules algorithm is used to extract item-item relationships among large transaction data sets. For example, in a market basket analysis, by analyzing transaction records from the market, we could use association rule algorithms to discover frequently purchased items such as bread and milk, because when customers buy bread, they will probably buy milk. This type of item-item associations can be used in the market analysis to increase the amount of milk sold in the market. *Support* and *confidence*

are the two most commonly used measures in this algorithm for itemset generation and rule generation [3]. We call the left hand side of a rule the antecedent, and the right hand side of a rule the consequent. Support measures how often the antecedent and the consequent of a rule appear together in the transaction set. Confidence gives a ratio of the number of transactions where the antecedent and the consequent appear together to the number of transactions where the antecedent appears. However, there are usually a few other measures applied in addition to support and confidence in rule generation. This is because, even with higher support and higher confidence values to reflect an increased interest in the rules, still a large number of rules are generated. Extra measures are necessary to help reduce the number of rules, and at the same time extract only interesting and important rules.

In the real world, deciding what kinds of rules are interesting is quite subjective. Domain experts are by far the best suited to evaluate whether a rule is interesting or not. For some domain experts, they may be interested in a certain set of rules, whereas the same set of rules may not be as interesting to other domain experts. For example, in the market-basket analysis, rules showing the relations between beer and diapers might be interesting to market business people; nutrition doctors may be more interested in finding rules containing the relations between people who buy nutritional items, such as vitamins and fruits. With the help of a domain expert, not only can interesting rules be found and recommended to users, but also the computational cost for rule generations can be largely reduced by combining the domain related background into the rule generation process.

We study rule templates, patterns to define items that appear both in the antecedent and consequent of association rules, as a subjective rule evaluation measure in this thesis. The concept of "Rule Template" was first presented by Klemettinen *et al.* in 1994 [45]. Rule templates describe patterns for those items that appear both in the antecedent and in the consequent of association rules [45]. A rule matches the defined template if this rule is an instance of the template. We will explain in more detail this concept and its usage in Section 3.2. Domain related knowledge is taken into account for the designing of rule templates towards certain applications. We demonstrate how to use the rule templates to integrate the domain knowledge,

and show the effectiveness of rule evaluations that utilize the rule templates. We also present experiments to show that with domain knowledge considered in the rule evaluation process, a smaller amount of rules with very high interestingness will be extracted.

- Rough Sets Theory based Rule Evaluations. Rough sets theory was introduced by Pawlak in the early 1980's [72]. He introduced an early application of rough sets theory to knowledge discovery systems, and suggested that a rough sets approach can be used to increase the likelihood of correct predictions by identifying and removing redundant variables. Efforts into applying rough sets theory to knowledge discovery in databases have focused on decision making, data analysis, discovering and characterizing the inter-data relationships, and discovering interesting patterns [73]. It is shown to be an interesting and powerful theory, and it has been used previously in attribute selection, rule induction, classification, multi-agent systems, medical diagnosis and other application domains. According to the rough sets theory, an information system can be considered as a decision table, which is used to specify what conditions lead to decisions. A sample decision table is shown in the following Table 1.1.

Table 1.1: A Decision Table for Predicting the Mileage of Cars

U	model	cylinder	transmission	weight	mileage
1	USACar	6	auto	heavy	low
2	JapanCar	4	manual	light	high
3	GermanCar	6	auto	heavy	medium
4	JapanCar	4	auto	light	medium

A decision table can be defined as $T = (U, C, D)$, where U is the set of objects in the table, C is the set of the condition attributes and D is the set of the decision attributes. In Table 1.1, $\{model, cylinder, transmission, weight\}$ is the set of condition attributes, and $\{mileage\}$ is the set of decision attributes. A reduct of a decision table

is a subset of the condition attributes that are sufficient to define the decision attributes. Reducts are often used in the attribute selection process at the data preprocessing stage in a knowledge discovery system. For example, {*model,transmission*} can be a reduct of Table 1.1. A reduct is not unique [72], and there may exist multiple reducts for one decision table. The core of a decision table is contained in every reduct, and it can be considered as the essential information of a decision table. Any reduct generated from the original data set cannot exclude the core attributes. Reduct and core are often used in the attribute selection process. We will give more detailed explanation for the rough sets theory and its usage in Section 2.1.

Since one of the uses for rough sets theory is to select the most relevant attributes towards a classification task, and to remove the unimportant attributes, rules generated by a learning algorithm can be considered more important if there exist important attributes in the rules. Therefore this theory can provide a theoretical foundation for rule evaluations. We study the mechanism of rough sets theory, and we propose a Rule Importance Measure, and a Rules-As-Attributes Measure that are both based on rough sets theory. We consider the important attributes as indications of the domain related knowledge. Therefore by combining such indications into rule evaluations, the important rules extracted through these measures can represent the important information contained in the original knowledge. We will demonstrate how to use such evaluation measures in practical applications such as medical diagnosis and personalization systems in Chapter 4 and Chapter 5.

- Data Preprocessing and Rule Evaluations. In practical applications, there usually exists incomplete data from the data collection process, because of the unavailability of information, redundant diagnosis tests, unknown data and so on. Discarding all data containing the missing attribute values cannot fully preserve the characteristics of the original data. Such incompleteness affects the rule generation process. Either the rule generation algorithms have to be adapted to handle the incomplete input, or the generated rules have to be further processed in order to understand the discovered knowledge with the incomplete information. Processing data with incomplete information becomes an important problem in data mining and knowledge discovery tasks. In this thesis, we focus on how to preprocess data containing in-

complete, missing attribute values. We believe that well preprocessed data can help with rule evaluations as well. Preprocessing data containing missing values can be integrated together with rule evaluations such as the Rule Importance Measure and the Rules-As-Attributes Measure to extract important rules.

Grzymala-Busse *et al.* [28] summarized nine approaches to solve missing values, such as assigning the most common values or assigning an average value to the missing attribute. These approaches make good use of all the available data. However the assigned value may not come from the information from which the data was originally derived; thus, noise is brought to the data. Let us consider the decision table Table 1.1 as an example. Suppose in the second data record, the value under attribute "transmission" is missing, as shown in Table 1.2. If we assign the most common value "auto" for attribute "transmission" to the missing value, we obtain information specifying that "a Japanese car, with four cylinders, automatic transmission, and light weight has a high mileage". This is contradictory to the fourth record, which indicates such a car has a medium mileage.

Table 1.2: Table 1.1 with One Missing Value

U	model	cylinder	transmission	weight	mileage
1	USACar	6	auto	heavy	low
2	JapanCar	4	?	light	high
3	GermanCar	6	auto	heavy	medium
4	JapanCar	4	auto	light	medium

In this thesis, we explore a new usage of association rule algorithms to predict missing attribute values, combining with the rough sets theory. We name it the "ItemRSFit" approach [58]. We will discuss the mechanism of this approach on predicting missing attribute values, and show empirical results on various applications in Chapter 6. Our experiments indicate this proposed approach obtains a higher prediction accuracy than other existing approaches.

We also demonstrate through a rough set based knowledge discovery system, illustrated by Figure 7.2 in Chapter 7, how the methods discussed in this thesis can be adapted in a practical system. One of the proposed rule evaluation measures, the Rule Importance Measure, has its usage and value demonstrated through a case study on an actual user-centric web personalization system.

1.4 Contributions

The focus of this thesis is to solve the problems of how to interpret discovered knowledge (in the format of rules), how to decide for a user whether one rule is better than another without examining all the rules one by one manually, and how to provide a way to extract such better and important knowledge.

The contributions in this thesis are:

- We demonstrate through empirical studies how appropriately defined rule templates can be used towards a recommendation application as rule interestingness measures. A recommender system is an intelligent system that can predict a user's interests based on a database of known users' profiles. We show a new method of using association rule algorithms for recommender systems, which apply properly defined rule templates to obtain interesting recommendations. Our method does not require users to provide scores for every item in the system, in order to generate recommendations. The rule templates can be used during the rule generation process to limit both the type of rules expected and the quantities of rules. This approach is a subjective rule interestingness measure, which can be combined together with other rule interestingness measures for rule evaluation purposes.

- We propose a Rule Importance Measure which is an automatic and objective approach to extract and rank important rules. This measure is applied throughout the rule generation process. The Rule Importance Measure differentiates rules by indicating which rules are more important than other rules. The Rule Importance Measure can be used in a variety of applications such as medical diagnosis, construction of spam filters, object labeling in criminology and so on. Our method is among

the few attempts to apply rough sets theory to association rule generation, in order to improve the utility of an association rule. The Rule Importance Measure is different from both rule interestingness measures [35] and rule quality measures [17], which are the two commonly used approaches on evaluating rules. Most of the rule interestingness measures are used to evaluate classification rules, and different people have different definitions for "interestingness". Our Rule Importance Measure is instead applied to evaluate association rules. It is an easy and objective measure. The Rule Importance Measure is different from the rule quality measures as well. They are often used in the post-pruning process of the knowledge discovery procedure to remove redundant rules, and applied on classification rules. The Rule Importance Measure is instead applied from the process of reduct generation to rule generation; therefore, our measure does not require the rules to be generated before being evaluated. One can use the Rule Importance Measure upon the data set directly, and obtain a list of ranked rules by their importance.

- We propose a Rules-As-Attributes method to evaluate important rules by taking advantage of rough sets theory. We consider rules generated from the original data set as attributes in the new constructed decision table. Reducts generated from this new decision table contain essential attributes, which are the rules. Only important rules are contained in the reducts. We call such rules "Reduct Rules". Experiments show that the Reduct Rules are more important, and this new method provides an automatic and effective way of ranking rules.

- We explore a new usage of association rule algorithms on predicting missing attribute values, combining with the rough sets theory. The experimental results show the new proposed ItemRSFit approach obtains higher prediction accuracy than most of the existing approaches [28]. It relies on its own data as a knowledge base and therefore the predicted values are not biased.

- We discuss how knowledge discovery systems can, in particular, benefit from the integration of the approaches proposed in this thesis. We also demonstrate how our approaches can be used for a specific real world problem. We show through an actual user-centric web personalization system the utilities of the proposed approaches in

this thesis for rule evaluations to predict whether an online purchase will happen. Such a prediction is made according to the observed online searching and browsing user behaviours. Through this case study, we also find that certain user-centric features are more important than others in predicting online purchases. For example, we can discover the fact that some user made a purchase online previously indicates such a user is more likely to perform online purchases in the future.

1.5 Organization of This Thesis

Throughout this thesis we present a series of rule evaluation approaches as well as case studies. There are three main parts in this thesis. In the first part, we introduce our proposed rule evaluation approaches. In the second part, we discuss our research efforts on processing missing attribute values, which can contribute to the process of rule generation as well as rule evaluation. In the third part of this thesis, we demonstrate through a specific case study how the proposed approaches in Part I and Part II can be utilized in an actual system.

As background, in Chapter 2, we introduce rough sets theory, the related important concepts, and current rough set based knowledge discovery systems. We also introduce the association rule algorithms and briefly discuss current rule evaluation techniques, including rule interestingness measures, the Rule Importance Measure, and the Rules-As-Attributes Measure. Background work on recommender systems is also presented in this chapter.

Part I. From Chapter 3 to Chapter 5, we discuss our rule evaluation approaches.

In Chapter 3, we provide a detailed survey of the current existing rule interestingness measures, and we analyze their application domains through experiments. We then introduce a novel approach of using rule templates as one of the interestingness measures to facilitate recommending rules generated by association rule algorithms. Experiments on a recommendation data set demonstrate the effectiveness of this rule measure.

In Chapter 4, we introduce the new measure, the Rule Importance Measure, based on rough sets theory. We explain the model, and show the experimental results through an artificial car data set [39], UCI machine learning data sets from the University of California,

Irvine [21], and a geriatric care data set [53], which is a real world data set from Dalhousie University Medical School to predict the patients' survival status. We demonstrate that the Rule Importance Measure provides a diverse way of ranking how important a rule is.

In Chapter 5, We explain a new method to discover important rules by considering rules as attributes, the Rules-As-Attributes Measure. Association rules are used for rule generations. A new decision table is constructed by considering all the rules as condition attributes. Reducts generated from this new decision table contain essential attributes, which are the rules. Only important rules are contained in the reducts. Experiments on an artificial data set, UCI [21] machine learning data sets and real-world data sets show that the reduct rules are more important and representative than those that do not exist in the reduct of the new decision table. This new method provides an automatic and effective way of ranking rules.

Part II. In Chapter 6, we discuss a method of processing missing attribute values based on rough sets theory and association rules. Empirical studies on UCI [21] machine learning data sets and a real world data set demonstrate a significant increase of prediction accuracy obtained from this new integrated approach.

Part III. In Chapter 7 we demonstrate how to use the proposed rule evaluation techniques and data preprocessing approaches proposed in Chapters 3, 4, 5, and 6 within a knowledge discovery system. We also provide a case study of using the Rule Importance Measure in a user-centric web personalization system, and as such we show empirically how the techniques discussed in this thesis can be adapted into an actual application.

In Chapter 8, we summarize the main contributions of this thesis, which lie in a series of rule evaluation measures and the explorations of their empirical applications. Researchers interested in rule evaluations, as well as their applications in data mining systems will be interested in this thesis. We discuss possible future work as the extension of our current work.

Our earlier thoughts on many issues presented in this thesis have appeared in several publications [51, 52, 53, 54, 55, 56, 57, 59, 60, 61], which are encouraging indications of the interests in our research within the data mining community.

Chapter 2

Background

The main focus of this thesis is to propose rough set based rule evaluation approaches; therefore, we review related work on rough sets theory and on rule generation and evaluation, as background work. We first introduce rough sets theory, which serves as the theoretical foundation for the rule evaluation approaches we proposed later in the thesis. Current rough set based knowledge discovery systems are summarized as well. We then discuss background work in rule generation and evaluation, beginning with a discussion of association rules, used as the basis for the rule generation and evaluation approaches presented within this thesis. Related work on recommender systems is discussed at the end of this chapter.

2.1 Rough Sets Theory

Rough sets theory was first introduced by Pawlak in the 1980's [72]. An early application of rough sets theory to knowledge discovery systems was introduced to identify and remove redundant variables, and to classify imprecise and incomplete information. Reduct and core are the two important concepts in rough sets theory. Based on Pawlak's book [72], we explain the basic concepts in rough sets theory in the following.

2.1.1 Decision Table

A data set can be represented as a decision table, which is used to specify what conditions lead to decisions. A decision table is defined as $T = (U, C, D)$, where U is the set of objects in the table, C is the set of the condition attributes and D is the set of the decision attributes. We show an example of a decision table in the following Table 2.1.

Table 2.1: Artificial Car Data Set

ID	make_model	cyl	door	displace	compress	power	trans	weight	Mileage
1	USA	6	2	Medium	High	High	Auto	Medium	Medium
2	USA	6	4	Medium	Medium	Medium	Manual	Medium	Medium
3	USA	4	2	Small	High	Medium	Auto	Medium	Medium
4	USA	4	2	Medium	Medium	Medium	Manual	Medium	Medium
5	USA	4	2	Medium	Medium	High	Manual	Medium	Medium
6	USA	6	4	Medium	Medium	High	Auto	Medium	Medium
7	USA	4	2	Medium	Medium	High	Auto	Medium	Medium
8	USA	4	2	Medium	High	High	Manual	Light	High
9	Japan	4	2	Small	High	Low	Manual	Light	High
10	Japan	4	2	Medium	Medium	Medium	Manual	Medium	High
11	Japan	4	2	Small	High	High	Manual	Medium	High
12	Japan	4	2	Small	Medium	Low	Manual	Medium	High
13	Japan	4	2	Small	High	Medium	Manual	Medium	High
14	USA	4	2	Small	High	Medium	Manual	Medium	High

This table shows an artificial data set about the cars [39]. The mileage of a car is related to the model of the car, the number of cylinders, the number of doors, the displacement, the compression, the power, the transmission, and the weight of the car. Table 2.1 can be used to decide whether a car has a high or medium mileage according to its features (e.g., the model, the transmission and the weight). For example, the first row of this table specifies that a USA car, with 6 cylinders, 2 doors, medium displacement, high compression, high power, automatic transmissions, and medium weight, has a medium mileage.

The rows in this table are called the objects, and the columns in this table are called attributes [40]. Condition attributes are the features of a car related to its mileage; therefore, $C = \{$*make_model, cyl, door, displace, compress, power, trans, weight*$\}$. *Mileage* is the decision attribute; therefore, $D = \{$*Mileage*$\}$. There are 14 objects in this data, and there do not exist missing attribute values.

Here we only look at the situation when the value of the decision attributes is binary. And we will not discuss the situation when the condition attributes have missing values.

The reduct and the core are important concepts in rough sets theory. Reduct sets contain all the representative attributes from the original data set. A reduct contains a subset of condition attributes that are sufficient to classify the decision table. A reduct may not be unique. The core is contained in all the reduct sets, and it is the necessity of the whole data. Any reduct generated from the original data set cannot exclude the core attributes.

Table 2.2: Multiple Reducts for the Artificial Car Data Set

No.	Reduct Sets
1	{make_model, compress, power, trans}
2	{make_model, cyl, compress, trans}
3	{make_model, displace, compress, trans}
4	{make_model, cyl, door, displace, trans, weight}

Table 2.2 shows the reducts of the car data set generated by the ROSETTA software [69]. For example, a reduct can be a set of condition attributes containing {*the model, the compression, the power and the transmissions*} of a car. With this reduct, all the 14 objects can be correctly classified completely (according to their mileage type). A subset of {*make_model, cyl*} is not a reduct of this car data, because with only these two attributes one cannot fully classify all the objects; in addition, there exists redundancy and contradictions. For example, in Table 2.1, with a subset of {*make_model, cyl*}, we cannot classify object No.7 and No.8. They both describe USA cars with 4 cylinders, but they have different mileage.

A reduct is often used in the attribute selection process to reduce unnecessary attributes towards decision making applications. According to the reduct No.1 in Table 2.2, one can generate a rule, i.e., a USA car with high compression, high power and automatic transmission has medium mileage, which is more succinct than a rule specifying that a USA car with 6 cylinders, 2 doors, medium displacement, high compression, high power, automatic transmissions and medium weight, has medium mileage.

Core attributes are the essential information in a data set. The core attributes are contained by all the reducts. From Table 2.2, we can see the intersection of all the listed reducts is as follows.

$$\{make\_model, trans\}$$

This set contains the core attributes. Core attributes can be obtained by the core generation algorithms proposed by Hu *et al.* [40], discussed in greater detail in Section 2.1.3.

More discussion on other related concepts in rough sets theory is included in Appendix A.

2.1.2 Rough Sets based Knowledge Discovery Systems

We briefly survey current rough sets based knowledge discovery systems. We discuss the individual functions of each system based on general characteristics, such as the data sets, the preprocessing tasks, the related rough sets tasks, the rule generations and so on.

1. **ROSETTA** ROSETTA [69] is freely distributed. Downloadable versions for both the Windows and Linux operating systems are available. The software supports the complete data mining process, from data preprocessing, including handling incomplete data, data discretization, generating reduct sets which contain essential attributes for the given data set, to classification, rule generation, and cross validation evaluation. Some discretization and reducts generation packages are from the RSES library [12].

2. **RSES2.2** RSES [12] stands for Rough Set Exploration System. There are downloadable versions for both the Windows and Linux operating systems. It is still maintained and being developed. The system supports data preprocessing, handling incomplete data, data decomposition, reducts generation, classification, and cross validations.

3. **ROSE2** ROSE [75] stands for Rough Sets Data Explorer. This software is designed to process data with large boundary regions. The software supports data preprocessing, data discretization, handling missing values, core and reducts generation, classifications and rule generation, as well as evaluations. This software provides not only the classical rough set model, but also the variable precision model, which is not provided by [12] and [69].

4. **LERS** LERS [24] stands for Learning from Examples based on Rough Sets. It is not publicly available. The system was designed especially to process missing values of attributes and inconsistency in the data set. Certain rules and possible rules are both extracted based on the lower and upper approximations.

In addition to the rough sets based systems mentioned above, there are other available knowledge discovery systems based on the methodologies of rough sets such as DBROUGH [41] and GROBIAN [44].

These systems demonstrate the use of rough sets theory for knowledge discovery by several researchers.

2.1.3 Current Reduct Generation and Core Generation Approaches

As discussed earlier, a *reduct* of a decision table is a set of condition attributes that is sufficient to define the decision attributes. A reduct does not contain redundant attributes towards a classification task. It is often used in the attribute selection process to reduce the redundant attributes, and to reduce the computation cost for rule generations. There may exist more than one reduct for each decision table. Finding all the reduct sets for a data set is NP-hard [48]. Approximation algorithms are used to obtain reduct sets [10]. The intersection of all the possible reducts is called the *core*. The core is contained in all the reduct sets, and it is the essential part of the whole data. Any reduct generated from the original data set cannot exclude the core attributes.

Previously, many research efforts on designing reduct generation and core generation approaches have been proposed. In this section, we summarize a few current algorithms and software that are commonly used and clarify where they are introduced in our thesis.

- **ROSETTA** In this thesis, we use ROSETTA [69] GUI version 1.4.41. For reduct generation, the software provides Genetic reducer, Johnson reducer, Holte1R reducer, Manual reducer, Dynamic reducer, RSES Exhaustive reducer and so on. Genetic reducer is an approximation algorithm based on a genetic algorithm for multiple reducts generation. The Johnson reducer generates only a single reduct with minimum number of attributes.

  In this thesis, we use Genetic reducer for multiple reduct generation in Chapter 4 and 7, and the Johnson reducer with the default option of full discernibility[1] [68] for single reduct generation in Chapter 4, 5 and 6. We choose Johnson reducer from ROSETTA, because it is designed to generate one single reduct with the minimal number of attributes. For the multiple reduct generations used in Chapter 4, we use ROSETTA's Genetic reducer because this allows no constraints on the number of generated reducts, and we consider this to be preserving the original characteristics of the data set, which is appropriate towards our goal of designing automatic rule evaluation approaches. The exhaustive reducers in ROSETTA and RSES also generate multiple reducts, but this function cannot be used for large datasets [68]. Therefore we do not use them in this thesis.

- **RSES** The current version of RSES [12] is RSES 2.2.1. RSES provides a genetic algorithm to control the number of reducts generated, which is appropriate for larger data sets to only generate representative reducts. The RSES system is described in more detail in Section 7.2, and its reduct generation algorithm is used within the case study described in Section 7.3.

Note that there are other reduct generation approaches provided by some other software such as ROSE2 [75]. In ROSE2 software, there are three reduct generation functions, the "lattice search", "heuristic search" and "manual search" approaches [75]. The "lattice

[1]For reduct generation, there are two options on discernibility provided by the ROSETTA software, which are full discernibility and object related discernibility [68]. With the option of full discernibility, the software will produce a set of minimal attribute subsets that can discern all the objects from each other. With object related discernibility, the software produces reducts that can discern a certain object from all the other objects.

search" approach for reduct generation is used when the expected number of reducts are rather small. The other two reduct generations support larger datasets although they require domain experts' knowledge on the data for choosing attributes. In this thesis we are interested in discovering knowledge from real world applications, which often involve large data sets; therefore, we do not use this software in this thesis.

Hu's Reduct and Core Generation

- Hu *et al.* [40] proposed a new rough set model based on database operations such as cardinality and projection. By combining a relational algebra with the rough sets theory, the approach is designed to increase the efficiency of the core and reduct computation. A reduct is redefined based on the database operations.

  The reduct is defined to be a subset $REDU(\subseteq C)$ of condition attributes with respect to the decision attribute D, where $REDU$ is a minimum subset of attributes that has the same classification power as the entire condition attributes. Let $K(REDU, D)$ be the proportion of the data instances in the decision table that can be classified. K is also defined to be the degree of dependency between $REDU$ and the decision attribute D, and is the stopping criteria for the algorithm, as shown in Eq. 2.1. $Card$ denotes the count operation in databases, and Π denotes the projection operation in databases.

$$K(REDU, D) = \frac{Card(\Pi(REDU + D))}{Card(\Pi(C + D))}. \tag{2.1}$$

  A measure of *merit value* is defined to evaluate the effect of each condition attribute on the decision attribute D. For a condition attribute $C_i \in C$, the merit of C_i can be calculated by

$$Merit(C_i, C, D) = 1 - \frac{Card(\Pi(C - \{C_i\} + D))}{Card(\Pi(C + D))}. \tag{2.2}$$

  During the reduct generation, the condition attribute with the highest merit value at the moment is included in the reduct. In case multiple highest merit values exist, the condition attribute with the least combination with other attributes in the current reduct is selected. The algorithm iterates until the minimum set of attributes which is as representative as the entire condition attributes is obtained. The reduct generation

algorithm is shown in Algorithm 1. The reduct generation is designed to guarantee

Algorithm 1: Hu's Reduct Generating Algorithm

input : Decision table $T(C, D)$, C is the condition attributes set; D is the decision attribute set.

output: $REDU$, reduct of C.

```
1  Core Generation Algorithm to generate Core ;
2  REDU = Core;
3  AR = C − REDU;
4  for each attribute C_i ∈ AR do
5  |  Merit(C_i, C, D) = 1 − Card(Π(C−{C_i}+D)) / Card(Π(C+D))
6  end
7  maximum(Merit(C_j, C, D)) ;
   /*In case there are several attributes with the same merit value,
     choose the attribute which has the least number of combinations with
     those attributes in REDU.  minimum(Card(Π({C_j} + REDU)))        */
8  REDU = REDU + {C_j}, AR = AR − {C_j};
9  if K(REDU, D) = 1 then return REDU ;
10 else go to Step 4
```

that the generated reduct will have the minimum number of attributes.

- Recall that the core represents the most important information of the original data set; all reducts contain the core.

  Since it is infeasible to obtain the core attributes by intersecting all the possible reducts, other approaches are proposed to generate the core attributes. Hu *et al.* [40] introduced a core generation algorithm based on rough sets theory and efficient database operations, without generating reducts. The algorithm is shown in Algorithm 2, where C is the set of condition attributes, and D is the set of decision attributes.

  This algorithm is developed to consider the effect of each condition attribute on the

Algorithm 2: Core Generating Algorithm

input : Decision table $T(C, D)$, C is the condition attributes set; D is the decision attribute set.

output: $Core$, Core attributes set.

```
1 Core ← φ;
2 for each condition attribute A ∈ C do
3 |   if Card(Π(C − {A} + D)) ≠ Card(Π(C − {A})) then
4 |   |   Core = Core + {A};
5 |   end
6 end
7 return Core;
```

decision attribute. The intuition is that, if the core attribute is removed from the decision table, the rest of the attributes will bring different information to the decision making. A theoretical proof of this algorithm is provided in [40]. The algorithm takes advantage of efficient database operations such as count and projection. Since the attributes of the core are contained in any reduct sets of a data set, this algorithm also provides an evaluation to justify the correctness of the reduct sets.

We use Hu's algorithm for core generation in Chapter 4, 5 and 6.

There are other reduct generation approaches such as the QuickReduct algorithm, which was first applied in information retrieval systems to reduce the dimensions of the input text data [19]. The algorithm uses the same stopping criteria of the degree of dependency as Eq. 2.1 to select a reduct. Comparing to Hu's reduct generation, this algorithm initializes the reduct set with an empty set, whereas for Hu's approach the reduct set is initialized to be the core set. Note that Hu's reduct generation and QuickReduct generation do not always produce the minimum reduct [89]. Recent research [95] indicates an addition-only strategy normally will not produce a minimum reduct.

2.2 Association Rules

The association rule algorithm was first introduced in [3] and is commonly referred to as the apriori association rule algorithm. It can be used to discover rules from transaction datasets. The algorithm first generates frequent itemsets, which are sets of items that have transaction support more than the minimum support; then based on these itemsets, the association rules are generated which satisfy the minimum confidence. Many contributions on how to efficiently generate frequent itemsets and generate rules have been reported [15, 32, 74, 96].

Association rule algorithms can be used to find associations among items from transactions. For example, in *market basket analysis*, by analyzing transaction records from the market, we could use association rule algorithms to discover different shopping behaviours such as, when customers buy bread, they will probably buy milk. This type of behaviour can be used in the market analysis to increase the amount of milk sold in the market.

An association rule [3] is a rule of the form $\alpha \rightarrow \beta$, where α and β represent itemsets which do not share common items. The association rule $\alpha \rightarrow \beta$ holds in the transaction set D with *confidence* c if c% of transactions in D that contain α also contain β. *Confidence* can be represented as $c = \frac{|\alpha \cup \beta|}{|\alpha|}$. The rule $\alpha \rightarrow \beta$ has *support* s in the transaction set D if s% of transactions in D contain $\alpha \cup \beta$. *Support* can be represented as $s = \frac{|\alpha \cup \beta|}{|D|}$. Here, we call α antecedent, and β consequent. Confidence gives a ratio of the number of transactions that the antecedent and the consequent appear together to the number of transactions the antecedent appears. Support measures how often the antecedent and the consequent appear together in the transaction set. The following example gives a sample association rule.

Example 1 In the market basket analysis, given customers' shopping carts containing {bread, beer, cheese, apple, banana, beef, ..., icecream}, an association rule indicating the associations between frequently purchased items in the shopping carts,
bread $\rightarrow$ *cheese* $(80\%, 60\%)$ states that 80% of the customers who bought bread, also bought cheese, and that 60% of customers bought both items.

A problem of using the association rules algorithm is that there are usually *too many rules generated* and it is difficult to analyze these rules. Support and confidence are often

used as interestingness measures to control how interesting the rules are. Generally, a rule is considered interesting if the rule has higher support and higher confidence than the predefined minimum support and confidence for rule generations [35].

2.3 Rule Evaluation Approaches

2.3.1 Rule Interestingness Measures

Knowledge discovered from data mining and knowledge discovery processes can be represented in different forms, such as association rules, classification rules, sequential patterns and so on. In general the amount of generated knowledge is very large, but not all of the knowledge is interesting and useful. This is because there is usually redundant information in the huge amount of input data, and the knowledge containing the redundant information is not interesting. In addition, some knowledge may be obvious according to a certain domain. The rule interestingness measure is a technique to evaluate how interesting, useful and relevant the knowledge is. Different applications may have different interestingness measures emphasizing different aspects of the applications. For example, support and confidence are used to measure how interesting the association rules are. These two measures are used to evaluate the item-item relations within a transaction data. Rules generated based on more frequently occurred together items are more interesting. On the other hand, the J-measure [82] evaluates classification rules. Rules more related to discrete-valued attributes are considered more interesting. Hilderman and Hamilton provided an extensive survey on the current interestingness measures [35] for different data mining tasks. We will discuss rule interestingness measures, and show how to use rule templates [45], which describe patterns appearing both in the antecedent and in the consequent of association rules, as one of the interestingness measures for a recommendation application in Chapter 3.

2.3.2 Rule Quality Measures

The concept of rule quality measures was first proposed by Bruha [17]. The motivation for exploring this measure is that decision rules are different with different predicting abilities, different degrees to which people trust the rules and so on. Measures evaluating these

different characteristics should be used to help people understand and use the rules more effectively. These measures have been known as the rule quality measures. The rule quality measures are often applied in the post-pruning step during the rule extraction procedure [6]. In general, rule generation system uses rule quality measures to determine the stopping criteria for the rule generations and to extract high quality rules. We will discuss the rule quality measure in Section 4.3.6 for comparison with the Rule Importance Measure.

2.3.3 Rule Importance Measures

The Rule Importance Measure [57] is a novel rough set based rule evaluation measure that we propose to evaluate association rules. It is applied from the process of reduct generation to rule generation to evaluate how important the association rules are. The intuition behind this measure is that, there exist multiple reducts for a data set. Each reduct is representative of the original data, therefore rules generated from reducts are representative rules extracted from the data set. Since a reduct is not unique, rule sets generated from different reducts contain different sets of rules. However, more important rules will be generated in most of the rule sets; less important rules will be generated less frequently than those more important ones. The frequencies of the rules can therefore represent the importance of the rules. We present the detail of this measure in Chapter 4.

2.3.4 Rules-As-Attributes Measure

A rule evaluation measure based on rough sets theory is proposed by considering rules as the conditional attributes within a decision table [59]. A set of association rules are first generated for a given data set. Then such rules are considered as the condition attributes in a new constructed decision table. The decision attributes in this new constructed decision table are from the original data set. Reducts further generated from this new decision table contain essential attributes, which are the rules. Only important rules are contained in the reducts. This new method provides an automatic and effective way of ranking rules. We present the details of this approach in Chapter 5.

2.4 Recommender Systems

A recommender system is an intelligent system that uses a database of known users' profiles to predict a new user's interests. There are two types of recommender systems: content-based recommender systems and collaborative filtering recommender systems [9]. Content-based recommender systems make recommendations to new users based on the content of the available users' interests. Content-based recommender systems, such as NewsWeeder [49] and InfoFinder [46], require representative properties from the data, which are hard to extract. On the other hand, collaborative filtering systems observe the behaviours and the patterns of the current users, and make recommendations based on the similarities between the current users and other users. Much research work on collaborative recommender systems, such as GroupLens [79] and Ringo [80], are receiving attention. Other researchers follow the model-based approaches, which construct proper models for different user behaviour patterns. The behaviour of a new user can be predicted based on these user behaviour models. In order to build the model, various methods are used, such as cluster analysis, Bayesian belief network and the most recent probabilistic mixture model [**?**, 38, 37, 97]. In this thesis, we are interested in collaborative filtering systems, which are also called personalization systems.

We introduce two well-known collaborative filtering systems as follows.

- Amazon. Amazon (http://www.amazon.com/) is an online shopping store mainly for books, music CD, video, and other products (such as apparel and electronics) as well. In a general searching for a particular product, for example, the book of "The Da Vinci Code", the web site will recommend products "Customers who bought this also bought", "Angels & Demons" by the same author, as one of the recommended books, shown in Figure 2.1. Frequent visiting users are expected to sign in with their accounts to access, search, and shop for the products they are interested in. This system therefore uses customers' identities, as well as the web pages they have browsed for creating and updating their recommendation databases (dynamically). Customers with similar user profiles may be clustered into the same user group. User-group-specific recommendations can be created for certain types of users. Recommendations are suggested to the new customers who have demonstrated

Figure 2.1: Amazon Online Recommender System

similar profiles to those profiles in the recommendation database. Therefore the system provides more precise recommendations to the customers who sign in with their accounts and have previous histories in the system based on matching their profiles with the created profiles.

- MovieLens. MovieLens (http://movielens.umn.edu/login) is a movie recommendation system developed by University of Minnesota. The project is still ongoing. Users must have an individual account to get signed in order to use the recommendation system; however, the user's personal information (such as age group, gender, etc.) is not required. If a new user is observed, the system requires the minimum input of a ranking for 15 movies from the user side. The movies ranked by the new users can be used not only to update the current databases, but also to create the user's profile.

The more movies the user ranks, the more precise the recommendation will be. The system uses a MySQL database to store the data, and Perl to parse the query and search, and achieve the matched results. Figure 2.2 shows the ranked movies by the user (the left figure, ranking shown in blue color) and the recommended movies by the system (the right figure, ranking shown by red color).

Figure 2.2: Rated and Recommended Movies by MovieLens

What is unique with the MovieLens system is that the system not only recommends the movies to the users, but also provides the ratings of the movies to the user, from the highest interested movies to the lowest interested movies.

In Chapter 3 we discuss the application of recommending movies to users as the context of our exploration of rule interestingness.

Chapter 3

Rule Templates as Rule Interestingness Measures

One of the main issues with data mining and knowledge discovery in databases is how to interpret and evaluate the discovered knowledge from large data. Many research efforts concentrate on how to use certain measures to rank the generated knowledge. The motivation for this chapter is that we would like to explore the possibilities of applying association rules for recommender systems. A recommender system [9] uses a database of known users' profiles to predict a new user's interests. A recommender system can be considered as a knowledge discovery system, and the purpose of such system is to generate rules that can help making "recommendations" such as decision support, medical diagnosis, revenue forecast and so on. Such "recommendations" can be represented as rules, which are generated from a knowledge discovery system. Association rule algorithms are used to discover associations among items in transaction datasets. These associations can serve as a rule generation engine for recommender systems, which suggests interesting items based on the associations. However, applying association rule algorithms directly to make recommendations usually generates too many rules; thus, it is difficult to find interesting recommendations for users.

In this chapter, we concentrate on interpreting the rules and patterns by using the rule interestingness measures. We first survey the existing interestingness measures which are designed for different application purposes. We then introduce a new approach of using rule

templates as one of the interestingness measures to facilitate recommending rules generated by association rule algorithms. Let us take a recommender system as an example. The interesting rules in such applications are rules which contain recommended items on the consequent part of the rules. Rule templates can be defined to satisfy this demand. When such templates are used in the association rule generation process, only rules used towards the recommendation purposes will be generated.

Rule templates restrict the form of association rules; therefore, they are used as one of the interestingness measures to reduce the number of rules in which users are not interested. By defining appropriate rule templates, we are able to extract interesting rules for users in a recommender system.

The survey of current interestingness measure is provided in Section 3.1. Section 3.2 explains the definitions of rule templates, as well as how to use the templates, including examples. Section 3.3 provides the experimental results on the EachMovie collaboration data set. The concluding remarks on this section are provided in Section 3.4.

3.1 A Review of the Current Methods to Evaluate Rule Interestingness

Knowledge discovered from data mining and knowledge discovery processes can be represented in different forms, such as association rules, classification rules, sequential patterns and so on. In general the amount of generated knowledge is very large, but not all of the knowledge is interesting and useful. This is because there is usually redundant information in the huge amount of input data; thus, the knowledge containing the redundant information is not interesting. Some knowledge may be obvious according to the domain. The rule interestingness measure is a technique to evaluate how interesting, useful and relevant the knowledge is. Different applications may have different interestingness measures emphasizing different aspects of the applications.

Hilderman and Hamilton provided an extensive survey on the current interestingness measures [35] for different data mining tasks. In this thesis, we are interested in exploring interestingness measures related to association rules. We give a brief introduction to the interestingness measures for association rules applications.

- Support and Confidence. The measures used for apriori association rule algorithm was proposed by Agrawal *et al.* [3]. The **support** of a rule measures how often the antecedent and the consequent of a rule appear together in the transaction. The **confidence** of a rule measures given that the antecedent appears in the transaction, how often the antecedent and the consequent exist together. The minimum values of support and confidence are predetermined to generate the association rules. These two measures evaluate rules based on the statistical significance of the rule. Support and confidence are used in the situation when the interest of the application is to find associations between different items. The higher these two measures are, the more interesting the rules are considered to be. These two measures are objective measures.

- Lift. In [31] it is shown that the confidence of an association rule is an estimate of the conditional probability of the consequent given the antecedent. Rules with high confidence are considered to be interesting, but the confidence cannot measure the independence between the consequent and the antecedent of the rule. The **lift** of an association rule is used to measure whether the consequent and the antecedent are independent or positively or negatively correlated. Suppose the association rule is $A \rightarrow B$. The lift is measured by the following formula.

$$lift = \frac{P(A \cap B)}{P(A)P(B)}$$

  $lift < 1$ implies the negative correlation between A and B; $lift > 1$ implies a positive correlation, which also shows that the occurrence of one implies the other; $lift = 1$ implies no correlations between A and B, and they are independent. Lift can be used to extract negative rules, which cannot be extracted by confidence measures.

- Chi-squared Test. Brin *et al.* [16] also pointed out the problems with the original confidence measure for association rules. The confidence measure may not rank interesting rules especially when correlation is the measure to be used. The chi-squared test for correlations is proposed to measure association rules because the measure not only ranks the correlations but also the negative implications. χ^2 statistic is defined

as follows, where E is the expectation.

$$\chi^2 = \sum_{j,k} \frac{(f_{jk} - E(f_{jk}))^2}{E(f_{jk})}$$

For a contingency table shown as Table 3.1, the chi-square value is expressed as follows [85].

Table 3.1: Sample 2×2 Contingency Table for Binary Variables

	B	$\bar{B}$	
A	f_{11}	f_{10}	f_{1+}
$\bar{A}$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	N

$$\chi^2 = \frac{(f_{11} - f_{1+}f_{+1}/N)^2}{f_{1+}f_{+1}/N} + \frac{(f_{10} - f_{1+}f_{+0}/N)^2}{f_{1+}f_{+0}/N} + \frac{(f_{01} - f_{0+}f_{+1}/N)^2}{f_{0+}f_{+1}/N} + \frac{(f_{00} - f_{0+}f_{+0}/N)^2}{f_{0+}f_{+0}/N}$$

The χ^2 measure can show whether the items in the rule are independent of each other, but it cannot rank the rules. Also when the contingency table is sparse, the measure is less accurate [31].

- Rule Templates. The concept of "Rule Template" was first presented by Klemettinen *et al.* in 1994 [45]. Rule templates describe patterns for those items that appear both in the antecedent and in the consequent of association rules [45]. A rule matches the defined template if this rule is an instance of the template. Rule templates are "syntactic constraints" [35] that specify certain forms of rules to be of interest. For a rule in the form of the following.

$$Attribute_1, Attribute_2, \ldots, Attribute_k \rightarrow Attribute_m$$

Rule templates can be specified for both the antecedent and the consequent parts of the rule to extract interested patterns. Since the templates are specified according

to the need of the application, they are a subjective measure. The advantages of the rule templates are that they can be defined quite flexibly according to the domain experts. Interesting templates, uninteresting templates, surprising templates, important templates and so on can all be defined according to the expected knowledge. The rule templates can be used during the rule generation process to only extract expected rules, or can be used after the rule generation to filter out rules that do not match the templates. More examples on how to use rule templates are found in the following Section 3.2.

- Gray and Orlowska's Interestingness Measure. An interestingness measure combining **support** and **confidence** together is introduced by Gray and Orlowskaz [25] as following. Given a rule $X \rightarrow Y$,

$$I = ((\frac{P(X \cap Y)}{P(X) \times P(Y)})^k - 1) \times (P(X) \times P(Y))^m$$

where $P(X \cap Y)$ is the confidence, $P(X \times Y)$ is the support. The discrimination factor is defined as $\frac{P(X \cap Y)}{P(X) \times P(Y)}$. k and m are predefined parameters that specify the importance of the discrimination factor and the support factor in the integrated interestingness measures. This measure is an objective measure, and higher measures imply more interesting rules.

- Neighborhood-Based Unexpectedness. Dong and Li [22] proposed an interestingness measure to evaluate association rules based on the unexpectedness comparing to other rules in its neighborhood. The neighborhood of a rule is decided by a distance function which compares certain characters of the rules. This measure is a subjective measure. Neighborhood-based interestingness measures such as unexpected confidence interestingness and isolated interestingness are also introduced.

- Subpatterns and Superpatterns. Subpatterns and superpatterns of a pattern is explored to measure the interestingness of the association rules [92]. The authors suggested that interestingness measures such as support and confidence should be designed based on the frequencies of a pattern to its subpatterns, and it is not necessary to consider the subpatterns of a pattern as interesting. Therefore by combining

superpatterns together, redundant subpatterns can be filtered out and only more interesting patterns are kept. This approach provides an objective measure of extracting interesting rules.

- Other Measures. There are a few other measures that are used for association rules. Liu *et al.* proposed a measure [29] to extract exceptional and reliable patterns from data. This measure is designed for the purpose of discovering rules that are not considered as interesting by the support and confidence measures. It is also efficient to extract weak and exceptional rules. The authors assume that exceptional rules usually have a lower support value, while the items contained in the rule are important items. Zhong *et al.* [67] introduced the concept of "peculiarity rules" that are the unexpected though interesting association rules. Peculiar data can be found through a peculiarity factor, which evaluates whether one attribute is very different from the other attribute. A threshold is predefined to evaluate the peculiarity of the data, then peculiar rules containing peculiar data are extracted. Other measures for different applications can be found at [36].

Not all the interestingness measures are the same. Depending on different application purposes, appropriate rule interestingness measures should be selected to extract proper rules. More than one measure can be applied together to evaluate and explain the rules. Tan *et al.* [86] evaluate twenty-one measures in their comparative experiments and suggest different usage domains for these measures. They provide several properties of the interestingness measures so that one can choose a proper measure for certain applications. Their experiments also imply that not all the variables perform equally well at capturing the dependencies among the variables. Furthermore, there is no measure that can perform consistently better than the others in all application domains. Different measures are designed towards different domains.

In this chapter, we show that using the support and confidence measures only is not sufficient for the application of generating recommendations for users. Rule templates, as one of the rule interestingness measures, can be combined together with support and confidence to facilitate evaluating rules. We further discuss and introduce other evaluation measures in Chapters 4 and 5.

3.2 Rule Templates

What is a Rule Template

The concept of "Rule Template" was first presented by Klemettinen *et al.* in 1994[45]. Rule templates describe patterns for those items that appear both in the antecedent and in the consequent of association rules [45]. A rule matches the defined template if this rule is an instance of the template; that is, we consider structure information inherent in the data. By defining patterns and expressions that account for the data under consideration and the kinds of interactions being sought, interesting rules are selected and uninteresting rules will be filtered out.

Examples

We provide examples to define rule templates, and show how to design templates to select interesting rules.

In a typical market basket analysis of grocery items purchased by customers, for example, we list transactions for each customer as follows in Table 3.2.

Table 3.2: Sample Transactions for Each Customer

Customer ID	Items Bought by Each Customer
1	Milk, Bread, Lettuce, Mushrooms
2	Cream, Cheese, Muffins, Shrimp, American Wine
3	Butter, Cheese, Lobster, Australian Wine
4	Cheese, Eggs, Peppers, Salmon
...	...

For the transaction data sets used in this chapter, all the items in the data sets can be classified into different classes. Table 3.3 lists items with their classes.

A rule template is a sequence of n components α_i followed by a component β, and is defined to be

$$\alpha_1, \alpha_2, \ldots, \alpha_n \Rightarrow \beta.$$

Table 3.3: Class Information and Their Items

Class Name	Items Belonging to this Class
Dairy	Milk, Cream, Eggs, Cheese, Butter, Yogurt, ...
Bakery	Bread, Croissants, Muffins, Pies, Tarts, Cheesecakes, ...
Vegetables	Broccoli, Carrots, Beans, Lettuce, Mushrooms, Peppers, ...
Seafood	Salmon, Tuna, Shrimp, Scallops, Crab, Lobster, ...
White Wine	American Wine, Australian Wine, Portugal Wine, ...
...	...

Each component α_i $(1 \leq i \leq n)$ and β is of the form A, C, $C+$ or $C*$, where A is an attribute name and C is a class name. A class name C followed by a plus (+) can be one or more instances of class C. A class name C followed a star (*) can be zero or more instances of class C.

For example, using the transaction data of Table 3.2, a possible rule template can be defined as shown in Example 2.

Example 2

$$Cheese, Seafood \Rightarrow White\ Wine$$

which implies that we are interested in the kind of rules, such that when customers buy cheese and seafood, they will probably buy white wine too. In this template, "Cheese" is an attribute name, "Seafood" and "White Wine" are both class names.

We say a rule matches the template if the rule is an instance of the pattern. An example of a rule matching the rule template defined by Example 2 is given by Example 3.

Example 3 The customer is interested in White Wine. It would be useful to know any associations where White Wine is on the consequent part, and cheese with a kind of seafood item are both on the antecedent part of the rule. For example,

$$Cheese, Shrimp \Rightarrow American\ Wine$$

$$Cheese, Lobster \Rightarrow Australian\ Wine$$

these rules match the template

$$Cheese, Seafood \Rightarrow White\ Wine$$

as defined by Example 2.

By defining templates, we can extract rules that are interesting, and filter out uninteresting rules if the rules do not match the template. The following Example 4 show rules not matching the template defined by Example 2.

Example 4

$$Cheese, Muffins \Rightarrow American\ Wine$$

$$Butter, Lobster \Rightarrow Australian\ Wine$$

Here, in the first example, "Muffins" do not belong to the "Seafood" class. In the second example, the attribute name is "Butter"; however, the attribute name defined in Example 2 is "Cheese". Different attribute names do not match this template.

Rule templates can also be used to define rules in which we are not interested. Therefore, we can use this kind of template to filter out rules, as illustrated in Example 5.

Example 5 It may be the case that we are not interested in rules with the item "Bread" in the antecedent part. We can define a template as such

$$Bread, Dairy* \Rightarrow Dairy$$

Therefore, rules matching this template will be filtered.

Why We Use Rule Templates

In the situation when a huge number of rules are generated, certain rule interestingness measures are needed to limit the quantities of the rules, and at the same time extract interesting rules. Depending on the applications, people may be interested in different rules and knowledge. Rule templates, as one of the subjective measures, can be defined

according to different applications, and can be applied during the rule generation process. Therefore, certain rules according to the applications can be extracted to facilitate the understanding of the knowledge.

How We Use Rule Templates

The concept of rule template can be applied to different applications. The type of applications we are interested in this chapter is to use association rule algorithms for recommender systems. By using appropriate rule templates, we could use the generated rules to make recommendations.

Usually we consider the consequent part of the rule for making recommendations.

Few research efforts can be found on applying association rule algorithms for collaborative recommender systems. The disadvantage of using an association rule algorithms is that there are usually too many rules generated, and it is difficult to make recommendations to the user effectively and efficiently. By examining domain related information, examining the inherent information of the data, we can define appropriate templates to generate interesting rules for the recommendation tasks.

Using Templates

We would like to apply the association rule algorithm for recommender systems. In addition to using *support* and *confidence*, we examine the role of the rule template to predict the items in which users are most likely interested. For example, a rule template like

$$Item_1, Item_2 \rightarrow Item_5[support = 0.6, confidence = 0.8]$$

means 60% of users like $Item_1$, $Item_2$ and $Item_5$, and 80% of users who like $Item_1$ and $Item_2$ also like $Item_5$. The consequent of the rule is used for making the recommendation.

In addition to the above basic templates for recommender systems, more templates could be defined by analyzing the data set, and deciding what kinds of information should be put into consideration.

For example, the items in the transaction data set Table 3.2 belong to different classes as displayed class information shown in Table 3.3. We can define certain templates specifying that if all the items in the antecedent part of the rule belong to the same class, and if the

item in the consequent part of the rule also belongs to the same class, then most likely, such rules are more interesting for those applications that look for items within the same class. The following example illustrates this type of template.

Example 6 A rule

$$Salmon, Tuna, Shrimp \Rightarrow Crab[support = 0.6, confidence = 0.8]$$

is found to be more interesting than a rule

$$Salmon, Croissants \Rightarrow Mushrooms[support = 0.6, confidence = 0.8]$$

because "Salmon", "Tuna" and "Shrimp" all belong to the seafood class. "Salmon", "Croissants" and "Mushrooms" are from different classes. Similarly, certain applications may consider rules with items belonging to different class as more interesting; therefore, corresponding templates can be defined for such purposes as well.

Note that we can also adopt an attribute-oriented generalization approach, "concept hierarchy" [30] to define the proper rule templates.

3.3 Experiments

In this section, we conduct experiments on using rule templates on an movie recommendation task, to show that rule templates can be used as one of the rule interestingness measures towards a recommendation application. We first introduce related work, then we describe the experimental data. We further discuss the evaluation measures that we consider appropriate for this experiment, and show the experimental results.

3.3.1 Related Work

Rule Interestingness Measures

One category of evaluating rules is to rank the rules by rule interestingness measures. Rules with higher interestingness measures are considered more interesting. The rule interestingness measures, originated from a variety of sources, have been widely used to

extract interesting rules. Different applications may have different interestingness measures emphasizing different aspects of the applications. In this experiment, we use rule templates [45] as one of the rule interestingness measures for a recommendation task.

Recommender Systems

A recommender system is an intelligent system that uses a database of known users' profiles to predict a new user's interests. There are two types of recommender systems: content-based recommender systems and collaborative filtering recommender systems [9]. Content-based recommender systems make recommendations to new users based on the content of the available users' interests. On the other hand, collaborative filtering systems observe the behaviours and the patterns of the current users, and make recommendations based on the similarities between the current users and other users. We are interested in the collaborative filtering systems. Publicly available data source for research on collaborative filtering systems is quite limited. We have observed research efforts on movie recommendations including the MovieLens [66] from University of Minnesota, the EachMovie [1] collaborative recommender systems from [37, 63, 65, 97] and so on.

Existing Challenges on Collaborative Filtering System

We summarize a few challenging problems in the current developing of collaborative filtering system.

- There is currently limited research on using association rule algorithms for making recommendations. This is because it is difficult to appropriately adjust the support and confidence measures to produce the right amount of association rules so that users can understand such huge amount of recommendations easily. Rules may contain from one item to more than 10 items on the right hand side; therefore, recommending so many items altogether is not realistic.

- The number of recommendations can be very large. It is difficult to recommend all these possible recommendations without processing them. How to make important and interesting recommendations are the important tasks in the postprocessing stage of the recommendations.

- For a collaborative filtering system to provide precise recommendations, users personal information is quite necessary for creating user profiles and form user groups. Therefore new users can be easily grouped into specific groups according to the users' personal information (such as MovieLens). However, in most situation, not many users would like to release their personal information. In such cases, recommendations may not be very precise or personalized towards an individual person (such as Amazon). How to develop a system that can use user's personal information while at the same time preserving users' privacy is very challenging.

EachMovie Data Set

For this work, we also use the EachMovie dataset, a well known test bed for collaborative recommender systems as our experimental data. This dataset is provided by Compaq's Systems Research Center [1].

The EachMovie data set is a collection of users' votes on $1,628$ different movies from $72,916$ users over an 18 month period. Each movie is assigned to no less than one movie genre, including action, art or foreign, romance, thriller, horror, animation, comedy, classics, drama and family. Each movie is voted based on a five star evaluation scheme, therefore has 6 possible voting values ranging from 0 to 1 with equal space. By removing all the movies that have no votes and users that have never voted, we are left with $61,265$ valid users, $1,623$ valid movies and $2,811,983$ votes.

We would like to gain some insight of the credibility of each user's opinion on which the collaborative recommender system is based to provide recommendations. For this purpose, we plot the cumulative frequency of user's votes as shown in Figure 3.1 [61].

The mean and standard deviation of the voting values approximate a normal distribution. The global mean of all the votes is around 0.6. Most importantly, the number of votes of each user follows a very skewed distribution as indicated in the cumulative frequency plots on the number of votes for each user.

From Figure 3.1 we can see that, around 40% of the users only voted for no more than 1% of the all the movies, 40% of the users voted for no less than 2% of all the movies, 20% of users voted for more than 4% of the total movies, 5% of the users voted for more than 10% of the total movies.

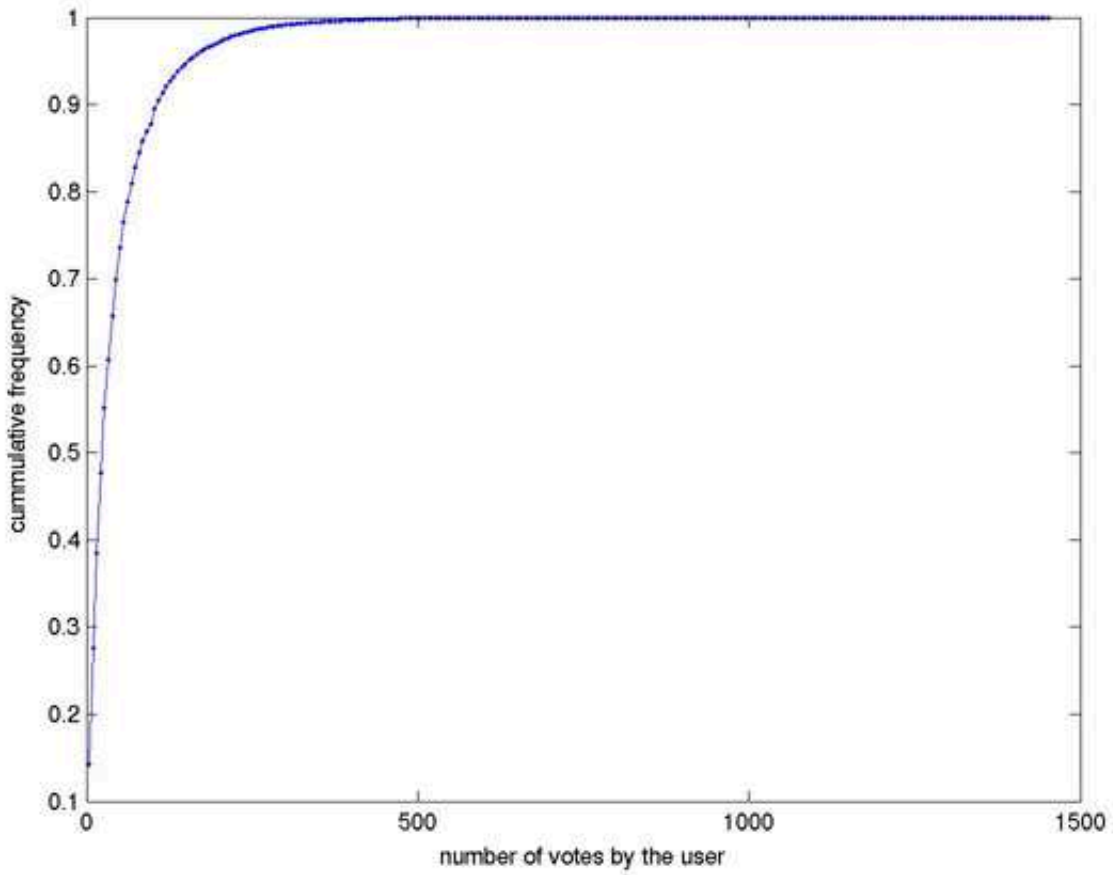

Figure 3.1: The Cumulative Frequency Plot on the Number of Votes by Each User

Based on Figure 3.1, the voting frequency of each user can be used as a threshold parameter for preprocessing. Here, we want to select users who are frequent movie viewers of a certain level. Observed from Figure 3.1, 40% of the users voted for no less than 2% of all the movies, which are 32 movies. We will only consider users who have voted for no less than 32 movies. Because this value represents the major number of votes, it is also small enough to avoid sampling users who are real movie fans.

The disadvantage of using association rule algorithms is that there are usually too many rules generated, and it is difficult to make recommendations to the user effectively and efficiently. Klemettinen *et al.* proposed a new interestingness measure, i.e., rule templates, to find interesting rules from a large set of association rules [45]. By defining rule templates more appropriately, only rules that match the templates will be extracted. Thus this method can be used in the post processing of the association rules generated, and can increase both the efficiency and the accuracy of recommendations.

It is important to note that the research on rule templates [45] was proposed in order to determine interesting rules during the rule evaluation process. Its application to assist with recommendations is a new approach proposed in this thesis. We believe that it is worthwhile to determine the value of rule templates in determining recommendations.

On the other hand, Lin *et al.* [63] proposed a new association rule algorithm for collaborative recommendation. New parameters, such as minimum number of rules generated, were defined in this algorithm to generate smaller rule sets. Rules with only one consequent were mined, and were used to predict the behaviours for a specific target user. The EachMovie dataset [1] was used as the collaborative filtering dataset. The authors were interested in two types of association rules, one was for movie associations, the other was for user associations. According to different voting scores, both movies and their scores could be used to generate associations. The minimum support value does not have to be specified in advance for rule generations; instead, the algorithm automatically adjusts the support value based on the number of rules expected. The data structure used in this algorithm considered both the movie and the score for every recommendation; therefore, the recommended rules contained more information. However, the authors did not explain their choice of the EachMovie dataset subset they used for testing, nor did they consider the effect of movie genres to enhance the performance. Since we are proposing the new

application of association rules on recommendations, it is valuable to see how recommendations can be done in the context of well established association rule algorithms that are employed already in many contexts.

3.3.2 Experimental Design

According to the kind of rules we wish to generate, we process the transaction dataset as depicted in Table 3.4.

Table 3.4: Preprocessing for User Transactions

User ID	Movie ID's
1	1, 2, 3, 4, 5, 6, 7, 8
2	1, 3, 4, 5, 10, 11, 19
3	2, 3, 5, 7, 110, 112, 150
4	1, 8, 9, 12, 17, 19, 22
...	...

Each transaction represents all the movies voted by a person.

Since we are interested in predicting movies a user would be interested in and making recommendations, we define rule templates to reduce the number of rules and specify the recommendation rules we are interested in.

Template 3.1 specifies that there is only one consequent in the generated rules. Template 3.2 specifies that only rules whose antecedents and consequent items all belonging to the same movie genre will be generated. For this template, we first prepare a new movie genre dataset, as shown in Table 3.5. In Table 3.5, all the possible genres to which one movie can belong are listed. We assign action movie to be Genre 1, art or foreign movie Genre 2, romance movie Genre 3, thriller movie Genre 4, horror movie Genre 5, animation movie Genre 6, comedy movie Genre 7, classics movie Genre 8, drama movie Genre 9, and family movie Genre 10. For example, the first row in Table 3.5 can be interpreted to state that movie 1 is both art or foreign movie, and comedy movie.

Table 3.5: Movie Genre Information

Movie ID	Genre ID's
1	2 [Art, Foreign], 7 [Comedy]
2	7 [Comedy]
3	5 [Horror]
4	6 [Animation]
5	5 [Horror]
6	1 [Action], 10 [Family]
7	3 [Romance]
8	4 [Thriller]
9	9 [Drama]
10	8 [Classics]
...	...

Template 3.1

$$\langle Movie_1, Movie_2, \ldots, Movie_m \rangle \rightarrow \langle Movie_n \rangle \tag{3.1}$$

Template 3.2

$$\langle Genre\_Movie_1 \cap \ldots \cap Genre\_Movie_m \cap Genre\_Movie_n \rangle \neq \phi \tag{3.2}$$

where $Movie_1$, ..., $Movie_m$ and $Movie_n$ are different from each other.

3.3.3 Evaluation Function

Our motivation for the experiments is to generate association rules that can be used for recommendations of movies. For example, when a person watched $movie_1$, and $movie_2$, this person will be recommended to watch $movie_3$. We perform cross validation by dividing the complete dataset into training data and testing data. Training data can be used to generate association rules. Then these rules such as $movie_1, movie_2 \rightarrow movie_3$ will be

validated on the testing data. For each transaction in the testing data, we consider that a rule can correctly classify the transaction if and only if both the antecedent and the consequent of the rule are contained in the transaction. For a transaction ($movie_1$, $movie_2$, $movie_3$, $movie_5$), we consider that the rule $movie_1, movie_2 \rightarrow movie_3$ can correctly classify this transaction because when a person watched $movie_1$, and $movie_2$, the recommended $movie_3$ is indeed watched by this person.

We use accuracy [78] to evaluate the performance of our method. Eq. 3.3 gives the accuracy computed as a function of c and t. c stands for the number of transactions such that the predictions are correct, which also implies the number of transactions containing both the antecedent and the consequent of a rule. t stands for the number of transactions for which the rule makes a prediction, which also implies the number of times the antecedent of a rule belongs to a transaction.

$$accuracy = \frac{c}{t} \tag{3.3}$$

The following example illustrates the use of accuracy function. Below we show the generated sample rules and the sample test dataset.

Generated Rules		
2	$\rightarrow$	3
1, 2	$\rightarrow$	4
1, 4	$\rightarrow$	5

Test Dataset
1 2 3
1 4 8
6 9
7 9 12

In this example, the three rules have their antecedents in one transaction respectively. We therefore have $t = 3 \times 1 = 3$. The rule $2 \rightarrow 3$ is the only rule whose antecedent and consequent are in the same transaction, which is (1, 2, 3), therefore $c = 1$. According to our accuracy function, the average accuracy for the sample rules on this sample test dataset is $\frac{1}{3} = 33.33\%$.

In our experiment, we applied Borgelt's apriori algorithm [15] to generate frequent itemsets.[1] And we use our algorithms to read these frequent itemsets as input, and implement the templates as well as our rule generating algorithm using C++, and the target

[1]Downloaded from http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html#assoc

compiler and platform is g++ and Unix respectively. We perform 4-fold cross validation for the following experiments.

All the following experiments were performed on Sun Fire V880, four 900Mhz Ultra-SPARC III processors, with 8GB of main memory [2].

Table 3.6: Recommendation Accuracies on The EachMovie dataset

	Accuracy			
Confidence	Support = 20%	Support = 30%	Support = 40%	Support = 50%
70%	86.96%	85.27%	84.10%	82.81%
80%	89.79%	88.49%	86.68%	83.77%
90%	93.73%	93.23%	92.14%	92.51%

Largest Data Set

This experiment is to test whether the whole EachMovie dataset can be used to generate rules. We applied Template 3.1, and only generated rules with one consequent part. We did not put any constraint on the number of rules generated. Training data is 80% of the transaction datasets, and test data is 20% of the total transactions.

Table 3.6 shows the recommendation accuracies when support ranges from 20% to 50% with different confidence levels.

We can see that for the same confidence value, as support increases, the accuracy becomes lower. This is because fewer rules are generated for higher support values, and thus the number of transactions containing both the antecedent and the consequent parts of the rules are fewer. For the same support value, when confidence gets higher, accuracy becomes higher. This is because confidence, as one of the interestingness measures, is a measure for how correct a rule is. The higher the confidence, the more correct a rule becomes and therefore, the higher accuracy we obtain. This test also shows that recent association rule algorithms can be applied to large datasets.

Table 3.7: Accuracy when confidence = 80% (First Trial)

Support	Accuracy_Rules	Accuracy_with_Genre
80%	72.89%	75.75%
70%	73.30%	75.28%
60%	75.83%	77.42%
50%	78.33%	79.52%
40%	80.78%	82.43%

Table 3.8: Accuracy when Confidence = 90%(First Trial)

Support	Accuracy_Rules	Accuracy_with_Genre
80%	73.95%	77.06%
70%	79.97%	80.25%
60%	81.20%	82.93%
50%	84.62%	85.87%
40%	87.25%	88.35%

Template 3.2 Performance

In order to test whether Template 3.2 increases the accuracy, we apply this template to two subsets of data which are commonly used for this dataset [13] and [63].

First Trial. The first subset we tried is from [13]. Training data represents the first $1,000$ users who have rated for more than 100 movies. Testing data comes from the first 100 users whose user ID is larger than $70,000$, and who also rated more than 100 movies.

Since the paper specified that the maximum length of a rule is 8, here we limit the length of a rule to be 8 as well.

Table 3.7 shows the accuracies of this experiment when confidence is 80%. The first column shows the support value, the second column shows the accuracy from applying the first template to our algorithm, and the third column shows the accuracy of adding genre

information, which is Template 3.2.

Table 3.8 shows the performance of this experiment when confidence is 90%.

From Table 3.7 and Table 3.8, we can see that when applying movie genre information, only extracting rules where all the movies belong to the same genre, we obtain a higher recommendation accuracy.

In order to show the computing overhead is also reduced by applying Template 3.2, we list the number of itemsets and rules generated according to different support values.

Table 3.9: Itemset Size and Rule Size when confidence = 80%(First Trial)

Minimum support	Frequent Itemsets	Association Rules	With Genre Rules
90%	2	0	0
80%	35	47	30
70%	272	608	198
60%	2,773	8,303	1,439
50%	35,276	139,796	10,385
40%	690,382	3,525,426	88,298

Table 3.10: Itemset Size and Rule Size when confidence = 90%(First Trial)

Minimum support	Frequent Itemsets	Association Rules	With Genre Rules
90%	2	0	0
80%	35	38	24
70%	272	392	127
60%	2,773	5,074	805
50%	35,276	79,353	5,404
40%	690,382	1,994,580	49,278

Table 3.9 and Table 3.10 show that using Template 3.2 reduces the number of rules

generated. As support values get lower, there are more frequent itemsets generated, thus more rules are generated; the accuracy increases as the support value decreases. This use of Template 3.2 shows that the more rules that are generated, the more accurate the recommendation will become. By adding the movie genre information, we extract only rules whose items in both antecedents and consequents belong to the same genre (as one of the interestingness measures), and the accuracy increases. Thus, if we want to recommend movies online to the user immediately, our method can generate movie recommendations with high accuracy.

By increasing the confidence, the accuracy will also be increased, thus better quality rules will be extracted. But some interesting rules may be filtered. By adding movie genre information, we can generate rules that apply for certain purposes.

Second Trial. The second subset we tried is from [63]. We used training data for the first $2,000$ users. Testing data comes from users whose like ratios are less than 0.75, from which we randomly selected 20 users as one test set. We repeated this choice of test set 4 times, from which we obtained the average accuracy.

The accuracy is shown by Table 3.11, and Table 3.12. The two tables show an average of more than 15% increase in accuracy using Template 3.2.

Table 3.11: Accuracy when confidence = 80%(Second Trial)

Support	Accuracy_Rules	Accuracy_with_Genre
30%	75%	100%
20%	78.30%	87.04%
10%	75.79%	91.83%
5%	78.65%	93.86%
4%	81.27%	94.72%

We list the number of itemsets and rules generated according to different support values for different confidence levels in Table 3.13 and Table 3.14.

Table 3.13 and Table 3.14 describe the size of frequent itemsets and the rule sets. As

Table 3.12: Accuracy when confidence = 90%(Second Trial)

Support	Accuracy_Rules	Accuracy_with_Genre
30%	0%	0%
20%	75%	100%
10%	81.44%	98.67%
5%	82.87%	97.64%
4%	83.37%	97.64%

Table 3.13: Itemset Size and Rule Size when confidence = 80%(Second Trial)

Minimum support	Frequent Itemsets	Association Rules	With Genre Rules
30%	17	1	1
20%	171	86	30
10%	9,023	15,671	1,360
5%	579,291	1,926,017	37,855
4%	2,326,891	9,154,962	104,589

we can see, when confidence gets higher, there are fewer rules generated; when support gets higher, fewer rules are generated.

3.4 Conclusions

In this chapter, we discuss how to use rule templates as rule interestingness measures to extract interesting rules. We proposed a new method of applying the association rule algorithms for recommender systems. By applying appropriately defined rule templates, we obtained interesting rules. Experiments on a recommendation data set EachMovie dataset demonstrate the effectiveness of this rule measure. Unlike most current recommender

Table 3.14: Itemset Size and Rule Size when confidence = 90%(Second Trial)

Minimum support	Frequent Itemsets	Association Rules	With Genre Rules
30%	17	0	0
20%	171	6	4
10%	9,023	3,788	297
5%	579,291	745,971	12,954
4%	2,326,891	3,900,287	41,626

systems, our method does not consider specific score or vote values associated with every recommended item. Our method relaxes the strictness of considering a user's preference to a certain item in the recommender system. Since requiring a user to input his or her preference is compulsory to most current recommender systems, we envisage that without the preference information, future recommender systems using rule templates will be more convenient for users, as well as providing accurate recommendations to the users. According to our experimental results, the rule templates can be used during the rule generation process to limit both the type of rules expected and the quantities of rules. This approach is a subjective rule interestingness measure, which can be combined together with other rule interestingness measures for rule post-processing, and can be applied in other application domains such as decision support, medical analysis and so on. We adapt the usages of rule templates in our rule generation and evaluation process in Chapter 4 and 5.

Chapter 4

Rule Importance

4.1 Introduction

In this chapter, we discuss how rough sets theory can help in evaluating rules. We introduce the Rule Importance Measure to evaluate how important a rule is. Rules generated from reducts are representative rules extracted from the data set; since a reduct is not unique, rule sets generated from different reducts contain different sets of rules. Some rules appear in most of the rule sets; some rules appear less frequently across all the rule sets. The frequencies of the rules can therefore be used to determine the most important rules.

To test our hypothesis, we first use the ROSETTA rough sets toolkit [69] to generate multiple reducts. We then use apriori association rules generation [3] to generate rule sets for each reduct set. We are interested in applying these rules for making decisions. Therefore, the type of rules we are looking for are rules which have, on the consequent part, the decision attributes, or items that can be of interest for making decisions. Some rules are generated more frequently than the others among the total rule sets. We consider such rules more important. We define the Rule Importance Measure according to the frequency of an association rule among the rule sets. We will show by the experimental results that our method provides diverse measures of how important the rules are, and at the same time reduces the number of rules generated.

Our method is among the few attempts on applying rough sets theory to association rules generation to improve the utility of an association rule. The Rule Importance Mea-

sure is different from either the rule interestingness measures or the rule quality measures, which are the two well-known approaches on evaluating rules. Most of the rule interestingness measures are used to evaluate classification rules, and different people have different definition for "interestingness". The Rule Importance Measure is instead applied to evaluate association rules. It is an easy and objective measure. The Rule Importance Measure is different from rule quality measure as well which is often used in the post-pruning process of the knowledge discovery procedure to remove the redundant rules, and is applied on classification rules. In contrary, the Rule Importance Measure is applied from the process of reduct generation to rule generation, and the rules evaluated are association rules.

We discuss related work on association rule algorithms and rough sets theory on rule discovery in Section 4.2. In Section 4.3 we introduce the Rule Importance Measure. In Section 4.4 we experiment the rule importance measure on an artificial car data set, UCI data sets and a geriatric care data set. We summarize this chapter and discuss the continuing work in Section 4.5.

4.2 Related Work

An association rules algorithm helps to find patterns which relate items from transactions. For example, in market basket analysis, by analyzing transaction records from the market, we could use association rule algorithms to discover different shopping behaviours such as, when customers buy bread, they will probably buy milk. Association rules can then be used to express these kinds of behaviours, thus helping to increase the number of items sold in the market by arranging related items properly. A well known problem for association rules generation is that too many rules are generated, and it is difficult to determine manually which rules are more useful, interesting and important. In our study of using rough sets theory to improve the utility of association rules, we propose a new Rule Importance Measure to select the most appropriate rules. In addition to the experimentations on artificial data sets and UCI (University of California, Irvine) [21] data sets, we also perform the experiments on a larger data set, a geriatric care data set from Dalhousie University Medical School [53], to explore the application of the proposed method.

Rough sets theory was proposed to classify imprecise and incomplete information. Reduct and core are the two important concepts in rough sets theory. A reduct is a subset of attributes that are sufficient to describe the decision attributes. Finding all the reduct sets for a data set is a NP-hard problem [48]. Approximation algorithms are used to obtain the reduct set [10]. All reducts contain the core. Core represents the most important information of the original data set. The intersection of all the possible reducts is called the core. We use ROSETTA [69] for multiple reducts generation. We use Hu's core algorithm to generate core attributes (details discussed in Chapter 2).

There have been contributions on applying rough sets theory to rule discovery. Rules and decisions generated from the reducts are representative of the data set's knowledge. In [43], two modules were used in the association rules mining procedure for supporting organizational knowledge management and decision making. Self-Organizing Map was applied to cluster sale actions based on the similarities in the characteristics of a given set of customer records. Rough sets theory was used on each cluster to determine rules for association explanations. Hassanien [33] used rough sets to find all the reducts of data that contain the minimal subset of attributes associated with a class label for classification, and classified the data with reduced attributes. In Sections 4.3.5 and 4.3.6 we discuss other related research specific to the content of those sections.

Rough sets theory can help to determine whether there is redundant information in the data and whether we can find the essential data needed for our applications.

4.3 Rule Importance Measures

4.3.1 Motivation

In medical diagnosis, a doctor requires a list of symptoms in order to make a diagnosis. For different diseases, there are different patient symptoms to examine. However, there are some routine exams that the doctor must perform for all the patients, such as the age of the patient, the blood pressure, the body temperature and so on. There are other symptoms that doctors may take into consideration, such as whether the patients have difficulty walking, whether the patients have bladder problems and so on. We would like

to find the most important symptoms for diagnoses. We know that the symptoms that are checked more frequently are more important and essential for making diagnoses than those which are considered less frequently. However, both the symptoms that require frequent checking and the symptoms that are checked less frequently are included in the list of checkup symptoms. In this way, the doctor will make a precise diagnosis based on all possible patient information.

4.3.2 Defining the Rule Importance Measure

The medical diagnosis process can be considered as a decision making process. The symptoms can be considered as the condition attributes. The diagnosed diseases can be considered as the decision attributes. Since not all symptoms need to be known to make a diagnosis, the essential symptoms are considered as representative. These symptoms can be selected by a reduct generation algorithm.

All the patient information can also be represented in a transaction data set, with each patient's record considered to be an item set. The association rules algorithm can be applied on this transaction data set to generate rules, which have condition attributes on the antecedent part and decision attributes on the consequent part of the rules. Rules generated from different reduct sets can contain different representative information. If only one reduct set is being considered to generate rules, other important information might be omitted. Using multiple reducts, some rules will be generated more frequently than other rules. We consider the rules that are generated more frequently more important.

We propose a new measure, *Rule Importance*, to evaluate the importance of association rules. A rule is defined to be important by the following definition.

Definition 1 If a rule is generated more frequently across different rule sets, we say this rule is *more important* than rules generated less frequently across those same rule sets.

Rule Importance Measure is defined as follows,

Definition 2

$$\text{Rule Importance Measure} = \frac{\begin{array}{c}\text{Number of times a rule appears in all}\\ \text{the generated rules from the reduct sets}\end{array}}{\text{Number of reduct sets}}.$$

The definition of the Rule Importance Measure can be elaborated by Eq. 4.1. Let n be the number of reducts generated from the decision table $T(C, D)$. Let $RuleSets$ be the n rule sets generated based on the n reducts. $ruleset_j \in RuleSets$ $(1 \leq j \leq n)$ denotes individual rule sets containing rules generated based on reducts. $rule_i$ $(1 \leq i \leq m)$ denotes the individual rule from $RuleSets$. RIM_i represents the Rule Importance Measure for the individual rule. Thus the Rule Importance Measures can be computed by the following

$$RIM_i = \frac{|\{ruleset_j \in RuleSets | rule_i \in ruleset_j\}|}{n}. \tag{4.1}$$

The following example shows how to compute the Rule Importance Measure. We use the Iris [21] data set as an example, which is a data set containing three classes of iris plants, which are Iris setosa, versicolour and virginica. For the four attributes, we use "sl" to stand for attribute "sepal length", "sw" for "sepal width", "pl" for "petal length" and "pw" for "petal width". There are $n = 4$ reducts available for rule generations. For each of the reducts, the rule sets generated based on the reduct are shown below.

Reducts	Rule Sets
$\{sl, sw, pl\}$	$\{sl_{4.4} \rightarrow setosa, sw_{2.9} \rightarrow versicolor, pl_{1.9} \rightarrow setosa, \ldots\}$
$\{sw, pl, pw\}$	$\{sw_{2.9} \rightarrow versicolor, pl_{1.9} \rightarrow setosa, pw_{1.1} \rightarrow versicolor, \ldots\}$
$\{sl, pl, pw\}$	$\{sl_{4.4} \rightarrow setosa, pl_{1.9} \rightarrow setosa, pw_{1.1} \rightarrow versicolor, \ldots\}$
$\{sl, sw, pw\}$	$\{sl_{4.4} \rightarrow setosa, sw_{2.9} \rightarrow versicolor, pw_{1.1} \rightarrow versicolor, \ldots\}$

Rule $sl_{4.4} \rightarrow setosa$ is generated across 3 rule sets, therefore the rule importance is $RIM = \frac{3}{4} = 75\%$. For rules $sw_{2.9} \rightarrow versicolor$, $pl_{1.9} \rightarrow setosa$, $pw_{1.1} \rightarrow versicolor$, they are all generated from 3 of the 4 rule sets, therefore their rule importance is also 75%.

4.3.3 Modeling the Rule Importance Measure

The general model on which we compute the Rule Importance Measure is shown in Figure 4.1.

First during the data preprocessing step, the inconsistent data instances and the data instances containing missing attribute values are processed. Several approaches on processing data instances with missing attribute values are discussed in Chapter 6. Semantic

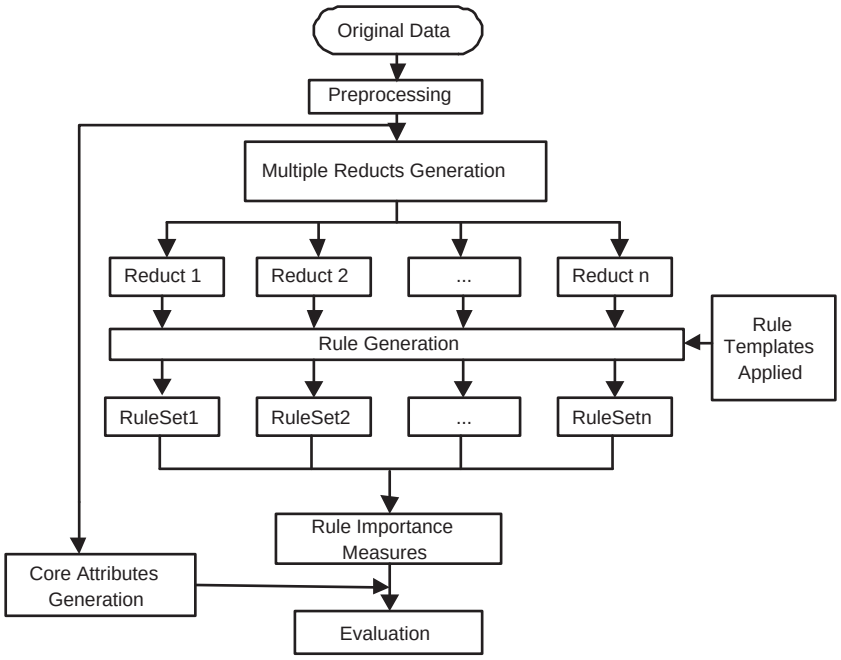

Figure 4.1: How to Compute the Rule Importance

methods on assigning missing values as well as objective methods (such as ignoring data instances containing missing attribute values, or assigning all possible average attribute values to the missing attribute [28]) can be used optionally. Inconsistency exists in a decision table when two or more data instances contain the same condition attribute values but different decision attribute values. These data instances must be removed. To remove them, we first sort the whole data set according to the condition attributes, excluding the decision attributes. Then we select data instances that contain the same condition attribute values, but different decision attribute values. They are removed during this stage. Discretization algorithms, such as equal frequency binning or entropy algorithm [69], are also applied during this stage if necessary. Core attributes are generated at the end of the data preprocessing stage. It is worthwhile to mention that core generation requires no inconsistencies in the data set.

After data is preprocessed, multiple reducts are generated. Various algorithms and rough set software provide multiple reducts generation. For example, ROSETTA's genetic

algorithm generates multiple reducts; RSES [12] provides a genetic algorithm for user defined number of reducts generation, which is appropriate in cases of larger data sets for generating representative reducts.

After multiple reducts are generated, the condition attributes contained in the reduct together with the decision attributes are used as the input data for rule generation. Rule templates, such as

$$\alpha_1, \alpha_2, \ldots, \alpha_n \Rightarrow \beta.$$

as discussed in Chapter 3, are applied in the rule generation step. Depending on different applications and the expected results, rule templates for desired types of rules and for subsumed rules are defined prior to the rule generation and are applied during the rule generation process. Multiple rule sets are therefore generated after the rule generations for multiple reducts. Rule Importance Measures are further calculated for each generated rule by counting the rule frequencies appearing across all the rule sets. Rules with their individual importance measures are ranked according to Eq. 4.1 and returned from the model.

In the evaluation stage of the model, core attributes play an important role for evaluating these ranked rules. Rules with 100% importance contain all the core attributes. Rules that contain more core attributes are more important than rules that contain fewer or none core attributes. Since core attributes are the most representative among all the condition attributes, more important rules contain these more representative attributes, which are the core attributes. Therefore by checking for the presence of the core attributes in the rules, we can evaluate the ranked rules with their rule importance.

4.3.4 Complexity Analysis

We analyze the time complexity for the proposed approach of generating important rules. Suppose there are N data instances in the data set, and M attributes for each data instance, N' is the number of distinct values in the discernibility matrix [72] which is a matrix composed of attributes for computing the core and the reduct, t is the number of multiple reducts for the data set, the time complexity in the worst case is analyzed as follows. The time complexity for multiple reducts generation is $O(N'^2)$ [88]. The core generation takes $O(NM)$ [40]. The apriori association rules generation takes $O(NM!)$ [3], therefore it

takes $O(tNM!)$ to generate multiple rule sets for multiple reducts. The calculation of the rule importance for the total rules k generated by the multiple rule sets takes $O(k \log k)$. In general, t is much smaller than N, therefore the time complexity of our approach is bounded by $O(N'^2 + NM + NM! + k \log k) \approx O(NM!)$ in the worst case.

4.3.5 How Rule Importance is Different from Rule Interestingness

Rule generation often brings a large amount of rules to analyze. However, only part of these rules are distinct, useful and interesting. How to select only useful, interesting rules among all the available rules has drawn the attention of many researchers. As discussed in Chapter 3, one of the approaches to help selecting rules is to rank the rules by "rule interestingness measures". Rules with higher measures are considered more interesting. The rule interestingness measures, originated from a variety of sources, have been widely used to extract interesting rules.

The Rule Importance Measure is a new measure to rank the rules. It is different from the rule interestingness measure in the following ways.

- The Rule Importance Measure is used to evaluate association rules. The rule interestingness measure applies to classification rules except that *support* and *confidence* are two necessary parameters used in association rules generation, and they are considered as rule interestingness measures. Rule interestingness measures cannot be used to measure association rules. The input data for association rules generation is transaction data, and there is usually no class label with the transaction data. However the class label is required for calculating the rule interestingness measure.

- The Rule Importance Measure is an objective measure. The rule interestingness measure can be either objective or subjective [1]. In order to determine whether a rule is "interesting" or not, different people may have different opinions. Therefore "domain experts" are required to help make evaluations. However the Rule Importance Measure does not require human supervision. Our Rule Importance Measure

[1]For example, in Chapter 3 we presented a rule template approach that labeled "genre" as a valuable feature. This would have been a subjective measure.

uses the notion of a "reduct" from rough sets theory. Recall that a reduct selects the maximally independent set of attributes from a data set; that is, the reduct contains a set of attributes that are sufficient to define all the concepts in the data. These attributes contained in the reduct are considered to be more important. The Rule Importance Measure is thus computed across all the rule sets generated from all the possible reducts of a data set. Since the reducts contain important attributes and rule sets generated from the reducts contain important rules, the Rule Importance Measure thus provides an evaluation of how important these rules are. There is no subjectivity involved in this measure. ROSETTA provides a genetic algorithm to generate multiple reducts. It is also not necessary to define and use the rule templates during the rule generations. On the other hand, the rule interestingness measure usually requires domain experts' evaluation [2].

- The Rule Importance Measure provides more direct and obvious measures. Rule interestingness measures often involve selections according to the specific applications. In [35] Hilderman and Hamilton showed that there is no rule interestingness measure that can always perform better than the others in all applications. Each individual rule interestingness measure is based on its selection bias on the data. In order to determine what is the best interestingness measure to use for certain application data, all the possible measures have to be compared to determine the best measure. But the Rule Importance Measure does not consider the type or applications of the data. It can be used directly on the data from any application field.

- The Rule Importance Measure reduces the amount of data required for rule generation by selecting only important attributes from the original data. The number of rules generated is thus greatly reduced. The rule interestingness measure is applied after the rules are generated. Therefore it requires more computational resources.

In summary, the Rule Importance Measure is simple, quick, easy to compute; it provides a direct and objective view of how important a rule is. Let us use the following example

[2]Rules that are considered interesting may not be important. We will discuss this through a comparison experiment as shown in Section 4.4.4.

to illustrate how the rule importance measure ranks rules according to the importance of a rule.

The data set used in the following example is an artificial data set about cars [39], as shown in Table 4.1. It is used to decide the mileage of different cars. The condition attributes are *make_mode, cyl, door, displace, compress, power, trans, weight*. *Mileage* is the decision attribute. There are 14 instances. The data set does not contain missing attribute values.

For the Car data set, ROSETTA software generates 4 reducts as shown in Table 4.2. The core attributes are, *make_model*, and *trans* as shown in the following Table 4.3.

Since we are interested in predicting the mileage of a car based on the model of a car, the number of doors, the compression, the weight as well as other factors related to a car, we would like to extract rules which have the decision attribute "mileage" on the consequent part of the rules. Therefore we specify the template for desired rules as shown by Eq. 4.2.

$$\langle model, cyl, \ldots, weight \rangle \rightarrow \langle mileage \rangle. \tag{4.2}$$

And if a rule

$$\langle JapanCar, weight\_medium \rangle \rightarrow \langle mileage\_High \rangle \tag{4.3}$$

is generated, rules such as Eq. 4.4

$$\langle JapanCar, trans\_manual, weight\_medium \rangle \rightarrow \langle mileage\_High \rangle \tag{4.4}$$

are removed, because this rule can be subsumed by the previous rule.

We generate the rule sets based on these 4 reduct sets with $support = 1\%, confidence = 100\%$, and we also rank their rule importance, as shown in Table 4.4.

Discussion

From Table 4.4, the first 2 rules have an importance of 100%. This observation matches our experiences on cars. The auto transmission cars usually have a lower mileage than the manual cars. Japanese cars are well known for using less gas and providing higher mileage. The rule "Door_4 → Mileage_Medium" has a lower importance because the number of doors belonging to a car does not affect car mileage. We noticed that the two rules with

Table 4.1: Artificial Car Data Set

make_model	cyl	door	displace	compress	power	trans	weight	Mileage
USA	6	2	Medium	High	High	Auto	Medium	Medium
USA	6	4	Medium	Medium	Medium	Manual	Medium	Medium
USA	4	2	Small	High	Medium	Auto	Medium	Medium
USA	4	2	Medium	Medium	Medium	Manual	Medium	Medium
USA	4	2	Medium	Medium	High	Manual	Medium	Medium
USA	6	4	Medium	Medium	High	Auto	Medium	Medium
USA	4	2	Medium	Medium	High	Auto	Medium	Medium
USA	4	2	Medium	High	High	Manual	Light	High
Japan	4	2	Small	High	Low	Manual	Light	High
Japan	4	2	Medium	Medium	Medium	Manual	Medium	High
Japan	4	2	Small	High	High	Manual	Medium	High
Japan	4	2	Small	Medium	Low	Manual	Medium	High
Japan	4	2	Small	High	Medium	Manual	Medium	High
USA	4	2	Small	High	Medium	Manual	Medium	High

Table 4.2: Reducts Generated by Genetic Algorithm for the Artificial Car Data Set

No.	Reduct Sets
1	{make_model, compress, power, trans}
2	{make_model, cyl, compress, trans}
3	{make_model, displace, compress, trans}
4	{make_model, cyl, door, displace, trans, weight}

Table 4.3: Core Attributes for the Artificial Car Data Set

Core Attributes
make_model, trans

Table 4.4: The Rule Importance for the Artificial Car Data Set

No.	Selected Rules	Rule Importance
1	Trans_Auto → Mileage_Medium	100%
2	JapanCar → Mileage_High	100%
3	USACar, Compress_Medium → Mileage_Medium	75%
4	Compress_High, Trans_Manual → Mileage_High	75%
5	Displace_Small, Trans_Manual → Mileage_High	50%
6	Cyl_6 → Mileage_Medium	50%
7	USACar, Displace_Medium, Weight_Medium → Mileage_Medium	25%
8	Power_Low → Mileage_High	25%
9	USACar, Power_High → Mileage_Medium	25%
10	Compress_Medium, Power_High → Mileage_Medium	25%
11	Displace_Small, Compress_Medium → Mileage_High	25%
12	Door_4 → Mileage_Medium	25%
13	Weight_Light → Mileage_High	25%

importance of 100% contain core attributes and only core attributes to make a decision of mileage. For the rest of the rules with importance less than 100%, the attributes on the left hand side of a rule contain non-core attributes. This observation suggests that core attributes are important when evaluating the importance of the rules. Our method of generating rules with reduct sets is efficient. There are $6,327$ rules generated from the original data without using reducts or rule templates. 13 rules are generated using reducts and rule templates.

4.3.6 How Rule Importance is Different from Rule Quality

The concept of rule quality measures was first proposed by Bruha [17]. The motivation for exploring this measure is that decision rules are different with different predicting abilities, different degrees to which people trust the rules and so on. Measures evaluating these different characteristics should be used to help people understand and use the rules more effectively. These measures have been known as rule quality measures.

The rule quality measures are often applied in the post-pruning step during the rule extraction procedure [6]. The measure is used to evaluate whether the rules overfit the data. When removing an attribute-value pair, the quality measure does not decrease in value, this pair is considered to be redundant and will be pruned. In general, rule generation system uses rule quality measures to determine the stopping criteria for the rule generations and extract high quality rules. In [7] twelve different rule quality measures were studied and compared through the ELEM2 [6] system on their classification accuracies. The measures include empirical measures, statistical measures and information theoretic measures.

The Rule Importance Measure is different from the rule quality measure because of the following.

- The Rule Importance Measure is used to evaluate how important is an association rule. Rule quality measures explore classification tasks of data mining, and are targeted towards improving the quality of decision rules. We cannot use rule quality measures to evaluate the association rules.

- The Rule Importance Measure takes transaction data as input. There is no class label from the transaction data. The measure evaluates how important is an association rule without considering other information from the data. Sometimes the transaction data can be processed by organizing the data into the form of a decision table. In this situation, the Rule Importance Measure evaluates the relations between the condition attributes and the class. However, the rule quality measures are used to evaluate the relations between the rules and the class.

- The Rule Importance Measure takes input of multiple reducts and multiple rule sets, then calculates the frequencies of each rule across multiple rule sets. The measure is used throughout the rule generation process. The rule quality measure is often used in the post-pruning process of the rule classification system.

- The Rule Importance Measure considers the representative attributes contained in the reducts, and rule generations are based on the reducts. Therefore, redundant attributes are removed before the rule generation, and the number of rules generated are much fewer than rules generated from the original data set. Thus the computation

cost is lower. When the rule quality measures are used to remove the low quality rules from the generated rules, the rule generation computation cost is greater than that of the Rule Importance Measure.

In summary, the Rule Importance Measure is different from the rule quality measure because of the differences between their application tasks, the processes where the measures are applied and the contents they measure.

4.4 Experiments

In this section, we explain the experiments we conducted to generate Rule Importance Measure on an artificial car data set, UCI data sets and a geriatric care data set.

The reduct is generated from ROSETTA GUI version 1.4.41. ROSETTA provides the following approximation algorithms for reducts generation: Johnson's algorithm, Genetic algorithm, Exhaustive calculation and so on. Johnson's algorithm returns a single reduct. Genetic algorithm returns more than one reduct. Exhaustive calculation returns all possible reducts, although given a larger data set, this algorithm takes a longer time to generate reduct sets [68]. In our experiment, we use the genetic algorithm [87] to generate multiple reduct sets with the option of full discernibility. The apriori algorithm [15] for large item sets generation and rule generation is performed on Sun Fire V880, four 900Mhz UltraSPARC III processors, with 8GB of main memory.

4.4.1 Specifying Rule Templates

The apriori association rules algorithm is used to generate rules. Because our interest is to make decisions or recommendations based on the condition attributes, we are looking for rules with only decision attributes on the consequent part. Therefore, we specify the following two rule templates to extract the rules we want as shown by Template 4.5, and to subsume rules as shown by Template 4.6.

$$\langle Attribute_1, Attribute_2, \ldots, Attribute_n \rangle \rightarrow \langle DecisionAttribute \rangle \tag{4.5}$$

Template 4.5 specifies that only decision attributes can be on the consequent part of a rule, and $Attribute_1$, $Attribute_2$, ..., $Attribute_n$ lead to a decision of $DecisionAttribute$.

We specify the rules to be removed or subsumed using Template 4.6. For example, given rule

$$\langle Attribute_1, Attribute_2 \rangle \rightarrow \langle DecisionAttribute \rangle \tag{4.6}$$

the following rules

$$\langle Attribute_1, Attribute_2, Attribute_3 \rangle \rightarrow \langle DecisionAttribute \rangle \tag{4.7}$$

$$\langle Attribute_1, Attribute_2, Attribute_6 \rangle \rightarrow \langle DecisionAttribute \rangle \tag{4.8}$$

can be removed because they are subsumed by Template 4.6.

The rule templates defined for the artificial car data set in the previous section, as shown by Eq. 4.2, can be used as an example to further explain how to define proper templates.

4.4.2 Experiments on UCI Data Sets

We experiment on selected UCI data sets [21] A through M described in Appendix C. In Table 4.5, we list the name of the data set, the number of condition attributes, the number of instances it contains, the number of reducts returned by ROSETTA's genetic algorithm, sample reducts and the core attributes returned by Algorithm 2 as shown in Chapter 2. In Table 4.6, we list the number of rules generated using the original data set with certain support and confidence values without applying the rule templates or the Rule Importance Measure, the number of rules generated from the reducts with the same support and confidence values, but now using the rule templates (as the Rule Importance Measure (RIM) procedure shown in Figure 4.1); and sample rules ranked by the Rule Importance Measure. The table demonstrates that we can make dramatic reductions in the number of rules that can be used for knowledge discovery and can generally provide some rules with a high measure.

Table 4.5: UCI Data Sets

Data Set	Condition Attributes	No. of Instances	No. of Reducts	Sample Reducts	Core Attributes
Abalone	8	4,177	16	{WholeWeight, ShuckedWeight, ShellWeight} {Height, WholeWeight, ShuckedWeight, VisceraWeight} {Sex, Length, Height, WholeWeight, ShellWeight}	Empty
Breast Cancer	9	286	1	{age, menopause, tumor-size, deg-malig, breast, breast-quad, irradiat}	age, menopause, tumor-size, deg-malig, breast, breast-quad, irradiat
Car	6	1,728	1	{buying, maint, doors, persona, lug_boot, safety}	buying , maint, doors, persona, lug_boot, safety
Glass	9	214	21	{RI, Al}, {Na, Si} {RI, Na, Mg}, {Na, Mg, K, Fe}	Empty
Heart	13	303	57	{age, chol, exang} {age, trestbps, chol} {chol, thalach, slope, ca} {sex, chol, oldpeak, ca, thal}	Empty
Iris	4	150	4	{sepalLength, sepalWidth, petalLength} {sepalLength, petal Length, petalWidth} {sepalWidth, petalLength, petalWidth} {sepalLength, sepalWidth, petalWidth}	Empty
Lympho-graphy	18	148	147	{blockofaffere, hangesinnode, changesinstru, specialforms, dislocationof, noofnodesin}	Empty
Pendigits	16	7,494	246	{C3, C6, C12, C13} {C3, C7, C10, C13, C14}	Empty
Pima Diabetes	8	768	28	{blp, pedigree, age} {times, glucose, pedigree} {glucose, blp, insulin, age}	Empty
Spambase	57	4,601	110	{will, report, you, credit, hp, george, meeting re, edu, (, !, average, total} {make, all, our, mail, report, free, you, credit, your george, technology, meeting, re, edu, !, average, total}	re, meeting, george you, !, total, edu
Wine	13	178	66	{Flavanoids, Color} {Proanthocyanins, Color} {MalicAcid, Alcalinity, Phenols}	Empty
Yeast	8	1,484	4	{mcg, alm, mit, vac}, {mcg, gvh, mit, vac} {mcg, gvh, alm, vac, nuc} {gvh, alm, mit, vac, nuc}	vac
Zoo	16	101	27	{eggs, aquatic, toothed, breathes, legs} {milk, aquatic, backbone, venomous, legs, catsize} {hair, eggs, aquatic, predator, breathes, fins, legs}	aquatic, legs

Table 4.6: UCI Data Sets with the Rule Importance Measures

Data set	No. Rules with Original Data	No. Rules by RIM	Sample Rules by Rule Importance Measure % indicates the Rule Importance
Abalone	($s = 0.1\%$, $c = 100\%$) 218	17	Viscera weight=0.1730 → Rings=9 [62.50%] Infant, Height=0.12, Length=0.5 → Rings=8 [18.75%] Female, Height=0.165, Diameter=0.48 → Rings=10 [12.50%]
Breast Cancer	($s = 1\%$, $c = 100\%$) 49,574	225	age30-39, tumor-size20-24, NoIrradiat → no-recurrence-events [100%] age50-59, menopause_premeno, degmalig_3 rightbreast → recurrence-events [100%] tumor-size30-34, degmalig_3, breast-quad_rightup → recurrence-events [100%]
Car	($s = 1\%$, $c = 100\%$) 341	9	BuyingPrice_v-high, Maintainance_v-high → Decision_unacceptable [100%] BuyingPrice_v-high, SizeLuggageBoot_small, Safety_med → Decision_unacceptable [100%]
Glass	($s = 0.5\%$, $c = 100\%$) 9,129	129	Si=72.19 → Type_2 [44.44%] Na=14.38 → Type_7 [33.33%] Na=13.48, Mg=3.74 → Type_1 [11.11%]
Heart	($s = 1\%$, $c = 100\%$) 71,534	237	maximum_heart_rate_179 → $class_0$ [61.40%] oldpeak_3.4 → $class_2$ [47.37%] age65, female, thal_normal → $class_0$ [3.51%] male, restingBloodPressure_130, no_exercise_induced_angina, no_major_vessels_colored_by_flourosopy → $class_0$ [1.75%]
Iris	($s = 1\%$, $c = 100\%$) 352	50	petalWidth1.1 → Iris-versicolor [75%] sepalWidth2.9 → Iris-versicolor [75%] petalLength1.9 → Iris-setosa [75%] sepalLength5.4, sepalWidth3.4 → Iris-setosa [50%]
Lympho-graphy	($s = 10\%$, $c = 100\%$) 75,731	43	changesinnode=lac.margin, bloflymphc=yes → metastases [51.02%] specialforms=vesicles, lymnodesenlar=4 → malign lymph [30.61%] blockofaffere=yes, bypass=no, earlyuptakein=no → metastases [7.48%]
Pendigits	($s = 0.5\%$, $c = 100\%$) 389	52	C3_0, C13_100 → Class 8 [31.30%] C3_0, C9_100, C12_100 → Class 0 [6.10%] C1_0, C12_50, C14_25 → Class 1 [0.41%]
Pima Diabetes	($s = 0.5\%$, $c = 100\%$) 429	126	Diabetes pedigree function_0.237 → Tested negative [60.71%] Plasma glucose concentration_187 → Tested positive [53.57%] Pregnant_twice, insulin_0, age_25 → Tested negative [3.57%]
Spambase	($s = 1\%$, $c = 100\%$) 37,374,343	2,190	you=0, re=0, !=0, average=1 → NotSpam [100%] !=0, captialCharacterLongest=2 →NotSpam [67.27%] george=0, re=0, edu=0, !=0, longest=3 → NotSpam[67.27]
Wine	($s = 1\%$, $c = 100\%$) 548	247	Nonflavanoid0.14 → $class_2$ [21.21%] Malic acid 1.64 → $class_1$ [18.18%] Nonflavanoid phenols0.53, Alcalinity of ash 21.00 → $class_3$ [10.61%] color intensity5.40, Hue 1.25 → $class_1$ [1.52%]
Yeast	($s = 0.2\%$, $c = 100\%$) 20,864	195	alm0.39, vac0.51 → ME3 [75%] alm0.51, vac0.51, gvh0.48 → CTY [50%] mcg0.43, nuc0.33 → NUC [25%] mcg0.46, vac0.51, nuc0.22 → CYT [25%]
zoo	($s = 10\%$, $c = 100\%$) 680,996	31	aquatic, 6 legs → Type 6 [100%] no eggs, 2 legs → Type 1 [66.67%] eggs, non breathes, non fin → Type 7 [7.41%]

4.4.3 Experiments on Geriatric Care Data Set

In this experiment, a sanitized geriatric care data set is used as our test data set. The attributes for this medical data set are listed in Table B.1 in Appendix B. This data set contains $8,547$ patient records with 44 symptoms and their survival status. This data set is an actual data set from Dalhousie University Faculty of Medicine to determine the survival status of a patient giving all the symptoms he or she shows. We use *survival status* as the decision attribute, and the 44 symptoms of a patient as condition attributes, which includes *education level, the eyesight, the age of the patient at investigation* and so on. [3] There is no missing value in this data set. Table 4.7 gives selected data records of this data set.

Table 4.7: Geriatric Care Data Set

edulevel	eyesight	...	trouble	livealone	cough	hbp	heart	...	studyage	sex	livedead
0.6364	0.25	...	0.00	0.00	0.00	0.00	0.00	...	73.00	1.00	0
0.7273	0.50	...	0.50	0.00	0.00	0.00	0.00	...	70.00	2.00	0
0.9091	0.25	...	0.00	0.00	0.00	1.00	1.00	...	76.00	1.00	0
0.5455	0.25	...	0.00	1.00	1.00	0.00	0.00	...	81.00	2.00	0
0.4545	0.25	...	0.00	1.00	0.00	1.00	0.00	...	86.00	2.00	0
0.2727	0.00	...	0.50	1.00	0.00	1.00	0.00	...	76.00	2.00	0
0.0000	0.25	...	0.00	0.00	0.00	0.00	1.00	...	76.00	1.00	0
0.8182	0.00	...	0.00	0.00	0.00	1.00	0.00	...	76.00	2.00	0
...	...	...	...	...	...	...	...	...	...	...	...

There are 12 inconsistent data entries in the medical data set. After removing these instances, the data contains $8,535$ records. [4]

Table 4.8 shows selected reduct sets among the 86 reducts generated by ROSETTA. All of these reducts contain the core attributes. For each reduct set, association rules

[3] Refer to Appendix B and [53] for details about this data set.

[4] Notice from our previous experiments that the core generation algorithm cannot return correct core attributes when the data set contains inconsistent data entries.

Table 4.8: Reduct Sets for the Geriatric Care Data Set after Preprocessing

No.	Reduct Sets
1	{edulevel,eyesight,hearing,shopping,housewk,health,trouble,livealone, cough,sneeze,hbp,heart,arthriti,eyetroub,eartroub,dental, chest,kidney,diabetes,feet,nerves,skin,studyage,sex}
2	{edulevel,eyesight,hearing,phoneuse,meal,housewk,health,trouble,livealon, cough,sneeze,hbp,heart,arthriti,evetroub,eartroub,dental, chest,bladder,diabetes,feet,nerves,skin,studyage,sex}
...	...
86	{edulevel,eyesight,hearing,shopping,meal,housewk,takemed,health, trouble,livealone,cough,tired,sneeze,hbp,heart,stroke,arthriti, eyetroub,eartroub,dental,chest,stomach,kidney,bladder,diabetes, feet,fracture,studyage,sex}

Table 4.9: Core Attributes for Geriatric Car Data Set

Core Attributes
eartroub, livealone, heart, hbp, eyetroub, hearing, sex, health, edulevel, chest, housewk, diabetes, dental, studyage

are generated with $support = 30\%, confidence = 80\%$. [5] There are 14 core attributes generated for this data set. They are *eartroub, livealone, heart, hbp, eyetroub, hearing, sex, health, edulevel, chest, housewk, diabetes, dental, studyage* as shown in Table 4.9.

Discussion

There are 218 rules generated and ranked according to their rule importance as shown in Table 4.10. We noticed there are 8 rules having importance of 100%. All attributes

[5]Note that the value of support and confidence can be adjusted to generate as many or as few rules as required.

Table 4.10: The Rule Importance for the Geriatric Care Data Set

No.	Selected Rules	Rule Importance
1	SeriousChestProblem → Dead	100%
2	SeriousHearingProblem, HavingDiabetes → Dead	100%
3	SeriousEarTrouble → Dead	100%
4	SeriousHeartProblem → Dead	100%
5	Livealone, HavingDiabetes, HighBloodPressure → Dead	100%
...	...	...
11	Livealone, HavingDiabetes, NerveProblem → Dead	95.35%
...	...	...
14	Livealone, OftenCough, HavingDiabetes → Dead	93.02%
...	...	...
217	SeriousHearingProblem, ProblemUsePhone → Dead	1.16%
218	TakeMedicineProblem, NerveProblem → Dead	1.16%

contained in these 8 rules are core attributes. These 8 rules are more important when compared to other rules. For example, consider rule No.5 and No.11. Rule No.11 has an importance measure of 95.35%. The difference between these two rules is that rule No.5 contains attribute *Livealone, HavingDiabetes, HighBloodPressure*, and rule No. 11 contains the first 2 attributes, and instead of *HighBloodPressure*, *NerveProblem* is considered to decide whether the patient will survive. Generally high blood pressure does affect people's health condition more than nerve problem in combination with the other 2 symptoms. Rule No.11 is more important than rule No.218 because in addition to the *NerveProblem*, whether a patient is able to take medicine by himself or herself is not as fatal as whether he or she has diabetes, or lives alone without care. With the same support and confidence, $2,626,392$ rules are generated from the original medical data set without considering reduct sets or rule templates. Our method efficiently extracts important rules, and at the same time provides a ranking for important rules.

We also performed experiments using Johnson's reduct generation algorithm [69] for rule

Table 4.11: Rules Generated by Johnson's Algorithm for the Geriatric Care Data Set

No.	Rules	Rule Importance According to Table 4.10
1	SeriousChestProblem → Dead	100%
2	SeriousHearingProblem, HavingDiabetes → Dead	100%
3	SeriousEarTrouble → Dead	100%
4	SeriousEyeTrouble → Dead	100%
5	SeriousHeartProblem → Dead	100%
6	Livealone, HavingDiabetes, HighBloodPressure → Dead	100%
7	VerySeriousHouseWorkProblem → Dead	100%
8	Sex_2 → Dead	100%
9	FeetProblem → Dead	96.51%
10	SeriousEyeSight → Dead	95.35%
11	Livealone, HavingDiabetes, NerveProblem → Dead	95.35%
12	TroublewithLife → Dead	81.40%
13	LostControlofBladder, HavingDiabetes → Dead	75.58%
14	Livealone, HighBloodPressure, LostControlofBladder → Dead	75.58%
15	HighBloodPressure, LostControlofBladder, NerveProblem → Dead	72.09%
16	Livealone, LostControlofBladder, NerveProblem → Dead	72.09%

generation based on one reduct with the minimum attributes. 16 rules are generated using this reduct [53] as shown in Table 4.11. The 8 rules with 100% importance in Table 4.10 are also generated. Although the reduct generated by Johnson's algorithm can provide all the 100% importance rules, the result does not cover other important rules. For example, rule No.14 in Table 4.10 implies that it is important for the doctors to pay attention to some patient who lives alone, coughs often and also has diabetes. This information is not included in Table 4.11 by just considering the rules generated by only one reduct.

The experimental results show that considering multiple reducts gives us more diverse view of the data set and the Rule Importance Measure provides a ranking of how important a rule is.

4.4.4 Comparison Experiments

Confidence is one of the interestingness measures discussed in Chapter 3. Given the antecedent of a rule existing in the data set, the confidence measures the probabilities of both the antecedent and the consequent of the rule appearing together in the data set. The higher the probability, the more interesting the rule is considered to be. Confidence is usually used to measure how frequently the items appear together in the data set, and how much associated one item is to the other item(s). Thus, if people are interested in how significant a rule is instead of how often the items contained in the rule appear together, a confidence measure cannot provide such knowledge. The Rule Importance Measure takes the semantic meaning of the data into consideration, and evaluates the significance of a rule through how significant the attributes are.

In order to show that the Rule Importance Measure is different from other existing measures on ranking the rules, e.g., confidence, we compare effects on ranking the rules from both the Rule Importance Measure and confidence measure.

We take the geriatric care data set as an example. The rules ranked with their importance are shown in Table 4.10. These rules are generated with the minimum confidence of 80%. We list the rules ranked by their confidence in Table 4.12. From Table 4.12 we can see that what the confidence measure considers to be interesting are not always important. For example, rule No. 4 and No. 5 have similar confidence, but intuitively, whether a patient has a serious heart problem is more important than whether he or she can walk

Table 4.12: Rules Ranked with Confidence for the Geriatric Care Data Set

No.	Selected Rules	Confidence	Rule Importance
1	TroublewithLife → Dead	85.87%	81.40%
2	VerySeriousHouseWorkProblem → Dead	84.77%	100%
3	TroublewithShopping → Dead	83.03%	41.86%
4	TroublewithGetPlacesoutofWalkingDistance → Dead	81.86%	16.28%
5	SeriousHeartProblem → Dead	81.66%	100%
6	TroublePrepareMeal → Dead	81.51%	69.77%
7	EyeTrouble → Dead	80.91%	95.35%
8	Sex_2 → Dead	80.87%	100%
9	SeriousEarTrouble → Dead	80.48%	100%
10	SeriousFeetProblem → Dead	80.83%	96.51%
11	TakeMedicineProblem, KidneyProblem → Dead	80.64%	13.95%
...	...	...	...
21	SeriousEyeTrouble → Dead	80.48%	100%
...	...	...	...
36	Livealone, OftenCough, HavingDiabetes → Dead	80.40%	93.02%
37	TakeMedicineProblem, LostControlBladder → Dead	80.39%	16.28%
38	SeriousHearingProblem, HavingDiabetes → Dead	80.39%	100%
...	...	...	...
125	SeriousHearingProblem, ProblemUsePhone → Dead	80.13%	1.16%
...	...	...	...
154	SeriousChestProblem → Dead	80.07%	100%
...	...	...	...
169	Livealone, HavingDiabetes, HighBloodPressure → Dead	80.05%	100%
...	...	...	...
177	Livealone, HavingDiabetes, NerveProblem → Dead	80.04%	95.35%
...	...	...	...
218	TakeMedicineProblem, NerveProblem → Dead	80.00%	1.16%

for a certain distance. When a patient has a heart problem, he or she normally would have trouble walking for long distances. As another example, rule No. 177 has a lower confidence, and therefore is not considered to be interesting. However, whether the patient has diabetes plays an important part in diagnosing diseases; this knowledge cannot be ignored. Comparison experiments between the Rule Importance Measure and support can be conducted similarly by ranking the rules with their support and rule importance, and compare the different effects they have on ranking the same set of rules. In comparison, Rule Importance Measure ranks rules containing important attribute(s) to be more significant. In certain applications, such as medical diagnosis, when the focus of knowledge discovery is on the important symptoms, the Rule Importance Measure can indeed help facilitate evaluating important knowledge.

4.5 Conclusions

We introduce a Rule Importance Measure which is an automatic and objective approach to extract and rank important rules. This measure is applied throughout the rule generation process. Although the rules we used in experiments in this chapter are rules with decision attributes on the consequent part, any forms of association rules can all be generated and ranked by this rule importance measure. The core attributes should be taken into consideration while choosing important and useful rules. By considering as many reduct sets as possible, we try to cover all representative subsets of the original data set. This measure can also be used jointly with other measures to facilitate the evaluation of the association rules.

Rough sets theory can help with selecting representative attributes from a given data set. By removing redundant attributes, only preserving representative attributes, we achieve representative rules. At the same time, the computation cost is lower comparing to rule generation with all the attributes.

During our experiments on actual data sets, we observed some interesting results. For the UCI breast cancer data set, we extract a rule with 100% importance that if the patient is in the age of 50 to 59, pre-menopause, with degmalig of 3 and the tumor is in the right breast, then the breast cancer belongs to a recurrence event. For the pima diabetes data

set, it is not necessary to consider the following rule as important that if a patient has been pregnant twice, the 2-hour serum insulin is 0, and she is 25 years old, her chance of getting diabetes is negative. For the spambase data set, one of the most important rules is when the word frequencies for "you", "re" and "!" are 0 in an email, and the average length of uninterrupted sequences of capital letters is 1, then this email is not considered possible to be a spam email. For the geriatric care data set, we found that given the same condition of a patient living alone and having lost control of bladder, high blood pressure brings more a severe effect to the patient than nerve problems [6].

Rule Importance Measures differentiate rules by indicating which rules are more important than other rules. Rule Importance Measures can be used in a variety of applications such as medical diagnosis, construction of spam filters, object labeling in criminology and so on. We will further demonstrate other possible applications in Chapter 7.

We observed a limitation for the Rule Importance Measure that when there is only one reduct for a data set, such as the UCI Car data set or the Breast Cancer data set, the Rule Importance Measure returns all the rules with the importance of 100%. The result is the same as rule generation for the data set itself. So, for a given data set, if there is only one reduct, the Rule Importance Measure does not differentiate the generated rules.

[6]Note that rules ranked as important may sometimes be tautological or non-unique. In such cases, the domain experts are needed for precise evaluations.

Chapter 5

Rules-As-Attributes Measure

Use of rough sets theory to select essential attributes that can represent the original data set is well known. A reduct is a subset of the original data set which contains the essential attributes. Decision rules generated from reducts can fully describe a data set. In this chapter, we introduce a new method of evaluating important rules by taking advantage of rough sets theory, the Rules-As-Attributes measure. We consider rules generated from the original data set as attributes in the new constructed decision table. Reducts generated from this new decision table contain essential attributes, which are the rules. Only important rules are contained in the reducts. Experiments on an artificial data set, UCI data sets and real-world data sets show that the Reduct Rules are more important, and this new method provides an automatic and effective way of extracting important rules.

5.1 Introduction

Rough sets theory [72] is commonly used for attribute selection in the decision making process. Efforts on applying rough sets theory to knowledge discovery in databases have focused on decision making, data analysis, discovering and characterizing the inter-data relationships, and discovering interesting patterns [73]. The decision table consists of condition attributes and decision attributes. As explained in Chapter 2, a reduct is a subset of condition attributes that can represent the whole data set. Traditionally reduct generation is designed to extract important condition attributes from a decision table. By

considering fewer attributes, the decision making process will become more efficient.

The association rule algorithm [3] is well known for discovering associations, e.g., shopping behaviours among transaction data. One of the main problems for association rule generations is that the number of rules generated is generally quite large; thus, it is very difficult to evaluate and rank these rules. In order to solve this problem, many novel approaches have been developed to extract more interesting rules. Rule templates [45] as one of the examples of the rule interestingness measures can be applied to extract appropriate rules towards certain applications. They are useful in decision making, recommender systems and other applications. The association rule algorithm can be used to extract rules from the decision table as well.

In this chapter we are interested in using rough sets theory to facilitate the association rule generation. We focus on how to use rough sets theory to discover important rules. The Rule Importance Measure introduced in Chapter 4 is also a rough set-based rule evaluation approach. The approach we will introduce in this chapter is different from the Rule Important Measure, although both measures consider the input data as a decision table. The Rule Importance Measure is applied through the rule generation procedure, the input of this measure is the original decision table, and the output is a set of rules ranked by their importance. The Rules-As-Attributes Measure takes any sets of rules as input, and it is to be used after the rules are generated. Such rules can be generated by various learning algorithms. The output of the Rules-As-Attributes Measure is a set of important rules, which is a subset of the original rule sets generated from the original data.

We utilize the concept of a reduct in a new perspective. Association rules are generated from the original decision table. Each rule is considered as a condition attribute in the new constructed decision table. The decision attributes are the original decision attributes. Therefore, a reduct of such a decision table represents the essential attributes, which are the most important rules that fully describe the decision. We call these rules *Reduct Rules*. The reduct rules contained by a reduct are therefore important, and all the other rules are not as important or as representative.

Related work on rough sets theory and rule discovery is discussed in Section 5.2. In Section 5.3 we introduce the *Reduct Rules* from the proposed Rules-As-Attributes measure. Experiments on an artificial data set, real-world data sets and UCI data sets are shown

in Section 5.4. Several observations and discussions of the experiments are included in Section 5.5. Conclusions for this chapter are discussed in Section 5.6.

5.2 Rough Sets Theory and Rule Discovery

We define the rule templates that are used in this chapter, and discuss previous work on using rough sets theory to facilitate rule discovery. Literature reviews on rough sets theory can be found in Chapter 2.

5.2.1 Defining Rule Template

Because our interest is to make decisions or recommendations based on the condition attributes, we are looking for rules with only decision attributes on the consequent part. Therefore, we specify the following 2 rule templates to extract rules we want as shown by Template 5.1, and to subsume rules as shown by Template 5.2.

$$\langle Attribute_1, Attribute_2, \ldots, Attribute_n \rangle \rightarrow \langle DecisionAttribute \rangle \tag{5.1}$$

Template 5.1 specifies only decision attributes can be on the consequent part of a rule, and $Attribute_1$, $Attribute_2$, ..., $Attribute_n$ lead to a decision of $DecisionAttribute$, as shown by Template 5.1.

We specify the rules to be removed or subsumed using Template 5.2. For example, given rule

$$\langle Attribute_1, Attribute_2 \rangle \rightarrow \langle DecisionAttribute \rangle \tag{5.2}$$

the following rules

$$\langle Attribute_1, Attribute_2, Attribute_3 \rangle \rightarrow \langle DecisionAttribute \rangle \tag{5.3}$$

$$\langle Attribute_1, Attribute_2, Attribute_6 \rangle \rightarrow \langle DecisionAttribute \rangle \tag{5.4}$$

can be removed because they are subsumed by Template 5.2.

We use the artificial car data set that is shown in Table 5.3 as an example to further explain how to define proper templates. Since we are interested in predicting the mileage

of a car based on the model of a car, the number of doors, the compression, the weight as well as other factors related to a car, we would like to extract rules which have the decision attribute "mileage" on the consequent part of the rules. Therefore we specify the template for desired rules as shown in Eq. 5.5

$$\langle model, cyl, \ldots, weight \rangle \rightarrow \langle mileage \rangle. \tag{5.5}$$

And if a rule

$$\langle JapanCar, weight\_medium \rangle \rightarrow \langle mileage\_High \rangle \tag{5.6}$$

is generated, rules such as Eq. 5.7

$$\langle JapanCar, trans\_manual, weight\_medium \rangle \rightarrow \langle mileage\_High \rangle \tag{5.7}$$

is subsumed, because this rule can be deduced by the previous rule.

5.2.2 From Reduct to Rule Generation

As discussed in Section 4.2, there have been other contributions on applying rough sets theory to rule discovery (e.g., [33, 43]). Another relevant work is that of Szczuka [84] who proposed a new method of constructing a classification system with a combination of a rule based system and neural networks. Reducts are generated from the original data using rough sets theory; then, rules (rule generation functions depend on the applications) are generated according to the attributes in the reducts. These rules are used as input for a neural network based classifier. The classifier constructed is smaller and simpler than the rough sets classifier, and the weights of the neural network imply the importance of particular rules.

Still, little effort to date has been expended on applying rough sets theory to association rules generation. In fact, rough sets can be used to determine whether there is redundant information in the data and whether we can find the essential data needed for our applications. Since the rough sets method can help to generate representative attributes, we expect fewer rules will be generated due to fewer attributes. And the rules will be as significant as the rules generated without using the rough sets approach.

Rules generated from the original data set can be used to represent original knowledge. After a reduct is generated, a rule based on this reduct is generated in the form such that

the antecedents of a rule are from the value of condition attributes in the reduct set, and the consequents of a rule are from the value of decision attributes from the original data set. Association rule generations also return rules with certain support and confidence.

5.3 Discovering Important Rules - Reduct Rules

As discussed in Chapter 4, a general problem with rule generation is how to automatically extract important rules from the large number of generated rules. In this section we propose a new approach of selecting important rules based on rough sets theory.

Let us consider the concept of a reduct. A reduct of a decision table contains attributes that can fully represent the original knowledge. When a reduct is given, rules extracted based on this reduct are representative of the original decision table. These representative rules are therefore considered more important than the rules generated without using the reduct. A reduct contains the most representative and important condition attributes of a decision table. Based on this intuition, each of the individual rules among the generated rules sets can be considered as a condition attribute in a decision table. The reduct extracted for such decision tables would contain representative and important attributes, which are the rules. Since the generation of reduct is an automatic process, we can use such an approach to discover important rules from a set of generated rules automatically.

5.3.1 Reconstructing Decision Tables by Considering Rules as Attributes

We consider a decision table $T = (U, C, D)$, where $U = \{u_0, u_1, ..., u_{m-1}\}$ is a set of records in the table, $C = \{c_0, c_1, ..., c_{p-1}\}$ is a set of the condition attributes and D is a set of the decision attributes. Let us consider decision tables with one decision attribute. A set of rules R is generated from this table T, where $R = \{Rule_0, Rule_1, ..., Rule_{n-1}\}$. The new decision table is constructed as follows.

We construct a new decision table $A_{m\times(n+1)}$, where each record from the original decision table $u_0, u_1, ..., u_{m-1}$ is the row, and the columns of this new table consists of $Rule_0, Rule_1, ..., Rule_{n-1}$ and the decision attribute. We say a rule can be **applied** to a

record in the decision table if both the antecedent and the consequent of the rule appear together in the record, which can also be interpreted as whether a rule can classify the record correctly. For each $Rule_j$ ($j \in [0, ..., n-1]$), we assign 1 to cell $A[i, j]$ ($i \in [0, ..., m-1]$) if the rule $Rule_j$ can be applied to the record u_i. We set 0 to $A[i, j]$ otherwise. The decision attribute $A[i, n]$ ($i \in [0, ..., m-1]$) remains the same as the original values of the decision attribute in the original decision table. Eq. 5.8 shows the conditions for the value assignments of the new decision table.

$$A[i,j] = \begin{cases} 1, & \text{if } j < n \text{ and } Rule_j \text{ can be applied to } u_i \\ 0, & \text{if } j < n \text{ and } Rule_j \text{ cannot be applied to } u_i \\ d_i, & \text{if } j = n \text{ and } d_i \text{ is the corresponding decision attributes for } u_i \end{cases} \tag{5.8}$$

where $i \in [0, ..., m-1], j \in [0, ..., n-1]$.

The following example explains how to construct the new decision table using the above proposed approaches and Eq. 5.8. Let us consider a decision table as shown in Table 5.1. c_1, c_2, c_3 are the condition attributes, and D is the decision attributes.

Table 5.1: Sample Decision Table

c_1	c_2	c_3	D
1	0	1	1
1	1	0	1
0	0	1	0

Suppose there are 2 rules generated based on Table 5.1, and the rule set is $R = \{Rule_0, Rule_1\}$. $Rule_0$ specifies "if $c_1 = 1$, then $D = 1$"; $Rule_1$ specifies "if c_2=1 and c_3=0, then $D = 1$". In this example, $m = 3$ which stands for the number of rows in the original decision table; $n = 2$ which stands for the number of rules in the rule set. A new decision table for ranking the important rules can therefore constructed as $A_{3\times3}$, the condition attributes in the new decision table are $Rule_0$ and $Rule_1$, and decision attribute is D, which comes from the original decision table. According to Eq. 5.8, for condition

attribute $Rule_0$, $A[0,0] = 1$ because $Rule_0$ can correctly classify the record in the first row in Table 5.1, $A[1,0] = 1$ because $Rule_0$ can correctly classify the record in the second row from Table 5.1; but $A[2,0] = 0$ because $Rule_0$ cannot be applied to the record in the third row from Table 5.1 since $c_1 = 0$ instead of 1. Therefore, the cells from the first column in Table A are assigned as

$Rule_0$
1
1
0

According to Eq. 5.8, the cells from the second column in Table A are assigned as

$Rule_1$
0
1
0

With the original decision attributes unchanged from Table 5.1, and the two columns for condition attributes, the new decision table $A_{3\times3}$ is constructed as shown in the following Table 5.2.

Table 5.2: New Decision Table $A_{3\times3}$

$Rule_0$	$Rule_1$	D
1	0	1
1	1	1
0	0	0

This new decision table is then used as the input decision table for discovering important rules.

5.3.2 Reduct Rules and Core Rules

We further define *Reduct Rule Set* and *Core Rule Set*.

Definition 3 Reduct Rule Set. We define a reduct generated from the new decision table A as the **Reduct Rule Set**. A *Reduct Rule Set* contains *Reduct Rules*.

The *Reduct Rules* are representative rules that can fully describe the decision attribute.

Definition 4 Core Rule Set. We define the intersection of all the *Reduct Rule Sets* generated from this new decision table A as the *Core Rule Set*. A *Core Rule Set* contains **Core Rules**.

The *Core Rules* are contained in every *Reduct Rule Set*.

By considering rules as attributes, reducts generated from the new decision table contain all the important attributes, which represent the important rules generated from the original data set; and it excludes the less important attributes. Core attributes from the new decision table A contain the most important attributes, which represent the most important rules.

5.3.3 Evaluation

A reduct of a data set contains a set of representative and important attributes that can determine the decision attributes. The proposed *Reduct Rules* are of interest and can be used to discover representative and important rules.

Since the Rule Importance Measure in Chapter 4 (see also [54]) provides a rank of different important rules, we use the Rule Importance Measure to evaluate our experimental results in Section 5.4.

Note that the Rule Importance Measure ranks rules generated from multiple reducts, which implies that these ranked rules all contain reduct attributes from the original data sets. However, the *Reduct Rules* are extracted from generated rules based on all the attributes of the original data sets. Therefore, if a *Reduct Rule* can be found in the rule sets ranked by the rule importance, it implies that this is a rule containing the attributes in the reduct and thus is more important than rules that are not ranked by the Rule Importance Measure.

5.4 Experiments

In this section, we perform experiments on an artificial car data set, a real world geriatric care data set, 10 UCI data sets and a marketing data set to show that the *Reduct Rules* are more important.

5.4.1 Procedures

Figure 5.1 illustrates our experimental procedure.

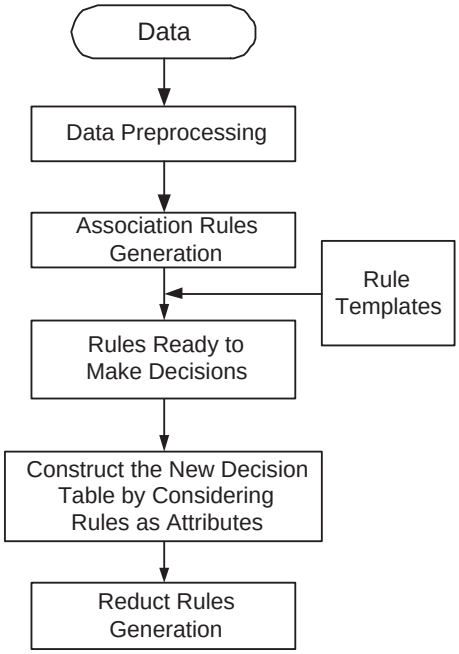

Figure 5.1: Experiment Procedure

In our experiments, we consider each data set as a transaction set. First during the data preprocessing step, the inconsistent data instances and the data instances containing missing attribute values are processed. The core algorithms require a consistent data set. Therefore in our experiments, the inconsistent data instances are considered as noise and are removed during the data preprocessing stage. Inconsistency exists in a decision table when two or more data instances containing the same condition attribute values

but different decision attribute values. These data instances must be removed. We first sort the whole data set according to the condition attributes, excluding the decision attributes. Then we select data instances that contain the same condition attributes values, but different decision attributes values. These data instances are inconsistent and they are removed during this stage. Discretization, such as equal frequency binning or the entropy algorithm [69], is also applied during this stage if necessary. Core attributes are generated at the end of the data preprocessing stage.

The apriori [15] association rule algorithm is then applied to generate association rules for each data set. Since our interest is to make decisions, we use the rule templates defined in Section 5.2.1 to generate only rules with decision attributes on the consequent part, and to remove subsumed rules. The new decision table is constructed by using these association rules as condition attributes. Note that there may be inconsistencies existing in the new decision table; therefore, the data instances that are inconsistent have to be removed.

We use Johnson's Reduct generation algorithm in ROSETTA [69] on the new decision table to generate *Reduct Rules*. Other reduct generation approaches may also be applied at this step.

We first apply this experimental procedure to two data sets. The first data set, the car data set, is a small artificial data set designed to illustrate in detail how to generate *Reduct Rules*. The second data set, geriatric care data, is an actual data set from Dalhousie University Medical School [53] to determine the survival status of a patient. It is used to illustrate that the methods we devised can scale to larger data sets. We then demonstrate the utility of our method on UCI [21] data sets and a real-world marketing data set [34].

5.4.2 Car Data Set

We first explain in detail our method of considering rules as attributes using an artificial data set about cars [39], shown in Table 5.3, which is used to decide the mileage of different cars. This data set contains 14 records, and 8 condition attributes.

There is no inconsistent data or incomplete data existing in this data set. Rule templates defined in Section 5.2.1 are applied, e.g., rules with only decision attribute *mileage* on the consequent part are generated; and subsumed rules are removed. There are 19 rules

Table 5.3: Artificial Car Data Set

make_model	cyl	door	displace	compress	power	trans	weight	mileage
USA	6	2	Medium	High	High	Auto	Medium	Medium
USA	6	4	Medium	Medium	Medium	Manual	Medium	Medium
USA	4	2	Small	High	Medium	Auto	Medium	Medium
USA	4	2	Medium	Medium	Medium	Manual	Medium	Medium
USA	4	2	Medium	Medium	High	Manual	Medium	Medium
USA	6	4	Medium	Medium	High	Auto	Medium	Medium
USA	4	2	Medium	Medium	High	Auto	Medium	Medium
USA	4	2	Medium	High	High	Manual	Light	High
Japan	4	2	Small	High	Low	Manual	Light	High
Japan	4	2	Medium	Medium	Medium	Manual	Medium	High
Japan	4	2	Small	High	High	Manual	Medium	High
Japan	4	2	Small	Medium	Low	Manual	Medium	High
Japan	4	2	Small	High	Medium	Manual	Medium	High
USA	4	2	Small	High	Medium	Manual	Medium	High

generated by the apriori algorithm with $support = 1\%, confidence = 100\%$[1], as shown in Table 5.4.

New Decision Table

The new decision table $A_{14\times 20}$ is constructed by using the 19 rules as condition attributes, and the original decision on the mileage as the decision attribute. For each rule we check whether it can be applied to the 19 records. For example, $Rule_0$,

$$\text{USACar}, \text{Displace\_Medium}, \text{Weight\_Medium} \rightarrow \text{Mileage\_Medium} \tag{5.9}$$

[1]The values of support and confidence can be adjusted to control the number of rules generated. For the rest of our experiments, we set the support and confidence during rule generations for each data set to obtain a certain amount of rules.

Table 5.4: Rule Set Generated by the Car Data Set

No.	Association Rules
0	USACar,Displace_Medium,Weight_Medium → Mileage_Medium
1	USACar, Compress_Medium → Mileage_Medium
2	USACar,Power_High → Mileage_Medium
3	Cyl_6 → Mileage_Medium
4	Door_4 → Mileage_Medium
5	Displace_Medium, Compress_High, Weight_Medium → Mileage_Medium
6	Displace_Medium, Power_High → Mileage_Medium
7	Compress_Medium, Power_High → Mileage_Medium
8	Trans_Auto → Mileage_Medium
9	JapanCar → Mileage_High
10	Cyl_4, Displace_Medium, Compress_High → Mileage_High
11	Cyl_4, Compress_High, Power_High → Mileage_High
12	Displace_Small, Compress_Medium → Mileage_High
13	Displace_Small, Power_High → Mileage_High
14	Displace_Small, Trans_Manual → Mileage_High
15	Displace_Medium, Compress_High, Power_Medium → Mileage_High
16	Compress_High, Trans_Manual → Mileage_High
17	Power_Low → Mileage_High
18	Weight_Light → Mileage_High

can be applied to the first record, because both the antecedent *USACar*, *Displace_Medium*, *Weight_Medium* and the consequent *Mileage_Medium* appear in the rule. Therefore, we assign $A[0,0] = 1$. $Rule_0$ can be applied to the second record as well. We assign $A[1,0] = 1$. However, $Rule_0$ cannot be applied to the third record, because the value for "displace" is "small" instead of "medium". Therefore $A[2,0] = 0$. Table 5.5 gives the new constructed decision table for car data set. Note that we set "Mileage_Medium" to be 0, and "Mileage_High" to be 1.

Table 5.5: New Decision Table for the Car Data Set

$Rule_0$	$Rule_1$	$Rule_2$	...	$Rule_{15}$	$Rule_{16}$	$Rule_{17}$	$Rule_{18}$	Mileage
1	0	1	...	0	0	0	0	0
1	1	0	...	0	0	0	0	0
0	0	0	...	0	0	0	0	0
1	1	0	...	0	0	0	0	0
1	1	1	...	0	0	0	0	0
1	1	1	...	0	0	0	0	0
1	1	1	...	0	0	0	0	0
0	0	0	...	1	1	0	1	1
0	0	0	...	0	1	1	1	1
0	0	0	...	0	0	0	0	1
0	0	0	...	0	1	0	0	1
0	0	0	...	0	0	1	0	1
0	0	0	...	0	1	0	0	1
0	0	0	...	0	1	0	0	1

Table 5.6: Reduct Rules for the Car Data Set

No. in Table 5.4	Reduct Rules	Rule Importance
9	JapanCar → Mileage_High	100%
16	Compress_High, Trans_Manual → Mileage_High	75%

There is no inconsistency in this new decision table. The core rule set generated by the core algorithm is empty. Johnson's Reduct generation algorithm generates one reduct, $\{Rule_9, Rule_{16}\}$. The *Reduct Rules* for the car data set is shown in Table 5.6.

Evaluation

The Rule Importance Measure provides a way to evaluate whether the reduct rules are more important. Table 5.7 shows the Rule Importance for the car data set. Core attributes from the original data set are generated by the core algorithm. The core for this data set are *make_model*, and *trans* as shown earlier in Table 4.3. From Table 5.7 we can see that $Rule_9$ and $Rule_{16}$ have the rule importance of 100%, and 75% respectively.

We also observe that, in $Rule_9$, *JapanCar* is the core attribute value, in $Rule_{16}$, *Trans_Manual* is the core attribute value.The *Reduct Rules* all contain core attributes.

Discussion

This example shows that by considering rules as attributes and constructing a new decision table, the rules in the reduct are important rules, and are representative knowledge of the original data set. Therefore the *Reduct Rules* could be considered as important knowledge discovered from the original data.

5.4.3 Experiment on the Geriatric Data

A sanitized geriatric care data set is tested. This data set contains 8547 patient records with 44 symptoms and their survival status. The data set is used to determine the survival status of a patient giving all the symptoms he or she shows. We use *survival status* as the decision attribute, and the 44 symptoms of a patient as condition attributes, which includes *patients' education level, the eyesight, the age of the patient at investigation, the sex of the patient* and so on[2]. There are no missing values in this data set. Table 6.5 gives selected data records of this data set.

We first check for inconsistency in this data set and 12 inconsistent data records are removed from this data set. There are 86 reducts generated for this geriatric data set by the genetic algorithm in ROSETTA. The apriori algorithm [15] is then used to generate 86 rule sets for each reduct with $support = 30\%, confidence = 80\%$. Rule templates are applied in the rule generation as well, e.g., extracting only rules with decision attribute

[2]Refer to [53] for details about this data set.

Table 5.7: Rule Importance for the Car Data Set

No. in Table 5.4	Rules	Rule Importance
9	JapanCar → Mileage_High	100%
8	Trans_Auto → Mileage_Medium	100%
16	Compress_High, Trans_Manual → Mileage_High	75%
1	USACar, Compress_Medium → Mileage_Medium	75%
14	Displace_Small, Trans_Manual → Mileage_High	50%
3	Cyl_6 → Mileage_Medium	50%
0	USACar, Displace_Medium, Weight_Medium → Mileage_Medium	25%
17	Power_Low → Mileage_High	25%
2	USACar, Power_High → Mileage_Medium	25%
7	Compress_Medium, Power_High → Mileage_Medium	25%
12	Displace_Small, Compress_Medium → Mileage_High	25%
4	Door_4 → Mileage_Medium	25%
18	Weight_Light → Mileage_High	25%

livedead on the consequent part and removing subsumed rules. For example, in the rule set, a rule shown as Eq. 5.10 exists

$$SeriousChestProblem \rightarrow Death \tag{5.10}$$

the following rule is removed because it is subsumed.

$$SeriousChestProblem, TakeMedicineProblem \rightarrow Death \tag{5.11}$$

218 unique rules are generated over these 86 reducts. These rules as well as their rule importance are shown in Table 5.9. Among these 218 rules, 87 rules have rule importance of no less than 50% , 8 of which have rule importance of 100%. All the rules with rule importance of 100% contain only core attributes.

Table 5.8: Geriatric Care Data Set

edulevel	eyesight	hearing	health	trouble	livealone	cough	hbp	heart	stroke	...	sex	livedead
0.6364	0.25	0.50	0.25	0.00	0.00	0.00	0.00	0.00	0.00	...	1	0
0.7273	0.50	0.25	0.25	0.50	0.00	0.00	0.00	0.00	0.00	...	2	0
0.9091	0.25	0.50	0.00	0.00	0.00	0.00	1.00	1.00	0.00	...	1	0
0.5455	0.25	0.25	0.50	0.00	1.00	1.00	0.00	0.00	0.00	...	2	0
0.4545	0.25	0.25	0.25	0.00	1.00	0.00	1.00	0.00	0.00	...	2	0
0.2727	0.00	0.00	0.25	0.50	1.00	0.00	1.00	0.00	0.00	...	2	0
0.0000	0.25	0.25	0.25	0.00	0.00	0.00	0.00	1.00	0.00	...	1	0
0.8182	0.00	0.50	0.00	0.00	0.00	0.00	1.00	0.00	0.00	...	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...

The core attributes for this data set are *eartrouble, livealone, heart, highbloodpressure, eyetrouble, hearing, sex, health, educationlevel, chest, housework, diabetes, dental, studyage* as shown earlier in Table 4.9. The new decision table $A_{8535\times 219}$ is constructed by using the 218 rules [3] as condition attributes, and the original decision attribute as the decision attribute. Note that after reconstructing the decision table, we must check for inconsistency again before generating reduct rules for this table. After removing the inconsistent data records, there are 5709 records left in the new decision table. The core rule set is empty. We use Johnson's reduct generation algorithm on this table $A'_{5709\times 219}$ and the reduct rule set is $\{Rule_0, Rule_1, Rule_3, Rule_5, Rule_{19}, Rule_{173}\}$. We show these rules in Table 5.10.

Evaluation

From Table 5.10 we can see that the reduct rule sets contain 6 rules. There are 4 rules judged to be the most important. The rule importance for $Rule_0$, $Rule_1$, $Rule_3$ and $Rule_5$ are all 100%. $Rule_{19}$ has the importance of 82.56%, but is still more important than most

[3] There are $1,615$ rules generated by the apriori algorithm from the original data set with $support = 30\%, confidence = 80\%$, after applying the rule template. We can circumvent problems inherent in considering all 1615 generated rules using the 218 unique rules that are derived from the 86 reducts obtained by ROSETTA's genetic algorithm.

Table 5.9: Rule Importance for the Geriatric Care Data

No.	Selected Rules	Rule Importance
0	SeriousHeartProblem → Death	100%
1	SeriousChestProblem → Death	100%
2	SeriousHearingProblem, HavingDiabetes → Death	100%
3	SeriousEarTrouble → Death	100%
4	SeriousEyeTrouble → Death	100%
5	Sex_Female → Death	100%
...	...	...
10	Livealone, HavingDiabetes, NerveProblem → Death	95.35%
...	...	...
216	SeriousHearingProblem, ProblemUsePhone → Death	1.16%
217	TakeMedicineProblem, NerveProblem → Death	1.16%

of the 218 rules.

5.4.4 Experiments on UCI Data Sets and a Marketing Data Set

We experiment on selected UCI data sets [21] (as shown in Appendix C) and a marketing data set described below. In Table 5.11, we list for each data set, the name of the data set, the number of condition attributes, the number of instances it contains; the number of reducts returned by the ROSETTA genetic algorithm, sample reducts; and core attributes.

A. Abalone Data 17 rules are ranked by the Rule Importance Measure (of Chapter 4) ranging from 6.25% to 62.50%.

C. Car Data We first use Hu's core algorithm (discussed in Chapter 2) to generate core attributes, and all the condition attributes are the core attributes. There is only one reduct generated for this data set, and the reduct contains all the core attributes. 9 rules are ranked by the Rule Importance Measure all with 100% importance.

Table 5.10: Reduct Rules for the Geriatric Care Data

No. in Table 5.9	Reduct Rules	Rule Importance
0	SeriousHeartProblem → Death	100%
1	SeriousChestProblem → Death	100%
3	SeriousEarTrouble → Death	100%
5	Sex_Female → Death	100%
19	Livealon, OftenSneeze, DentalProblems, HavingDiabetes → Death	82.56%
173	ProblemHandleYourOwnMoney → Death	27.91%

D. Glass Data 129 rules are ranked by the Rule Importance Measure ranging from 5.56% to 44.40%.

F. Iris Data We apply the association rules algorithm [3] with rule templates, and there are 50 rules generated, which are ranked by the Rule Importance Measure ranging from 50.00% to 75.00%.

G. Lymphography Data 147 reducts are generated from this data set. 43 rules are ranked by Rule Importance Measure ranging from 1.36% to 51.02%.

H. Pendigits Data 52 rules are ranked by the Rule Importance Measure ranging from 0.41% to 31.30%.

I. Pima Indians Diabetes Data 126 rules are ranked by the Rule Importance Measure ranging from 3.57% to 60.71%.

K. Wine Recognition Data 247 rules are ranked by the Rule Importance Measure ranging from 1.52% to 21.21%.

L. Yeast Data 195 rules are ranked by the Rule Importance Measure ranging from 25.00% to 75.00%.

M. Zoo Data 31 rules are ranked by the Rule Importance Measure ranging from 3.70% to 100.00%.

K. Marketing Data This data set was collected from questionnaires filled in by shopping mall customers in the San Francisco Bay area [34]. The data is used to predict the annual income of each household from the 13 condition attributes, including the customer's sex, marital status, age, education, occupation, period of living in the local area, dual incomes if married, number of people living in the household, number of people in the household under age 18, the status of the household, the type of the home, the ethnic classifications and the languages spoken in the house. There are 8993 data instances. After removing inconsistencies and missing attribute values, there are 5625 data instances. All the condition attributes are core attributes. There is only one reduct generated for this data set, and the reduct contains all the core attributes. 102 rules are ranked by the Rule Importance Measure all with 100% importance.

Evaluations

In Table 5.12, we list the number of rules generated using the original data set with certain support and confidence values, and with the rule templates; the size of the new decision table with the same number of rows as the original data, and the number of columns are from the number of rules plus the original decision attributes, where the values in this new decision table are assigned according to Eq. 5.8. We also list the *Reduct Rules* returned by Johnson's reduct generation from ROSETTA, shown as "R" followed by the rule number; the results of the core rules generated by Algorithm 2 from Chapter 2 are also included. For each *Reduct Rule* in each data set, we show the Rule Importance Measures.

From Table 5.12, we notice that a majority of reduct rules have high rule importance, although there exist rules with lower importance measures as well.

5.5 Observations

From the Reduct Rules generation results in Section 5.4, we make the following observations.

Table 5.11: UCI Data Sets and Marketing Data

Data Set	Condition Attributes	No. of Instances	No. of Reducts	Sample Reducts	Core Attributes
Abalone	8	4,177	16	{WholeWeight, ShuckedWeight, ShellWeight} {Height, WholeWeight, ShuckedWeight, VisceraWeight} {Sex, Length, Height, WholeWeight, ShellWeight}	Empty
Car	6	1,728	1	{buying, maint, doors, persona, lug_boot, safety}	buying, maint, doors, persona, lug_boot, safety
Glass	9	214	21	{RI, Al} {Na, Si} {RI, Na, Mg} {Na, Mg, K, Fe}	Empty
Iris	4	150	4	{sepalLength, sepalWidth, petalLength} {sepalLength, petal Length, petalWidth} {sepalWidth, petalLength, petalWidth} {sepalLength, sepalWidth, petalWidth}	Empty
Lympho-graphy	18	148	147	{blockofaffere, hangesinnode, changesinstru, specialforms, dislocationof, noofnodesin}	Empty
Pendigits	16	7,494	246	{C3, C6, C12, C13} {C3, C7, C10, C13, C14}	Empty
Pima Diabetes	8	768	28	{blp, pedigree, age} {times, glucose, pedigree} {glucose, blp, insulin, age}	Empty
Wine	13	178	66	{Flavanoids, Color} {Proanthocyanins, Color} {MalicAcid, Alcalinity, Phenols}	Empty
Yeast	8	1,484	4	{mcg, alm, mit, vac}, {mcg, gvh, mit, vac} {mcg, gvh, alm, vac, nuc} {gvh, alm, mit, vac, nuc}	vac
Zoo	16	101	27	{eggs, aquatic, toothed, breathes, legs} {milk, aquatic, backbone, venomous, legs, catsize} {hair, eggs, aquatic, predator, breathes, fins, legs}	aquatic, legs
Marketing Data	13	5625	1	{sex,marital,age,education, occupation,year,dullincom,persons, persons18,household,home, ethic,language}	sex, marital, age, education, occupation,year, dullincom,persons, persons18,household, home,ethnic,language

Table 5.12: Reduct Rules for UCI Data Sets and Marketing Data

Data Set	No. of Decision Table Rules (support, confidence)	Decision Table Row × Column	Reduct Rules by Johnson's from ROSETTA	Sample Reduct Rules & Rule Importance (Indicated by %)	Core Rules
Abalone	20 (s=0.1%, c=100%)	$4,177 \times 21$	R1,R2,R3,R4, R5,R6,R7,R8, R9,R14,R15,R16, R17,R18,R19,R20 (16 Reduct Rules)	R1: VisceraWeight=0.0065 → Rings=4 [62.50%] R3: Length=0.245 → Rings=4[43.75%] R19: Height=0.17, Diameter=0.480 → Rings=10 [18.75%] R20:Height=0.195,Diameter=0.54→ Ring=11[18.75%]	Empty
Car	9 (s=1%, c=100%)	$1,728 \times 10$	Empty	N/A	Empty
Glass	138 (s=0.5%, c=100%)	214×139	R21,R49,R50,R57,R62, R65,R67,R69,R71,R72, R73,R74,R75,R76,R80, R81,R82,R85,R86,R87, ... R127,R128,R131 (45 Reduct Rules)	R21:Al=1.12 → Type=1[33.33%] R81: Si=72.19 → Type=2 [44.44%] R86: Si=72.83 → Type=2 [44.44%] R87: Si=72.87 → Type=2 [44.44%] R123: Na=14.95 → Type=7 [33.33%] R127: Mg=0, Fe=0.09 → Type=7 [5.56%] R128: Al=1.99→Type=7 [33.33%] R131: K=0, Cal=8.67 → Type=7 [N/A]	Empty
Iris	50 (s=1%, c=100%)	150×51	R9,R17,R19, R20,R21,R22 (6 Reduct Rules)	R9: sepalWidth3.5 →Iris-setosa [75.00%] R17: petalLength1.7 → Iris-setosa [75.00%] R21: petalWidth0.3 → Iris-setosa [75.00%] R22: petalWidth0.4 → Iris-setosa [75.00%]	Empty
Lympho-graphy	43 (s=10%, c=100%)	148×44	R30,R31, R37,R38 (4 Reduct Rules)	R30: changesinlym=oval, specialforms=vesicles, blockofaffere=no → malign lymph [16.33%] R38: specialforms=vesicles, dislocationof=yes, blockofaffere=no → malign lymph [17.01%]	Empty
Pendigits	74 (s=0.5%, c=100%)	7494×75	R14,R16,R17,R18,R22, R25,R26,R27,R28,R29, R30,R31,R32,R33,R34, ... R65,R66,R67,R68 (49 Reduct Rules)	R22: C1_0,C12_51 → Class=1 [10.57%] R25: C7_100, C8_100, C12_51 → Class=1 [2.85%] R31: C9_0, C14_0 → Class=2 [5.28%] R64: C3_0, C13_100 → Class=8 [31.30%] R65: C5_100, C8_0 → Class=8 [9.35%] R67: C8_0, C9_0, C13_100 → Class=8 [3.66%]	Empty
Pima Diabetes	134 (s=0.5%, c=100%)	768×135	R125,R126,R127, R128,R129,R130, R132,R133,R134 (9 Reduct Rules)	R127: glucose=168 → Tested Positive [53.57%] R128: glucose=181 → Tested Positive [53.57%] R130: blp=78, age=31 → Tested Positive [17.86%] R132: insulin=0, BMI=32.0 → Tested Positive [7.14%]	Empty
Wine	247 (s=1%, c=100%)	178×248	R1,R2,R3,R5,R6,R7 R8,R9,R11,R12,R14,R15, R16,R17,R20,R21,R24, ... R222,R224,R225,R233 (68 Reduct Rules)	R1: Nonfla=0.20, → Class=1 [21.21%] R7: Nonfla=0.31 → Class=1 [21.21%] R16: MalicAcid=1.77 → Class=1 [18.18%] R221: Hue=0.56 → Class=3 [16.67%] R222: Hue=0.57 → Class=3 [16.67%] R233: ODDiluted=1.33 → Class=3 [16.67%]	Empty
Yeast	209 (s=0.2%, c=100%)	1453×210	R47,R91,R92,R93,R94, R95,R96,R97,R98,R99, R100,R101,R102,R103, ... R195,R197,R199,R200 (106 Reduct Rules)	R47: gvh=0.48,alm=0.51,nuc=0.27 → CYT [50.00%] R95: mcg=0.33,vac=0.51 → NUC [75.00%] R103: mcg=0.42,alm=0.50 → NUC [50.00%] R107: mcg=0.47,vac=0.51 → NUC [75.00%] R169: gvh=0.62, vac=0.50 → MIT [75.00%] R200: alm=0.36,mit=0.26 → ME3 [50.00%]	Empty
Zoo	65 (s=10%, c=100%)	59×66	R22,R36,R55 (3 Reduct Rules)	R22: milk=1→Type=1 [33.33%] R36: hair=0, legs=2 →Type=2[40.74%] R55: aquatic=0, legs=6 → Type=6 [100.00%]	Empty
Marketing Data	102 (s=1%, c=100%)	5625×103	Empty	N/A	Empty

- The number of *Reduct Rules* is always less than the number of rules generated with the same support, confidence and the same rule templates. Since the *Reduct Rules* are rules generated based on the definitions of the reduct in rough sets theory, these rules are sufficient to describe the decision attributes in the original decision table.

- The *Core Rule Set* is always empty. This means that none of the *Reduct Rules* is contained by all the *Reduct Rule* sets.

- For UCI Car data set and Marketing data set, the *Reduct Rule* sets from ROSETTA are empty as shown in Table 5.12. This is because there is only one decision attribute value that exists in the new decision table after removing the inconsistencies. Therefore there is no subset of "condition attributes", which are the *Reduct Rules*, that can differentiate different concepts. It is also interesting to notice that all the condition attributes for these data sets are core attributes, and there is only one reduct for the data as shown in Table 5.11.

- There exist *Reduct Rules* that are not ranked by rule importance measures, such as R131 in Glass data set in Table 5.12. Such *Reduct Rules* are not ranked by the Rule Importance Measure because they either do not contain important attributes, or because the attribute values are not frequently occurring.

5.6 Conclusions

We introduced a Rules-As-Attributes measure to discover important rules by considering rules as attributes. Association rules are used for rule generation. A new decision table is constructed by considering all the rules as condition attributes. Reducts generated by ROSETTA from this new decision table are representative of rules from the original data set. The experimental results for discovering important rules are promising and exciting. The process of extracting *Reduct Rules* is automatic. *Reduct Rules* are a subset of the original rules which are representative and important. This method can be used together with the Rule Importance Measure to take a further step to evaluate rules. We discuss the relationship between these two measures in Section 7.2.2.

We are interested in applying the Rules-As-Attributes measure to recommender systems for interesting recommendations. In particular, we are interested in collaborative filtering systems which observe the behaviours and the patterns of the current users, and make recommendations based on the similarities between the current users and other users. A decision table can be constructed by considering user's interests as condition attributes, and different recommended items as decision attributes. Therefore, association rules generated from this decision table, with decision attributes on the consequent part, can be used to make recommendations. The *Reduct Rules* extracted from the proposed approach can thus be used to provide representative and interesting recommendations.

Chapter 6

Frequent Itemset and Missing Attribute Values

How to process missing attribute values is an important data preprocessing problem in data mining and knowledge discovery tasks. A commonly-used and naïve solution to process data with missing attribute values is to ignore the instances which contain missing attribute values. This method may neglect important information within the data and a significant amount of data could be easily discarded. Some methods, such as assigning the most common values or assigning an average value to the missing attribute, make good use of all the available data. However the assigned value may not come from the information which the data originally derived from; thus, noise is brought to the data.

In this chapter, we introduce two approaches RSFit and ItemRSFit to effectively predict missing attribute values. The frequent itemset is generated from the association rules algorithm and it displays the correlations between different items in a transaction data set. Considering a data set as a transaction, each data instance as an itemset, frequent itemset can be used as a knowledge base to predict missing attribute values. However this approach alone cannot predict all the existing missing attributes. RSFit [55] is a newly developed approach to predict missing attribute values based on the similarities of attribute-value pairs by only considering attributes contained in the core or the reduct of the data set. The RSFit approach provides a faster prediction and can be used for predicting the cases that cannot be covered by the itemset approach. We name the integrated approach ItemRSFit.

Empirical studies on UCI data sets and a real world data set demonstrate a significant increase of predicting accuracy obtained from this new integrated approach.

6.1 Introduction

We propose two approaches based on rough sets theory and association rule algorithms for processing data with missing attribute values. We first discuss the current approaches for processing missing attribute data. We then introduce an approach **RSFit** for processing data with missing attribute values based on rough sets theory. By matching attribute-value pairs among the attributes from the same core or reduct of the original data set, the assigned value preserves the characteristics of the original data set. We compare our approach with "closest fit approach globally" and "closest fit approach in the same concept", which are the two recent rough sets approaches for processing missing attribute values [27]. We conduct experiments on complete data sets with a randomly selected number of missing attribute values. Then we compare the accuracy of the predictions using the proposed RSFit approach and other existing approaches. Experimental results on UCI data sets and a real geriatric care data set show that the RSFit approach can obtain a comparable prediction accuracy on assigning the missing values while at the same time significantly reducing the computation time. However, the RSFit approach, like most other existing approaches, cannot provide a high percentage of prediction to all the missing attribute values.

In the second part of this chapter, we introduce an integrated approach **ItemRSFit** to effectively predict missing attribute values by combining the frequent itemset approach and RSFit together. Frequent itemsets are generated from the association rule algorithm for transaction data. The itemsets demonstrate correlations between different items from the transaction. Therefore, the frequent itemsets can be considered as a knowledge base for correlations between items. If one item in a transaction is missing, it can be predicted by the correlations from the frequent itemsets based on other transactions containing this item. Since a general data set can be considered as transaction data, missing attribute values can be considered as missing item values. We can use frequent itemsets to predict those missing attribute values. The experimental results show that using frequent itemset

as a knowledge base to predict missing attribute values can provide a high prediction accuracy. However this approach alone cannot guarantee a complete prediction to all the existing missing attributes in the data set, because not all the attributes are associated with other attributes. Although with a lower *support* value, the association rule algorithm can extract rare associations between different possible values, it is computationally time consuming to use a larger knowledge base on prediction. We would like to discover a tradeoff between an acceptable prediction accuracy for missing attribute values and an acceptable computation time. Adopting the fast prediction advantage of RSFit approach, we can use this approach to predict those data instances that cannot be predicted by the itemset approach. Empirical studies on artificial data sets and a real world data set demonstrate a significant increase of predicting accuracy obtained from this new integrated approach.

The rest of the chapter is organized as follows. Section 6.2 presents existing approaches on processing missing attribute values. Our proposed rough sets based approach RSFit is explained in Section 6.3, and initial experimental results for the RSFit approach are demonstrated in this section. Section 6.4 introduces the ItemRSFit approach. Section 6.5 gives concluding remarks and discuss future work.

6.2 Related Work

Various approaches on how to cope with missing attribute values have been proposed in the past years. We list some representative approaches as follows.

6.2.1 From Rough Sets Theory

In [28] nine approaches on filling in the missing attribute values were introduced, such as selecting the "most common attribute value", the "concept most common attribute value", "assigning all possible values of the attribute restricted to the given concept", "ignoring examples with unknown attribute values", "treating missing attribute values as special values", etc. We will enumerate them in the following.

- The approach of most common attribute value. This approach will assign to the

missing attribute value the most common value among all the possible values of the attribute.

- The approach of concept most common attribute value. This approach will assign the most common value among all the possible values of the attribute, by only considering data instances with the same concept (decision attribute) as the concept of the missing instance.

- The approach of assigning all possible values of the attribute restricted to the given concept. This approach will consider only data instances with the concept value the same as the given concept, and assign all the possible values from attributes contained in these data instances for the missing attribute.

- The approach of ignoring examples with unknown attribute values. This approach simply discards all the data instances containing any unknown or missing attribute values.

- The approach of treating missing attribute values as special values. The missing value is considered as one of the possible values of the attribute.

In [27] a "closest fit" approach was proposed to compare the vectors of all the attribute pairs from a preterm birth data set, and assign the value from the most similar pair to the missing value. A distance function was used to calculate the similarities between the attribute pairs. In more recent research [26] four interpretations on the meanings of missing attribute values such as "lost" values and "do not care" values are discussed. Different approaches from rough sets theory are demonstrated on selecting values for the individual interpreted meanings.

Although these approaches provide a simple and direct processing to the missing attribute values, noise is usually brought into the data set as well.

Consider the approach of "assigning most common attribute values" [28] as an example. This approach assigns the most frequently appeared value among the attribute to the missing value. Shown in Table 6.1 as an example, there are 4 data instances existing in a data set $T(C, D)$, where C is the condition attribute set, D is the decision attribute set, U is the set of data instances, $C = (c_1, c_2, c_3, c_4), D = (0, 1), U = \{u_1, \ldots, u_4\}$. There is

a missing value for c_3 in u_2, represented with "?". According to this approach, the most common value for attribute c_3 is 2. However, if we assign the value, the data set becomes inconsistent. u_1 and u_2 will have the same condition attributes with different decision attributes.

Table 6.1: Sample Data Set with Missing Attribute Values

	Condition				Decision
U	c_1	c_2	c_3	c_4	D
1	1	2	2	1	1
2	1	2	?	1	0
3	1	1	3	1	0
4	1	0	2	0	1

Another approach of "treating missing attribute values as special values" [28] may also bring noise to the original data. The missing value is considered as an individual "unknown" value for the attribute. However, the attribute may not at all be possible to have another value in certain scenario. For example, suppose in a data set, the missing attribute is "gender of a patient" with values of either "male" or "female". In case of missing value for this attribute, we cannot assign a "unknown" to this attribute.

More research efforts are concentrating on how to predict the missing attribute values by obtaining the most information out of the original data set. In [47], support and confidence for the association rules generated from data containing missing attribute values were considered not precise. Rough sets theory was used to estimate the support and confidence values for the generated association rules. For each large itemset, based on which the association rules would be further generated, the maximal sets of tuples that are matched, or may match, or certainly did not match, or may not match the item set were listed. The lowest and the highest possible support and confidence values were further defined and computed based on these sets. Different approaches from rough sets theory are demonstrated on selecting values for the individual interpreted meanings.

6.2.2 From Data Mining

In addition to the efforts from rough sets theory on processing missing attribute values, strategies from data mining area are also widely applied in predicting the missing values. In [31] it is suggested that using regression or inference-based tools on the data set can produce a more precise prediction for the missing attributes. A robust algorithm of generating optimal association rules to solve the missing attribute value problems in the testing data set has been discussed in [50]. In [91], the authors discussed a new approach on using association rules generation on completing missing values. Data associations are created based on an association rule algorithm and are then used to find the associated values for the missing data. Formulas, based on support, confidence and lift, were applied to help choosing the better options when multiple matches existed. Recently Zhu and Wu [100] introduced methods on processing missing attribute values by considering the attribute cost. They point out that the common problems on assigning missing values are that not all the missing values can be predicted by current data mining approaches, and the predictions do not usually bring higher prediction accuracy. They consider in the real world, it is expensive to predict all the missing attributes, therefore a technique is needed on balancing the prediction percentage, the prediction accuracy and the computational cost. They evaluate the importance of different missing data instances by information-gain ratio.

6.2.3 Motivations

Inspired by, though different from, the related work, we are interested in predicting missing attribute values in the data preprocessing stage. We consider rough sets theory as an effective approach on attribute selection; therefore, a subset of the whole condition attributes can be used to make effective prediction instead of considering the complete data as the knowledge base. We are motivated to develop a technique that can predict all the missing attribute values with a high precision.

We discuss how to effectively predict missing attribute values from both the data mining technique and the rough sets theory. We show how to avoid bias and use more information from the data itself to predict the missing values.

We are interested in integrating two techniques into our research. One of them is the association rule algorithm [3], which is well known in data mining for discovering item relationships from large transaction data sets. Prior to the association rule generation, frequent itemsets are generated based on the item-item relations from the large data set according to a certain *support*. Thus the frequent itemsets of a data set represent strong correlations between different items, and the itemsets represent probabilities for one or more items existing together in the current transaction. When considering a certain data set as a transaction data set, the implications from frequent itemsets can be used to find which attribute value the missing attribute is strongly connected to and the frequent itemset can be used for predicting the missing values. We call this approach "itemset-approach" for prediction. Apparently, the larger the frequent itemsets used for the prediction, the more information from the data set itself is used for prediction; hence, the higher the accuracy will be obtained. However, generating frequent itemsets for a large data set is time-consuming. Itemsets with lower support, which leads to larger size itemsets, usually costs a significant amount of computation time. Although itemsets with higher support need less computation time, they show restricted item relationships and the applicable number of itemsets are fewer; therefore, not all the missing values can be predicted. In order to balance the tradeoff between computation time and the percentage of the applicable prediction, another approach has to be taken into consideration.

Rough sets theory has been used for attribute selection, rule discovery and many knowledge discovery applications in the areas such as data mining, machine learning and medical diagnoses. Core and reduct are among the most important concepts in this theory. A reduct contains a subset of condition attributes that are sufficient enough to represent the whole data set. The intersection of all the possible reduct is the core. Therefore the attributes contained in the reduct or core are more important and representative than the rest of the attributes. Therefore by examining only attributes within the same core or reduct to find the similar attribute value pairs for the data instance containing the missing attribute values, we can assign the most relevant value for the missing attribute. Since this method only considers a subset of the data set, which is either the core or the reduct, the prediction is quite fast. This approach "RSFit" is an alternative approach for fast prediction and it can be used to predict missing attributes that cannot be predicted by the frequent itemset.

We integrate the prediction based on frequent itemset and the RSFit approach into a new approach **ItemRSFit** to predict missing attribute values. This approach can predict missing values from the data itself; therefore, less noise is brought into the original data.

6.3 RSFit Approach to Assign Missing Values

In this section, we introduce the RSFit approach for predicting missing values. We first make definitions to be used in the following descriptions of the proposed approaches. The input to our approach is a decision table $T = (C, D)$, where $C = \{c_1, c_2, \ldots, c_m\}$ is the condition attribute set, and $D = \{d_1, d_2, \ldots, d_l\}$ is the decision attribute set. $U = \{u_1, u_2, \ldots, u_n\}$ represent the set of data instances in T. For each u_i $(1 \leq i \leq n)$, an **attribute-value pair** for this data instance is defined to be $u_i = (v_{1i}, v_{2i}, \ldots, v_{mi}, d_i)$, where v_{1i} is the attribute value for condition attribute c_1, v_{2i} is the attribute value for condition attribute c_2, ..., v_{mi} is the attribute value for condition attribute c_m.

6.3.1 Detailed Explanation

The core or the reduct of a data set contains a set of attributes that are able to represent the original data set. The attributes contained in the same core or the reduct set are related to each other to a certain degree. We consider attribute-value pairs contained in the same core or reduct set to find the best match for the missing values. This approach is inspired by the "closest fit" approach by Grzymala-Busse [27]; however, it is different from it. Instead of searching the whole data set for closest matched attribute-value pairs, RSFit searches only for the attribute-value pairs within the core or a reduct.

For each missing attribute value, we let the attribute be the "target attribute" (represented as c_k in the following). We assume that missing attribute values only exist in the condition attributes not in the decision attributes. We explain the RSFit approach for how to find the matched value for this target attribute, in detail.

First, we obtain the core of the data set $T = (C, D)$ based on Hu's core algorithm introduced in [40] (explained in Chapter 2). If the target attribute c_k does not belong to the core, we include c_k into the core. In case there is no core for T, we consider a reduct of T. ROSETTA software [69] is used for reduct generation. There are a few

reduct generation algorithms provided by ROSETTA. We use Johnson's algorithm for single reduct generation. In the case of no reducts containing the target attribute c_k, we include the target c_k into the reduct.

Secondly, a new decision table $T' = (C', D)$ is created based on the previous step, where $C' = \{c'_1, c'_2, \ldots, c_k, \ldots, c'_{m'}\}$, $1 \leq m' \leq m$, $1 \leq k \leq m'$, and $C' \subseteq C$, C' is either the core or the reduct of C, $U' = \{u_1, u_2, \ldots, u_{n'}\}$, $1 \leq n' \leq n$. There are two possibilities for selecting the data instances. One possibility is to include other data instances with missing values to predict the current target attribute value; the other option is to exclude all the other data instances containing missing attribute values. We allow the other missing attribute values to exist by designing the proper match function.

Thirdly, in T', when considering the match cases, there are two possibilities. One possibility is that we consider all the data instances; the other is to consider data instances having the same decision attribute values while finding a matched attribute-value pair. Here we call the first possibility *global*, and the second *concept*. We perform experiments to test both possibilities. We would like to examine the prediction difference (if any) between the two possibilities and to determine whether they bring inconsistencies into the data.

Fourthly, we define a distance function to compute the similarities between different attribute-value pairs. The details of the distance function is elaborated in the following. Let $u_i = (v_{1i}, v_{2i}, \ldots, v_{ki}, \ldots, v_{m'i}, d_i)$ $(1 \leq i \leq n')$ be the attribute-value pair containing the missing attribute value v_{ki} (represented as $v_{ki} =?$) for c_k $(1 \leq k \leq m')$. Distance functions, such as Euclidean distance and Manhattan distance, are used in instance-based learning to compare the similarity between a test instance and the training instances [90]. We use Manhattan distance [1]to evaluate the distance between an attribute-value pair containing missing attribute values with other attribute-value pairs. This formula is also used in the "closest fit" approach [27]. Let u_j be a data instance from U. The distance between u_j to

[1]In our experiments, the prediction results by Manhattan distance and Euclidean distance returned the same accuracy. Because the computation for Manhattan distance is faster, we use Manhattan distance as the distance function.

the target data instance u_i is defined as[2]

$$distance(u_i, u_j) = \frac{|v_{i1} - v_{j_1}|}{\max v_1 - \min v_1} + \frac{|v_{i2} - v_{j_2}|}{\max v_2 - \min v_2} + \ldots + \frac{|v_{im} - v_{j_m}|}{\max v_m - \min v_m}.$$

For attributes which are the missing attribute values, the distance is set to be 1, which specifies the maximum difference between unknown values. The best match has the smallest difference from the target attribute-value pair. After the best matched attribute-value pair is returned by the algorithm, the corresponding value will be assigned to the target attribute. We consider all the attributes as numerical attributes. In case of symbolic attributes, we convert them to numerical ones during the preprocessing stage. In case there are multiple matched attribute-value pairs for the missing attribute, one of the values is randomly selected to be assigned to the missing value.

6.3.2 A Walk Through Example

We demonstrate the RSFit approach by an artificial car data set which appeared in [40] as shown in Table 6.2. One missing attribute value is randomly selected across the data set as shown by Table 6.3.

First, the core is obtained for this data set as "Make_model" and "trans". Since the core attributes exist and the missing attribute "compress" does not belong to the core, we add attribute "compress" to the core set. The new data set containing the core attributes, target attribute "compress" and the decision attribute are created and shown in Table 6.4. Then we will find the match for attribute "compress" in u_8. For "RSFit-global", we find the u_{14} has the smallest difference, which is 0, from u_8, therefore, u_{14} is the best match. We assign $c_{compress_{14}}$ to $c_{compress_8}$, which is "High" (correct prediction). For "RSFit-concept", we only look for attribute-value pairs that have the same decision attribute value as u_8, which is $mileage = high$. We find that u_{14} is the best match. We assign $c_{compress_{14}}$ to $c_{compress_8}$, which is "High" (correct prediction). For "closest fit-global" approach, we examine all the instances in the data set. u_5 is the closest fit, $c_{compress_5}$ = "Medium" (wrong prediction). For "closest fit-concept" approach, we examine only the data with decision attribute "High". We find u_{10} with $c_{compress_{10}}$ = "Medium" as the match (wrong prediction).

[2]In the algorithm, $|x|$ returns the absolute value of x.

Table 6.2: Artificial Car Data Set

U	Make_model	cyl	door	displace	compress	power	trans	weight	mileage
1	usa	6	2	medium	high	high	auto	medium	medium
2	usa	6	4	medium	medium	medium	manual	medium	medium
3	usa	4	2	small	high	medium	auto	medium	medium
4	usa	4	2	medium	medium	medium	manual	medium	medium
5	usa	4	2	medium	medium	high	manual	medium	medium
6	usa	6	4	medium	medium	high	auto	medium	medium
7	usa	4	2	medium	medium	high	auto	medium	medium
8	usa	4	2	medium	high	high	manual	light	high
9	japan	4	2	small	high	low	manual	light	high
10	japan	4	2	medium	medium	medium	manual	medium	high
11	japan	4	2	small	high	high	manual	medium	high
12	japan	4	2	small	medium	low	manual	medium	high
13	japan	4	2	small	high	medium	manual	medium	high
14	usa	4	2	small	high	medium	manual	medium	high

Table 6.3: Artificial Car Data Set with One Missing Attribute Value

U	Make_model	cyl	door	displace	compress	power	trans	weight	mileage
1	usa	6	2	medium	high	high	auto	medium	medium
2	usa	6	4	medium	medium	medium	manual	medium	medium
3	usa	4	2	small	high	medium	auto	medium	medium
4	usa	4	2	medium	medium	medium	manual	medium	medium
5	usa	4	2	medium	medium	high	manual	medium	medium
6	usa	6	4	medium	medium	high	auto	medium	medium
7	usa	4	2	medium	medium	high	auto	medium	medium
8	usa	4	2	medium	?	high	manual	light	high
9	japan	4	2	small	high	low	manual	light	high
10	japan	4	2	medium	medium	medium	manual	medium	high
11	japan	4	2	small	high	high	manual	medium	high
12	japan	4	2	small	medium	low	manual	medium	high
13	japan	4	2	small	high	medium	manual	medium	high
14	usa	4	2	small	high	medium	manual	medium	high

6.3.3 Evaluation Method

Our goal is to test the accuracy of using the RSFit method to predict the missing values, and compare the accuracy and the computation time with "closest fit-global" and "closest fit-concept" approaches. We use the following way to perform the evaluation process. We consider complete data sets as the input data. For each data set, we randomly select

Table 6.4: New Decision Table for Car Data Set Based on Core Set with One Missing Attribute Value

U	Make_model	compress	trans	mileage
1	usa	high	auto	medium
2	usa	medium	manual	medium
3	usa	high	auto	medium
4	usa	medium	manual	medium
5	usa	medium	manual	medium
6	usa	medium	auto	medium
7	usa	medium	auto	medium
8	usa	?	manual	high
9	japan	high	manual	high
10	japan	medium	manual	high
11	japan	high	manual	high
12	japan	medium	manual	high
13	japan	high	manual	high
14	usa	high	manual	high

a certain number of the attribute-value pairs among the whole data set and remove the values to produce x missing attribute values per data set. We test different approaches on assigning the missing values, and compare the accuracy of the prediction. In order to average the odds of the randomly selected missing attributes, we perform this process 100 times for each data set for each x missing attribute values and average the accuracy.

6.3.4 Experimental Results for the RSFit approach

In order to test our proposed approach, we experiment on selected UCI data sets [21] and a geriatric care data set [53], which contain no missing attribute values.

These data sets can be further divided into two categories. One category of data sets contain core attributes, such as, geriatric care data, spambase data and zoo data. The other set of data sets do not contain core attributes, such as lymphography data. We do not discuss the type of data set in which the core attributes are all the condition attributes in this thesis (in this case, the method of RSFit is the same as the closest fit approach).

Geriatric Care Data Set We perform experiments on a geriatric care data set from Dalhousie University Medical School. This data set contains 8547 patient records with 44 symptoms and their survival status. The data set is used to determine the survival status of a patient given all the symptoms he or she shows. We use *survival status* as the decision attribute, and the 44 symptoms of a patient as condition attributes, which includes *education level, the eyesight, the age of the patient at investigation* and so on [3]. There is no missing value in this data set. There are 12 inconsistent data entries in the medical data set. After removing these instances, the data contains 8535 records [4]. Table 6.5 gives selected data records of this data set. There are 14 core attributes generated

Table 6.5: Geriatric Care Data Set

edulevel	eyesight	hearing	health	trouble	livealone	cough	hbp	heart	stroke	...	sex	livedead
0.6364	0.25	0.50	0.25	0.00	0.00	0.00	0.00	0.00	0.00	...	1	0
0.7273	0.50	0.25	0.25	0.50	0.00	0.00	0.00	0.00	0.00	...	2	0
0.9091	0.25	0.50	0.00	0.00	0.00	0.00	1.00	1.00	0.00	...	1	0
0.5455	0.25	0.25	0.50	0.00	1.00	1.00	0.00	0.00	0.00	...	2	0
0.4545	0.25	0.25	0.25	0.00	1.00	0.00	1.00	0.00	0.00	...	2	0
0.2727	0.00	0.00	0.25	0.50	1.00	0.00	1.00	0.00	0.00	...	2	0
0.0000	0.25	0.25	0.25	0.00	0.00	0.00	0.00	1.00	0.00	...	1	0
0.8182	0.00	0.50	0.00	0.00	0.00	0.00	1.00	0.00	0.00	...	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...

for this data set. They are *eartroub, livealone, heart, hbp, eyetroub, hearing, sex, health, edulevel, chest, housewk, diabetes, dental, studyage.*

Lymphography Data The data set contains 148 instances and 18 condition attributes. There are no missing attribute values in this data. We check that there is no inconsistent data. The core is empty for this data set. Johnson's reduct generated from this data set contains *blockofaffere, changesinnode, changesinstru, specialforms, dislocationof, noofnodesin.*

[3]Refer to [53] for details about this data set.

[4]Notice from our previous experiments that core generation algorithm can not return correct core attributes when the data set contains inconsistent data entries.

Spambase Data This data set originally contains 4,601 instances and 57 condition attributes. It is used to classify spam and non-spam emails. Most of the attributes indicate whether a certain word (such as, order, report) or character (such as !, #) appears frequently in the emails. There are no missing attribute values. There are 6 inconsistent data instances that are removed. The core attributes, which are essential to determine whether an email is not a spam email, are, the word frequency of "george", "meeting", 're", "you", "edu", "!", and the total number of capital letters in the email. In addition, it is interesting to pay attention to the reducts as well. They are important information on identifying the possible spam emails.

Zoo Data This artificial data set contains 7 classes of animals, 17 condition attributes, 101 data instances, and there are no missing attribute values in this data set. Since the first condition attribute "animal name" is unique for each instance, and we consider each instance a unique itemset, we do not consider this attribute in our experiment. There are no inconsistent data in this data set. The core attributes are *aquatic, legs*.

6.3.5 Comparison Results

The compared approaches are implemented by Perl and the experiments are conducted on Sun Fire V880, four 900Mhz UltraSPARC III processors. Our proposed rough sets based approach considers a subset of the attributes (the reduct or the core). In order to compare whether the reduct or the core provide a better choice of attributes, we also compare our approach against a randomly selected subset of the attributes as reduct or core. Given a reduct of size n, we randomly choose a combination of n attributes. The comparison results on processing missing attribute values between the RSFit approach, closest fit approach and random approach on geriatric care data set, spambase data set, lymphography data set and zoo data set are shown in Figure 6.1, Figure 6.2, Figure 6.3, and Figure 6.4. The reduct and core generation time are not included in the comparison results.

The comparison results are shown in the following Table 6.6, Table 6.7, Table 6.8 and Table 6.9.

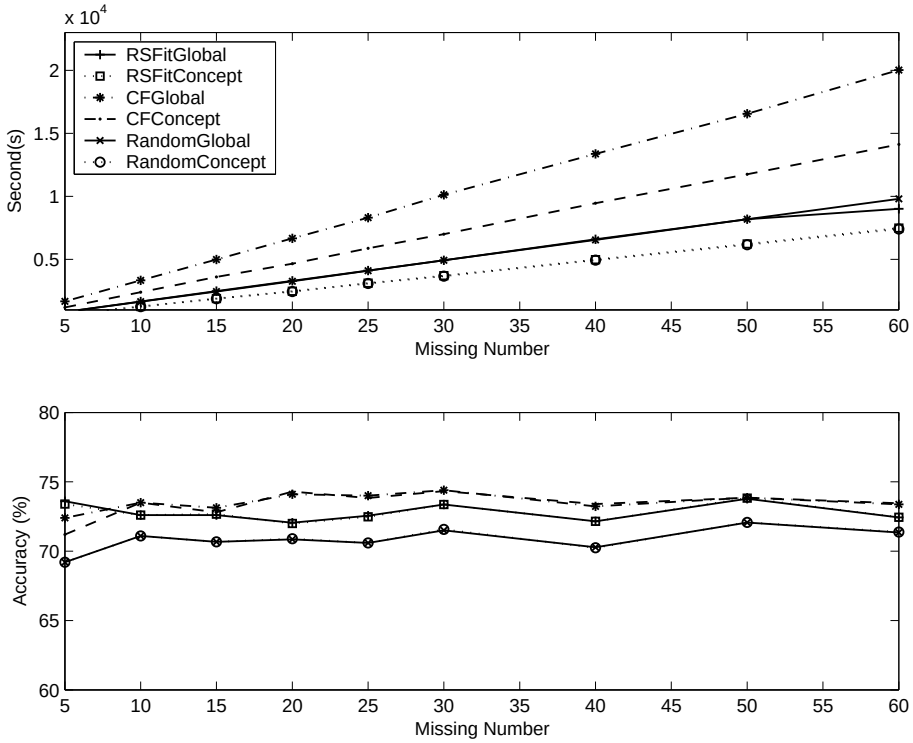

Figure 6.1: Comparison Figure for Geriatric Care Data

6.3.6 Discussions

In the comparison figures (Figure 6.1, 6.2, 6.3, and 6.4), "RSFitGlobal" and "RSFitConcept" stand for the new approach proposed in this chapter. "CFGlobal" and "CFConcept" stand for the "closest fit" approach from [27]. "RandomGlobal" and "RandomConcept" stand for the random selected attributes approach. For each figure, the upper chart shows the prediction time; the lower chart shows the prediction accuracy. Our proposed rough sets theory based method RSFit achieved significant saving on computation time for assigning missing attribute values. It can be used in the situation when time is the most important issue, with the sacrifice of less precision. The time saving is quite noticeable for larger data sets such as geriatric care and spambase data set. Taking the geriatric care data as an example, among the 44 condition attributes, we only consider 14 of them which are core attributes. Comparing "RSFitGlobal" to "CFGlobal", the prediction precision

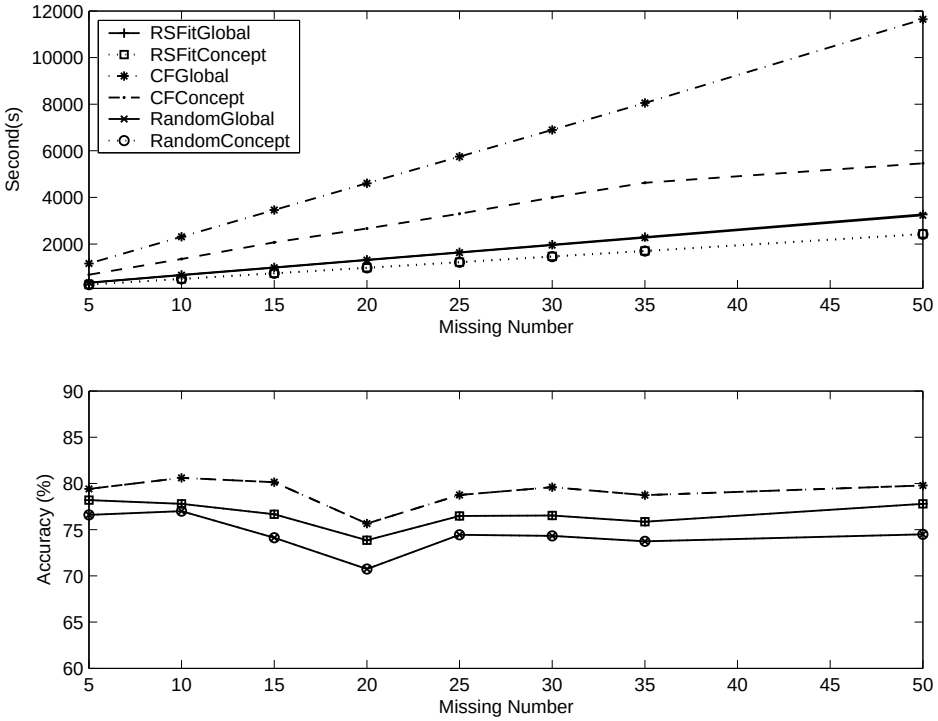

Figure 6.2: Comparison Figure for Spambase Data

of RSFit is on average 0.762% lower than the "closest fit" approach; however, the computation time of ours is on average 49.026% of the computation time for the "closest fit" approach. "RSFitConcept" and "RSFitGlobal" achieve similar prediction accuracy; however, "RSFitConcept" takes slightly less computation time because the amount of data the approach processes is less. The fact that concept related prediction is faster than global prediction also applies to the "closest fit" approach and the random approach. The experimental results also shows that the RSFit approach provides a higher prediction accuracy than the random approach. The reduct from the rough sets theory presents a better choice of attributes than the randomly selected attributes on representing the original data.

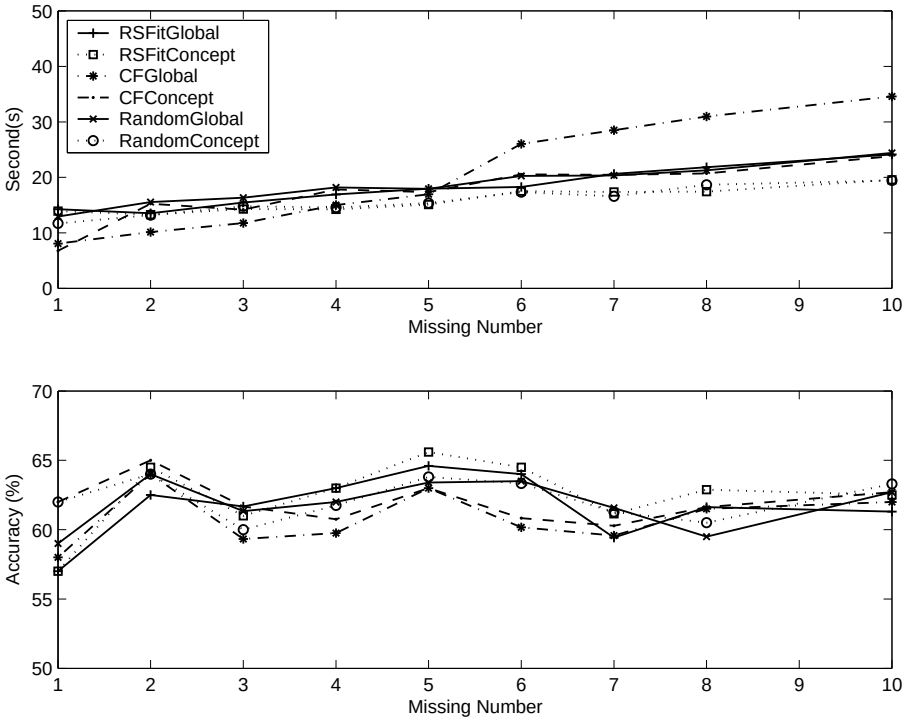

Figure 6.3: Comparison Figure for Lymphography Data

6.4 The ItemRSFit Approach

The RSFit approach cannot provide a very high prediction precision, although it is computationally faster than the "closest fit" approach [55]. This is because this approach does not fully consider the item-item relationships inside the data set. The RSFit uses the subset of a transaction as a knowledge base to find the similar object for prediction. It is actually comparing the similarity between the subsets of transactions and assigns the values from the most similar transaction to the missing item. This kind of similarity does not consider the item-item relationship. The frequency of a certain item existing in the transaction in fact indicates how frequently the other item(s) exist(s) in the transaction. The indictions from the strong associations between different items can be discovered by the association rule algorithm.

In this section, we discuss how to use the association rule algorithm to help process

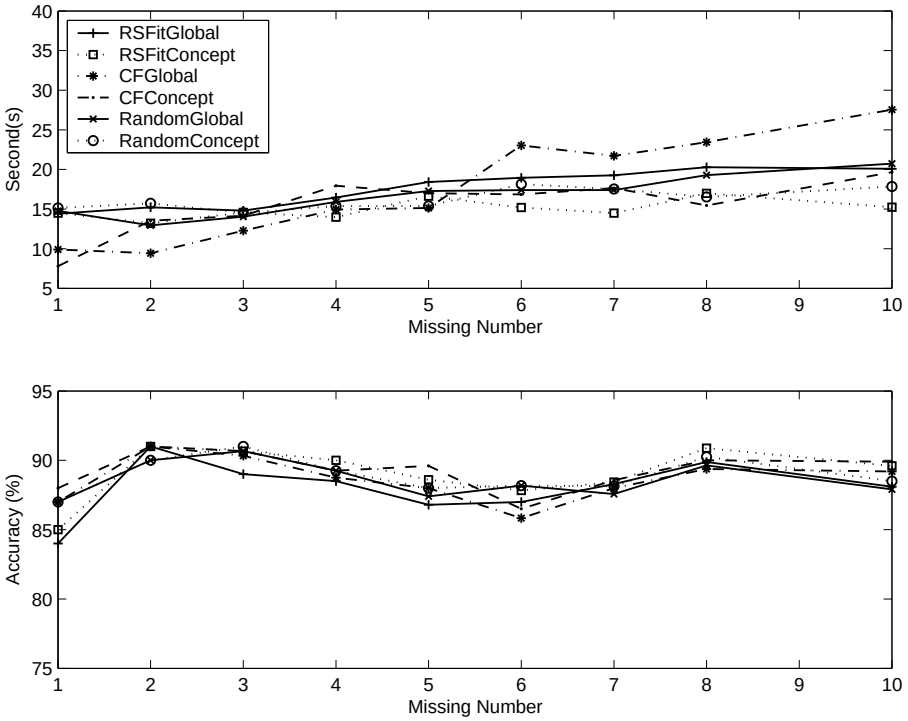

Figure 6.4: Comparison Figure for Zoo Data

missing attribute values. We then introduce the ItemRSFit approach which integrates both RSFit and frequent itemsets.

The association rule algorithm was first introduced in [3], and it can be used to discover rules from transaction datasets. Association rule algorithms can be used to find associations among items from transactions. For example, in *market basket analysis*, by analyzing transaction records from the market, we could use association rule algorithms to discover different shopping behaviours such as, when customers buy bread, they will probably buy milk. This type of behaviours can be used in the market analysis to increase the amount of milk sold in the market.

Frequent itemset generation is the first step of the two for association rule generation. Itemsets that frequently occur together in the transactions are generated. Rules based on these itemsets are further extracted to represent the associated relations.

Here we consider data in the form of a decision table as the transaction data for gen-

Table 6.6: Comparisons on Accuracies and Time For Geriatric Care Data

Data Sets	Computation Time (Second) For 100 Run					
Missing Values	RSFit Global	RSFit Concept	ClosestFit Global	ClosestFit Concept	Random Global	Random Concept
5	842.637	642.607	1673.972	1193.496	835.15	638.928
10	1660.891	1266.308	3337.450	2403.185	1645.864	1255.622
15	2481.925	1896.875	4993.163	3611.637	2454.688	1880.036
20	3298.103	2479.502	6668.741	4663.448	3265.128	2452.722
25	4118.382	3116.954	8315.181	5878.851	4106.38	3088.842
30	4933.933	3714.456	10126.725	7000.339	4928.184	3676.568
40	6595.240	4978.143	13375.369	9462.916	6552.399	4936.833
50	8183.797	6222.527	16557.562	11747.613	8188.923	6162.332
60	9908.241	7479.915	20024.138	14126.664	9807.790	7413.180
Data Sets	Average Accuracy (Percentage %) over 100 Times					
Missing Values	RSFit Global	RSFit Concept	ClosestFit Global	ClosestFit Concept	Random Global	Random Concept
5	73.6%	73.4%	72.4%	71.2%	69.2%	69.2%
10	72.6%	72.6%	73.5%	73.5%	71.1%	71.1%
15	72.6%	72.67%	73.13%	72.8%	70.67%	70.67%
20	72.05%	72.00%	74.10%	74.30%	70.85%	70.90%
25	72.56%	72.48%	74.00%	73.84%	70.60%	70.60%
30	73.37%	73.37%	74.40%	74.33%	71.50%	71.57%
40	72.15%	72.15%	73.23%	73.40%	70.28%	70.25%
50	73.78%	73.82%	73.86%	73.86%	72.06%	72.08%
60	72.43%	72.45%	73.38%	73.45%	71.35%	71.38%

Data Instances: 8535. Condition Attributes: 44.

erating the frequent itemsets. For a rough set approach we define the following concepts.

Table 6.7: Comparisons on Accuracies and Time For Spambase Data

Data Sets	Computation Time (Second) For 100 Run					
Missing Values	RSFit Global	RSFit Concept	ClosestFit Global	ClosestFit Concept	Random Global	Random Concept
5	343.581	262.547	1163.445	685.700	339.987	260.068
10	676.317	504.733	2310.732	1361.200	663.801	500.903
15	995.771	746.852	3458.372	2069.000	987.348	742.708
20	1323.940	986.152	4605.719	2667.500	1309.055	977.558
25	1647.742	1223.637	5752.945	3300.900	1629.186	1216.533
30	1970.233	1470.366	6896.710	3992.000	1949.636	1460.533
35	2299.640	1705.113	8051.766	4625.400	2270.247	1695.289
50	3276.769	2437.121	11642.691	5461.400	3236.639	2420.724
Data Sets	Average Accuracy (Percentage %) over 100 Times					
Missing Values	RSFit Global	RSFit Concept	ClosestFit Global	ClosestFit Concept	Random Global	Random Concept
5	78.20%	78.20%	79.40%	79.40%	76.60%	76.60%
10	77.80%	77.80%	80.60%	80.60%	77.00%	77.00%
15	76.67%	76.67%	80.13%	80.13%	74.13%	74.13%
20	73.85%	73.85%	75.65%	75.65%	70.75%	70.75%
25	76.48%	76.48%	78.76%	78.76%	74.44%	74.44%
30	76.53%	76.53%	79.60%	79.60%	74.33%	74.33%
35	75.86%	75.86%	78.74%	78.74%	73.74%	73.74%
50	77.80%	77.80%	79.78%	79.78%	74.50%	74.50%

Data Instances: 4601. Condition Attributes: 57.

Table 6.8: Comparisons on Accuracies and Time For Lymphography Data

Data Sets	Computation Time (Second) For 100 Run					
Missing Values	RSFit Global	RSFit Concept	ClosestFit Global	ClosestFit Concept	Random Global	Random Concept
1	14.269	13.913	8.063	6.760	12.93	11.717
2	13.532	13.275	10.124	15.255	15.561	13.208
3	15.454	14.292	11.765	14.185	16.332	14.781
4	16.92	14.237	15.006	17.814	18.189	14.566
5	17.965	15.09	16.964	17.351	17.926	15.36
6	18.273	17.511	26.036	20.546	20.259	17.352
7	20.626	17.38	28.503	20.468	20.331	16.582
8	21.842	17.418	30.979	20.712	21.264	18.651
10	24.121	19.558	34.579	23.815	24.405	19.444
Data Sets	Average Accuracy (Percentage %) over 100 Times					
Missing Values	RSFit Global	RSFit Concept	ClosestFit Global	ClosestFit Concept	Random Global	Random Concept
1	57.00%	57.00%	58.00%	62.00%	59.00%	62.00%
2	62.50%	64.50%	64.00%	65.00%	64.00%	64.00%
3	61.67%	61.00%	59.33%	61.67%	61.33%	60.00%
4	63.00%	63.00%	59.75%	60.75%	62.00%	61.75%
5	64.60%	65.60%	63.00%	63.00%	63.40%	63.80%
6	64.00%	64.50%	60.17%	60.83%	63.50%	63.33%
7	59.43%	61.14%	59.57%	60.28%	61.57%	61.28%
8	61.63%	62.88%	61.50%	61.63%	59.50%	60.50%
10	61.30%	62.50%	62.00%	62.70%	62.70%	63.30%

Data Instances: 148. Condition Attributes: 18.

Table 6.9: Comparisons on Accuracies and Time For Zoo Data

Data Sets	Computation Time (Second) For 100 Run					
Missing Values	RSFit Global	RSFit Concept	ClosestFit Global	ClosestFit Concept	Random Global	Random Concept
1	14.404	14.495	9.900	7.790	14.767	15.135
2	15.219	13.219	9.421	13.524	12.938	15.737
3	14.813	14.579	12.284	14.154	14.043	14.522
4	16.445	13.974	14.937	17.948	15.909	15.376
5	18.416	16.639	15.136	17.012	17.277	15.385
6	18.938	15.195	23.021	16.837	17.387	18.133
7	19.259	14.500	21.720	17.647	17.401	17.543
8	20.278	16.990	23.439	15.456	19.290	16.528
10	20.089	15.236	27.541	19.657	20.738	17.846
Data Sets	Average Accuracy (Percentage %) over 100 Times					
Missing Values	RSFit Global	RSFit Concept	ClosestFit Global	ClosestFit Concept	Random Global	Random Concept
1	84.00%	85.00%	87.00%	88.00%	87.00%	87.00%
2	91.00%	91.00%	91.00%	91.00%	90.00%	90.00%
3	89.00%	90.67%	90.33%	90.67%	90.67%	91.00%
4	88.50%	90.00%	88.75%	89.25%	89.25%	89.25%
5	86.80%	88.60%	87.99%	89.60%	87.40%	87.99%
6	87.00%	87.83%	85.83%	86.50%	88.17%	88.17%
7	88.29%	88.43%	88.00%	88.57%	87.57%	88.14%
8	89.88%	90.88%	89.38%	90.00%	89.63%	90.25%
10	88.10%	89.60%	89.20%	89.90%	87.90%	88.50%

Data Instances: 101. Condition Attributes: 16.

Definition 5 Transaction. The set of transactions to the frequent itemsets generation is in a form of a decision table T=(C, D), where $C = \{c_1, c_2, \ldots, c_m\}$ is the condition attribute set where m is the number of condition attributes, and $D = \{d_1, d_2, \ldots, d_l\}$ is the decision attribute set where l is the number of decision attributes. $U = \{u_1, u_2, \ldots, u_n\}$ represent the itemsets in T, where n is the number of transactions in T. Each transaction contains $(m + l)$ items.

Therefore each attribute value is considered an item in the transaction.

An association rule [3] is a rule of the form $\alpha \rightarrow \beta$, where α and β represent itemsets which do not share common items.

Definition 6 Support. A support of an itemset is the percentage of the number of transactions containing the itemset to the total number of transactions.

Support can be represented as

$$support = \frac{|\alpha \cup \beta|}{|T|}.$$

Frequent itemsets are itemsets that satisfy the minimum support. A frequent itemset that contains l items is a l-itemset.

6.4.1 Frequent Itemset on Prediction

The frequent itemset generation in an association rule algorithm first counts the frequencies of each individual item among the whole transaction. Then based on the 1-itemsets whose support are no less than the predefined minimum support, frequent 2-itemsets are generated. Those itemsets with occurrence no less than the minimum support are selected for frequent 3-itemsets generation. Frequent l-itemset are generated based on the frequent $(l - 1)$-itemset. The process continues until no new frequent itemsets are found. The l value can also be specified in the itemset generation algorithm to achieve limited itemsets within a preferred time period.

We explain in the following how to use itemsets to predict missing attribute values.

Let $T = (C, D)$ be the decision table that contains missing attribute values, where $C = \{c_1, c_2, \ldots, c_k, \ldots, c_m\}$, $1 \leq k \leq m$, and $U = \{u_1, u_2, \ldots, u_n\}$, $1 \leq n$.

First, the data input to the association rule algorithm is prepared. Data instances with missing attribute values are all removed from T, and we call the new decision table T''. T'' does not contain any missing values.

Secondly, frequent l-itemsets are generated based on T'' with a given minimum support. Let $Itemsets = \{S_1, S_2, \ldots, S_g\}$, where S_i $(1 \leq i \leq g)$ is a frequent l-itemset generated based on $T = (C, D)$ according to a minimum support, $S_i = \{v_{p_1}, v_{p_2}, \ldots, v_{p_l}\}$, l is the number of items contained in S_i, and v_{p_j} $(1 \leq j \leq l)$ is an attribute value in T.

Thirdly, we use the frequent itemsets generated in the previous step as our knowledge base to find a match for the missing value. Let $u_i = (v_{1i}, v_{2i}, \ldots, v_{ki}, \ldots, v_{mi}, d_i)$ $(1 \leq i \leq n)$ be the data instance in T containing the missing attribute value v_{ki} (represented as $v_{ki} =?$) for attribute c_k $(1 \leq k \leq m)$. We search from $Itemsets$ for all the itemsets containing the missing attribute v_k, and check which itemset among the itemsets can be **applied** to u_i. We say a frequent itemset can be applied to this data instance if all the items in this itemset, except the missing attribute, have exactly the same attribute values as contained by the data instance that has the missing attribute value. If this itemset can be applied, we assign the attribute value contained in this itemset to the missing attribute. In case there are multiple matched attribute-value pairs for the missing attribute, one of the values is randomly selected to be assigned to the missing value.

Example 7 Suppose u_i is one of the data instances in T that contain missing attribute values, $u_i = (v_{1i} = 1, v_{2i} = 2, v_{3i} = 4, v_{4i} =?, v_{5i} = 8)$. An itemset generated from T is $S = \{v_2 = 2, v_3 = 4, v_4 = 6, v_5 = 8\}$. Since all the items in S can be applied to u_i, we assign $v_{4i} = 6$.

6.4.2 ItemRSFit Approach

The frequent itemset is generated from the original data set without missing values. We use itemsets as our knowledge base to predict missing attribute. Since the knowledge base is generated with a certain support value, when support is high, the item-item relations are stronger, and the available knowledge for prediction is less. Missing attribute values from some data instances can be predicted by frequent itemsets. We call these data instances *Compatible Records*. There also exist data instances for which no possible match can be

found to predict the missing values.

Definition 7 Compatible Record. A compatible record (CR) is a record whose missing attributes can be predicted by an itemset. More formally, a record r with p missing attributes is a CR if there exists an itemset I such that $|I \cap r| \leq p$.

The missing attributes of a CR are predicted using the technique described in Section 6.4.1. If a record is not CR, the RSFit method is applied to predict the rest of the missing attribute values. We call this integrated approach **ItemRSFit**. The details on the integrated approach is shown in the following Figure 6.5.

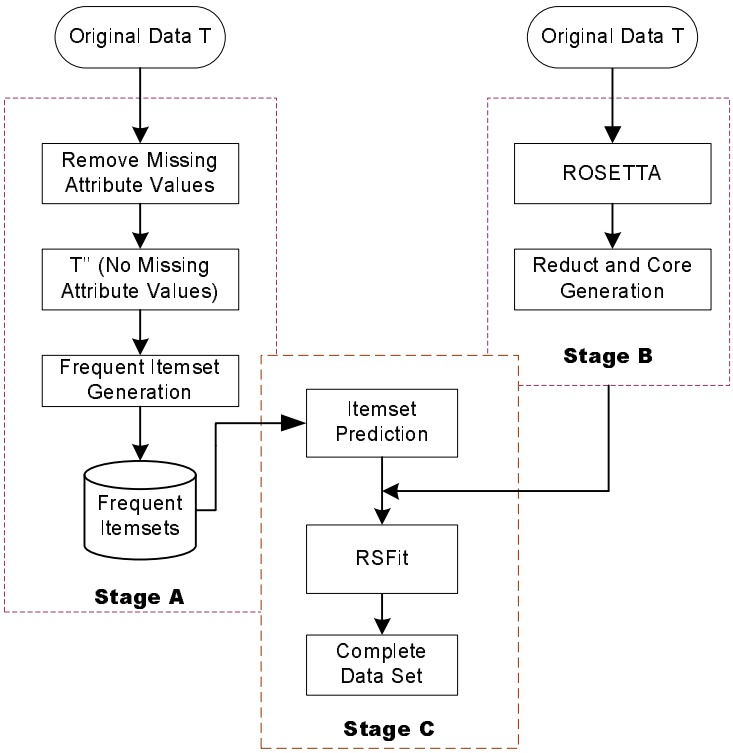

Figure 6.5: ItemRSFit Approach

The procedure of the ItemRSFit approach is shown in Figure 6.5. **Stage A** illustrates the itemset approach, in which the frequent itemsets, as the knowledge base, are generated based on using the apriori association rule algorithm from complete data instances. The

reduct and core are generated in **stage B** for the use of the RSFit approach. In **stage C**, the frequent itemsets are used to predict the missing attribute first, then the RSFit approach is applied to the rest of the missing cases.

6.4.3 Evaluation Method

We use the same approach to perform the evaluation process as the RSFit approach evaluation in Section 6.3.3. We use the following approach to perform the evaluation process. We consider complete data sets as the transaction data set T. For each data set, we randomly select a certain number of missing values among the whole data set to produce n missing attribute values per data set. We then apply both the RSFit approach and the ItemRSFit approach on predicting missing values, and compare the accuracy of the predictions from these two approaches.

6.4.4 Experimental Results for the ItemRSFit Approach

The ItemRSFit approach is implemented by Perl and the experiments are conducted on Sun Fire V880, four 900Mhz UltraSPARC III processors. We use apriori frequent itemset generation [15] to generate frequent 5-itemset. The core generation in the RSFit approach is implemented with Perl combining the SQL queries accessing MySQL (version 4.0.12). ROSETTA software [69] is used for reduct generation.

Experiments on Geriatric Care Data

We perform experiments on a geriatric care data set as shown in Table 6.5.

Table 6.10 lists the prediction accuracy comparisons for the RSFit and the ItemRSFit approaches. RSFit is used to predict missing attribute values based on the attribute-value pairs from the core or the reduct. The ItemRSFit approach is the new integrated approach introduced in this chapter. Table 6.10 lists the prediction accuracy for both RSFit and ItemRSFit according to different number of missing attribute values and different support values. We also list the numbers and the percentage of compatible records by only using frequent itemsets as knowledge for prediction. In this research, we experiment on geriatric care with 50 to 200 missing attribute values.

Table 6.10: Comparisons on Geriatric Care Data on Prediction Accuracy

Data Sets	Average Accuracy(Percentage%)				
Missing Values	RSFit	Support	# CR	% CR	Integrated ItemRSFit
50	64.00%	90%	11	22%	64.00%
		80%	22	44%	68.00%
		70%	26	52%	68.00%
		60%	38	76%	72.00%
		50%	41	82%	70.00%
		40%	43	86%	72.00%
		30%	43	86%	78.00%
		20%	46	92%	90.00%
		10%	46	92%	96.00%
100	69.00%	90%	26	26%	69.00%
		80%	53	53%	74.00%
		70%	58	58%	74.00%
		60%	69	69%	77.00%
		50%	80	80%	75.00%
		40%	87	87%	76.00%
		30%	87	87%	81.00%
		20%	95	95%	87.00%
		10%	95	95%	96.00%
150	73.33%	90%	43	29%	75.33%
		80%	85	57%	79.33%
		70%	94	63%	79.33%
		60%	120	80%	80.00%
		50%	133	89%	81.33%
		40%	137	91%	82.00%
		30%	137	91%	83.33%
		20%	142	95%	89.33%
		10%	142	95%	96.67%
200	73.50%	90%	39	20%	73.50%
		80%	103	52%	77.00%
		70%	118	59%	76.50%
		60%	146	73%	75.50%
		50%	169	84%	73.50%
		40%	182	91%	79.00%
		30%	182	91%	79.50%
		20%	192	96%	88.50%
		10%	194	96%	96.00%

From Table 6.10 we can see, the smaller the support becomes, the more itemsets are generated, and the larger the number of compatible records from frequent itemset becomes. The ItemRSFit approach always obtains higher or the same prediction accuracy as the RSFit approach.

Figure 6.6 shows the comparison for the number of compatible records by Itemsets prediction according to different support for different numbers of missing values. Frequent itemsets with lower support value can provide a larger knowledge base to find predictions, and this is not related to the number of missing values existing in the data set. We can also see from Figure 6.6 that using itemsets alone cannot predict all the missing values. For instance, when there are 50 missing values existing in the data set, given $support = 10\%$, there are still 8% of the missing instances that cannot be predicted by the itemsets.

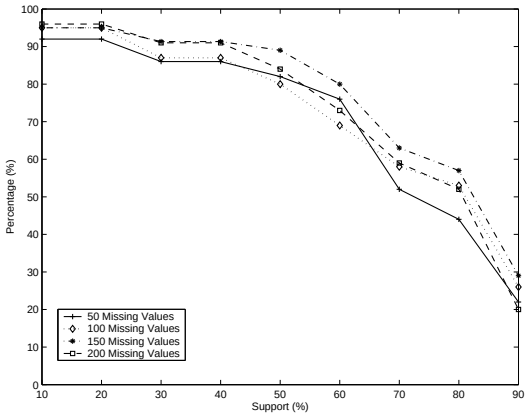

Figure 6.6: Comparisons on the Percentage of CR for Geriatric Care Data

In order to show that the ItemRSFit approach obtains better prediction accuracy than RSFit, we show the prediction accuracy comparisons on the geriatric care data set with 150 missing attribute values, as shown in Figure 6.7. We can see from Figure 6.7 when support value is lower, the prediction accuracy of ItemRSFit is significantly higher than RSFit prediction. This result demonstrates that frequent itemsets as a knowledge base can effectively be applied for predicting missing attribute values.

Figure 6.8 demonstrates the prediction accuracy comparisons for different number of missing attribute values with different support for the geriatric care data set using ItemRS-

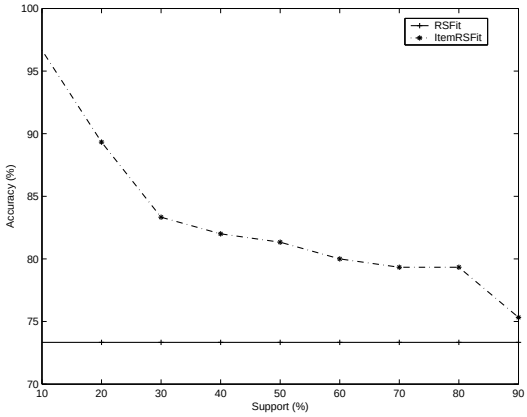

Figure 6.7: Geriatric Care Data with 150 Missing Attribute Values

Fit. We can see from the comparisons that the ItemRSFit approach obtains higher accuracy when the support value is lower. The number of missing attribute values existing in the data set does not affect this fact.

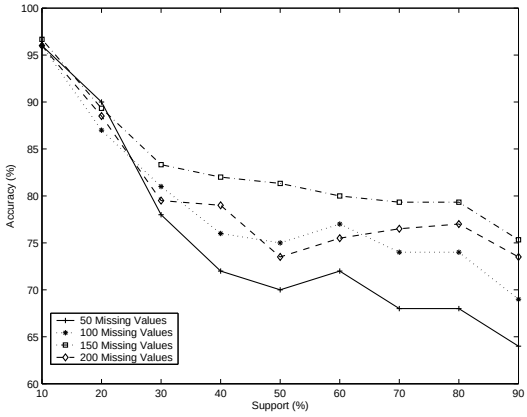

Figure 6.8: Geriatric Care Data with Different Number of Missing Attribute Values

Comparisons on the size of Frequent Itemset

We compare the size of the frequent itemsets on the prediction accuracy, with different frequent l-itemset, for $l = 5$, $l = 4$, $l = 3$, $l = 2$, and $l = 1$, on the geriatric care data

with 50 missing attribute values for *support* = 30%. The comparison results are shown in Table 6.11. For frequent itemsets whose size is larger than 7, the computation time is

Table 6.11: Comparisons on Frequent l-Itemsets for Prediction Accuracy

l-itemset	Accuracy (Percentage)	Time for Itemset (Seconds)	Time for Prediction (Seconds)	Total (Seconds)
7	78.00%	2371.943	95000.011	97371.954
6	78.00%	511.351	21942.596	22453.947
5	78.00%	84.097	4003.857	4087.954
4	78.00%	13.121	576.679	589.800
3	78.00%	1.849	86.640	88.489
2	78.00%	0.471	36.708	37.179
1	64.00%	0.342	101.403	101.745

excessive. From Table 6.11, we can see that the size of the itemsets from $l = 2$ to $l = 5$ bring the same prediction accuracy on the missing attribute, while the frequent 2-itemset gives a much faster computation time.

Discussions for the Result on Geriatric Care Data

From the experimental results shown in Figure 6.6, 6.7, and 6.8, we notice that

- The prediction accuracy for the ItemRSFit approach increases while the support value decreases.
- The frequent Itemset approach can provide a higher prediction by itself. But this approach cannot predict all the missing values in the geriatric care data set.
- For the ItemRSFit approach on the geriatric care data, the highest accuracy is obtained when *support* = 10%; the lowest accuracy is obtained when *support* = 90%. This can be explained as follows. "Support" is a measure to evaluate the occurrence of both the antecedents and the consequents of an association rule in the data set. The higher the support is, the more frequent this occurrence becomes and the less knowledge for prediction is obtained. When the support value is increased, fewer

matched cases are found from the itemset approach; therefore, more missing values have to be predicted by the RSFit approach.

- The lowest accuracy of the ItemRSFit approach is equal to the accuracy from the RSFit approach. The RSFit approach gives the baseline prediction accuracy for the ItemRSFit approach.

- For different numbers of missing attribute values, the frequent itemset with the lowest support brings the highest prediction accuracy. The frequent itemset alone as the knowledge base to predict the missing values cannot fully find all the matches for the missing value for geriatric care data.

Experiments on UCI Data Sets

In the experiments on the UCI data sets [21] we study how the ItemRSFit approach can be applied for predictions on different types of data sets. We experiment on data sets with no missing attribute values. For each data set, we randomly select 5% of the total possible missing values (total number of condition attributes $\times$ total number of data instances) as missing attribute values, and list the prediction accuracy comparisons for the ItemRSFit and RSFit approaches according to different support values.

Abalone Data This data set is used to predict the age of abalone from physical measurements. There are $4,177$ instances and 8 condition attributes in this data set. There are no missing attribute values or inconsistent data instances in the data set. For this data set, we randomly select 0.5% missing attribute values, which is 167 missing values. The prediction comparisons between RSFit and ItemRSFit approaches are shown in Figure 6.9.

Observation. As we can see from Figure 6.9, when the support value decreases, the prediction accuracy increases.

Lymphography Data The data set contains 148 instances and 18 condition attributes. There are no missing attribute values in this data. We check that there is no inconsistent data. The core is empty for this data set. We randomly select 133 missing attribute values from this data set, which is around 5% of the data set. The prediction comparisons between RSFit and ItemRSFit approaches are shown in Figure 6.10.

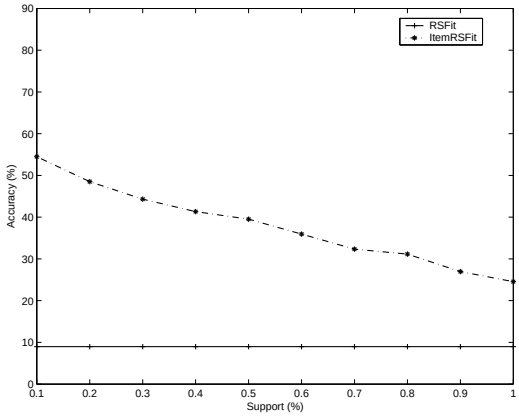

Figure 6.9: Accuracy Comparisons for Abalone Data

Observation. As we can see, when support value decreases, the prediction accuracy increases. We further explore the prediction accuracy on a smaller number of missing values with this data set, as shown in Figure 6.11. For 10 missing values, when support reaches less than or equal to 20%, the accuracy is 100%. This observation implies that a smaller number of frequent itemsets can also be used to provide high predictions for missing attributes.

Glass Data This data set is used for the study of classification of types of glass by criminological investigation. At the scene of the crime, the glass left can be used as evidence. There are 214 instances and 9 condition attributes. There are no missing attribute values or inconsistent data instances. We randomly select 96 missing attribute values from this data set, which is around 5% of the data set. The prediction comparisons between RSFit and ItemRSFit approaches are shown in Figure 6.12.

Observation. For the glass data set, the support values rank from 1% to 10% for frequent itemset generation. We can see as support decreases, the prediction accuracy increases. The highest prediction accuracy obtained when $support = 1\%$. ItemRSFit always achieves higher prediction than RSFit.

Iris Data For Iris data set, there are 4 condition attributes, 150 instances. There is no inconsistent data existing in the data. We first use the core algorithm to generate core attributes, but the result is empty. This means none of the attributes is indispensable.

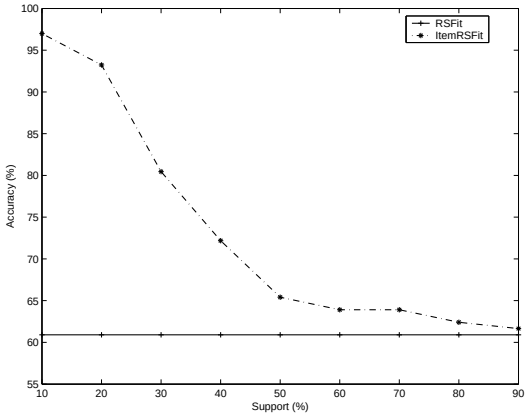

Figure 6.10: Accuracy Comparisons for Lymphography Data

We randomly select 30 missing attribute values from this data set, which is around 5% of the data set. The prediction comparisons between RSFit and ItemRSFit approaches are shown in Figure 6.13.

Observation. For the Iris data set, the support values rank from 1% to 10% for frequent itemset generation. We can see as support decreases, the prediction accuracy increases. The highest prediction accuracy of 83.33% is obtained when $support = 1\%$. The ItemRSFit always achieves higher prediction than RSFit. It is also interesting to notice how drastically the prediction accuracy increases from 20% to 83.33% within a small range of support values decreasing from 7% to 1%.

6.4.5 Discussions and Related Work

Experimental results from both the real-world geriatric care data set and UCI data sets have demonstrated the high prediction characteristics of the proposed ItemRSFit approach on processing data with missing attribute values. The frequent itemsets can be used as a knowledge base to predict missing attribute values.

We find the approach introduced in [91] close to our work. An approach of using association rules generations on completing missing values is discussed. However, our proposed ItemRSFit approach is quite different from the approach introduced in [91]. First, only frequent 1-itemset and 2-itemset are used in [91] to find the possible values for

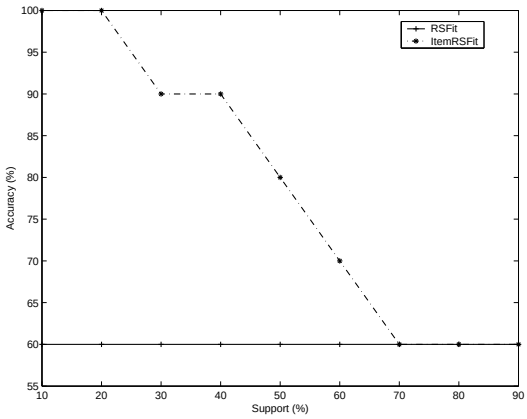

Figure 6.11: Accuracy Comparisons for Lymphography Data

the missing data, and data associations with missing attributes on the consequent part are used for prediction. It is not discussed what percentage of the missing data can be predicted with the data association. We use frequent 5-itemset as the knowledge base for prediction. We explore the relations between different support and the percentage of the *compatible records* using frequent itemsets as shown in Figure 6.6. Second, in case there is no match from the data association, the missing value is assigned by the most common value of the missing attribute in [91]. We use frequent itemsets as the knowledge base for prediction, and the RSFit approach for the *non-compatible records* where the itemset cannot be applied, which guarantee that more important attributes are taken into considerations while predicting attributes. The proposed ItemRSFit approach provides predictions based on the data domain itself, which better preserves the originality of the data sets and avoids noise. Third, in [91], data associations, which are similar to associated rules, are generated according to both support and confidence and used as a knowledge base for predictions. Our approach is more efficient because we do not need to generate associated rules based on both support and confidence for prediction. Only support is used for frequent itemsets generation in the ItemRSFit approach.

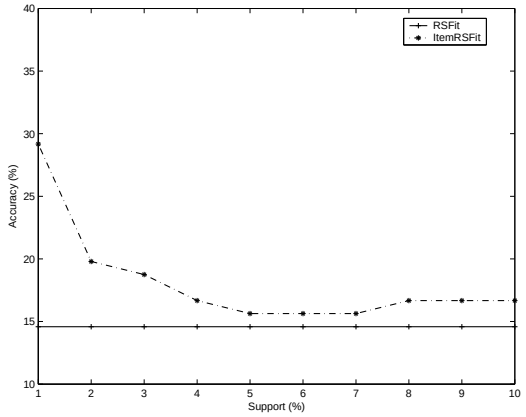

Figure 6.12: Accuracy Comparisons for Glass Data

6.5 Conclusions

In this chapter we explore a new usage of association rule algorithms on predicting missing attribute values, combining with the rough sets theory. We first introduce a new approach RSFit to assign missing attribute values based on rough sets theory. Comparing to the "closest fit" approach proposed by Grzymala-Busse [28], this approach significantly reduces the computation time and a comparable accuracy is achieved. In the second part of the chapter, we introduce an integrated approach ItemRSFit based on both association rule algorithm and rough sets theory to assign missing attribute values. The experimental results show the new approach obtains higher prediction accuracy than most of the existing approaches. It relies on its own data as a knowledge base and therefore the predicted values are not biased.

The ItemRSFit approach uses the RSFit approach for predicting non-compatible records. We would like to experiment with other techniques on predicting missing values for the non-compatible records. In our research, we also adopt the strategies used by [100] on balancing the computational cost and the prediction accuracy. Lower support value can bring a higher prediction accuracy; however, frequent itemsets with lower support requires more time for computation than frequent itemsets with higher support. In the future, we are also interested in exploring a satisfactory balance between the support value and the prediction accuracy. Given the available computational cost and the affordable compu-

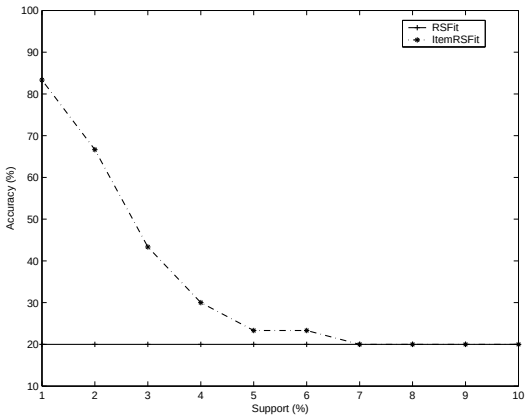

Figure 6.13: Accuracy Comparisons for Iris Data

tation time, it is interesting to explore what percentage of the missing attributes can be effectively predicted, and what are the most effective attributes to be predicted. In case of a higher prediction cost, the idea of giving more important attributes higher priorities for predictions can be applied as an heuristic. We would like to explore the relationships between the prediction accuracy of ItemRSFit and the percentage of missing attribute values contained within a data set. In our experiments, we have obtained a satisfactory predicting accuracy for the ItemRSFit approach on data sets with an average of 5% of total missing attribute values. One future study we intend to make is to determine what database characteristics (distribution of the data, functional dependencies among the data, complexity of relations, and so on) must be taken into account and how should they be taken into account to develop a parameterized analytic measure of missing values that can be meaningfully calculated.

Chapter 7

Rough Set Based Knowledge Discovery with Case Study in Personalization

The goal of knowledge discovery in databases and data mining is to extract information from a large amount of real world data and to generate such information into knowledge, in the form of explainable rules, to help human beings understand certain applications. In the previous chapters of this thesis, we study and develop techniques on the postprocessing of the knowledge (rule generation and rule evaluation in Chapter 3, 4, 5), and preprocessing of the data in order to be processed by the data mining tasks (missing attribute processing in Chapter 6). In order to demonstrate the applicabilities and the usages of the techniques developed in Chapter 3, 4, 5, 6, we consider a rough set based knowledge discovery system and explain the utilities of the proposed techniques in such systems. We then demonstrate how such techniques can be adapted to the specific application of an online user-centric personalization system through a case study, to further illustrate the value of our research.

7.1 Introduction

The general models of knowledge discovery in databases (KDD) contain processes including data preprocessing, knowledge discovery algorithms, rule generations and evaluations.

Rule evaluation is a significant process in KDD. How to automatically extract rules that are important and representative for human beings instead of selecting those useful rules manually is an interesting problem. Specific difficulties make the research of rule evaluation very challenging.

One of the difficulties is that real-world large data sets normally contain missing attribute values. They may come from the collecting process, or redundant scientific tests, change of the experimental design, privacy concerns, ethnic issues, unknown data and so on. Discarding all the data containing the missing attribute values cannot fully preserve the characteristics of the original data, and wastes part of the data collecting effort. Knowledge generated from missing data may not fully represent the original data set; thus, the discovery may not be as sufficient. Understanding and utilizing of original context and background knowledge to assign the missing values seems to be an optimal approach for handling missing attribute values. In reality, it is difficult to know the original meanings for missing data from certain application domains. Another difficulty is that a huge number of rules are generated during the knowledge discovery process, and it is infeasible for humans to manually select useful and interesting knowledge from such rule sets.

We are interested in tackling difficult problems in knowledge discovery from a rough sets perspective. In this thesis, we discuss how rough sets-based rule evaluations are utilized in knowledge discovery systems. Three representative approaches based on rough sets theory are proposed. The first approach is to provide a rank of how important each rule is by a Rule Importance Measure (RIM) (Chapter 4). The second approach is to extract representative rules by considering rules as condition attributes in a decision table, the Rules-As-Attribute Measure (Chapter 5). The third approach is applied to data containing missing values (Chapter 6). This approach provides a prediction for all the missing values using frequent itemsets as a knowledge base. Rules generated from the complete data sets contain more useful information. The third approach can be used at the data preprocessing process, combining with the first or second approach at the rule evaluation process to enhance extracting more important rules. It can also be used alone as preprocessing of missing attribute values. An interesting personalization system based on this rule-enhanced knowledge discovery system is studied. The approaches of discovering important rules are further demonstrated in this personalization system.

We discuss related work on current knowledge discovery systems based on rough sets theory in Chapter 2. Section 7.2 examines RSES as a representative rough sets based knowledge discovery system, and discusses an enhanced knowledge discovery system by integrating the techniques proposed in this thesis. Section 7.3 contains a case study of a user-centric personalization system. How the proposed rule evaluation approaches (i.e., rule important measures) can be applied to such a system are demonstrated.

7.2 Rule Evaluations and Knowledge Discovery Systems

In this section, we first examine a current rough set knowledge discovery system, and suggest the importance of rule evaluations. We then discuss how to integrate our proposed rule evaluation approaches and their functions in knowledge discovery systems. Other rough set based knowledge discovery systems are presented in Section 2.1.2 with related rule evaluations covered in Section 2.3.

7.2.1 Analyzing RSES – Rough Set Exploration System

We take the RSES (Rough Set Exploration System) [12] system introduced in Chapter 2 as an example system, and study in more detail of the role of rule evaluations. We show that current systems are limited with regard to the rule evaluations, and we emphasize the importance of rule evaluation in current knowledge discovery systems.

RSES is a well developed knowledge discovery system focusing on data analysis and classification tasks, which is currently under development. Figure 7.1 shows a use of the system on a heart disease data set for classification rule generation.

The data input to RSES is in the form of a decision table $T = (C, D)$, where C is the condition attribute set and D is the decision attribute set. Preprocessing is conducted once the data is imported to the system, during which stage the missing attribute values are handled and discretization is performed if necessary as well. Reducts are then generated and classification rules based on the reducts are extracted.

RSES provides four approaches for processing missing attribute values, such as removing

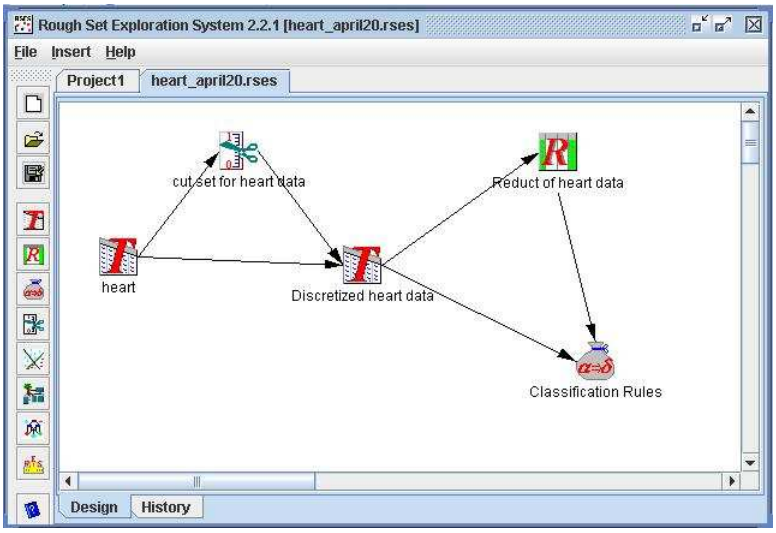

Figure 7.1: Using Rough Set Exploration System on the Heart data

data records with missing values, assigning the most common values of the missing attribute within the same decision class and without the same decision class, and considering missing attribute values as a special value of the attribute [12]. These approaches are used during the data preprocessing stage in the system. Although these approaches are fast and can be directly applied in the data, they lack the ability of preserving the semantic meanings of the original data set. Missing values may be assigned; however, the filled values may not be able to fully represent what is missing in the data.

RSES provides rule postprocessing procedures which are "rule filter", "rule shorten" and "rule generalize". "Rule filter" removes from the rule set rules that do not satisfy certain support. "Rule shorten" shortens the length of the rules according to certain parameters [12]. "Rule generalization" generalizes rules according to a system-provided parameter on the precision level. Although these rule postprocessing approaches provide an easier presentation of all the rule sets, these approaches do not provide ways to evaluate which rules are more interesting, and which rules are higher quality rules. These functions cannot provide a rank of rules according to a rule's significance to the users.

7.2.2 Enhanced Knowledge Discovery System based on Rough Sets

We present an enhanced rough set based knowledge discovery system as shown in Figure 7.2, and indicate where the new techniques proposed in this thesis would be integrated.

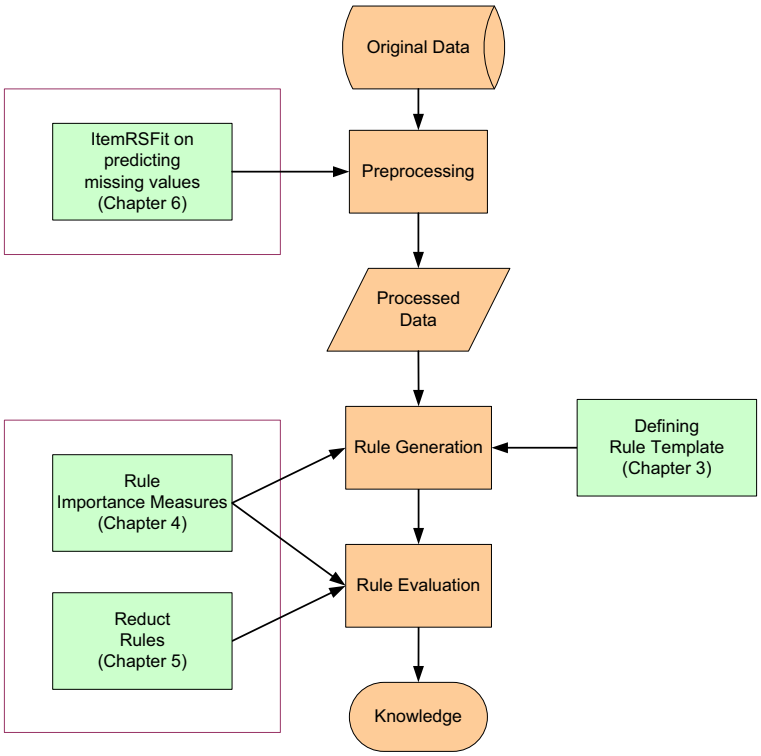

Figure 7.2: The Knowledge Discovery System Based on the Rough Sets Theory

In this general purpose knowledge discovery system, data from different application domains are first imported into the system. Preprocessing including missing attribute values processing and discretization are conducted in this stage. After the data is preprocessed, attribute selections are conducted. Depending on the output, different attribute selection approaches can be applied here. Rule generation algorithms extract rules. After the rule sets are obtained, the important postprocessing of rule evaluation is performed in this

stage. Rules are finally represented, possibly visualized in a certain format, as knowledge to the end users.

The three approaches from Chapter 4, 5, and 6 can be integrated into this general purpose KDD system as shown in Figure 7.2. The first approach *ItemRSFit* (as discussed in Chapter 6) is used in the data preprocessing stage. The second approach, *Rule Importance Measure* (as discussed in Chapter 4) is used to rank rules during the rule evaluation process. It is also applied through the rule generation procedure. The third approach, *rules-as-attributes measure* (as discussed in Chapter 5) is used only during the rule evaluation stage. We will elaborate on the utilities of these approaches as follows.

I. Predicting missing attribute values based on Frequent Itemsets.

The ItemRSFit approach is an approach on predicting missing attribute values based on association rules algorithm and rough sets theory. It has been shown on both large scale real world data set and UCI machine learning data sets on the improved prediction accuracies.

ItemRSFit approach is an integration of two other approaches from the association rule algorithm and from rough sets theory. As a first step in the association rule generation, frequent itemsets are generated based on the item-item relations from the large data set according to a certain *support*. Thus the frequent itemsets of a data set represent strong correlations between different items. When considering a certain data set as a transaction data set, the implications from frequent itemsets can be used to find which attribute value any missing attributes are strongly connected to. Thus the frequent itemset can be used for predicting the missing values. We call this approach the "itemset-approach" for prediction. The larger the frequent itemsets used for the prediction, the more information from the data set itself will be available for prediction; hence, the higher the accuracy that will be obtained. However, generating frequent itemset for large data sets is time-consuming. Although itemsets with higher support need less computation time, they restrict item-item relationships; therefore, not all the missing values can be predicted. In order to balance the tradeoff between computation time and the percentage of the available predictions, another approach is taken into consideration.

A reduct contains a subset of condition attributes that are sufficient enough to represent the whole data set. The intersection of all the possible reducts is the core. Therefore the

attributes contained in the reduct or core are more important and representative than the rest of the attributes. Thus by only examining attributes within the core or a reduct to find the similar attribute value pairs for the data instance containing the missing attribute values, we can assign the most relevant value for the missing attribute. Since this method only considers a subset of the data set, which is either the core or a reduct, the prediction is quite fast. Our approach is called "RSFit" [55], and it is an alternative approach designed for fast prediction. It can be used to predict missing attributes that cannot be predicted by the frequent itemset.

We integrate the prediction based on frequent itemsets and the RSFit approach into a new approach **ItemRSFit** to predict missing attribute values. Frequent itemsets are first used to predict missing values as the knowledge base, and the RSFit approach is then used to predict the rest of the missing values that cannot be predicted by the frequent itemsets. This approach can predict missing values from the data itself; therefore, less noise is brought into the original data. The details on the ItemRSFit approach are presented in Chapter 6.

Properly processed data can improve the quality of the generated knowledge. Therefore the ItemRSFit approach is used in this system at the preprocessing stage. It helps to preserve the qualities of the original input data to this system, thus facilitating the rule evaluation process.

II. Rule Importance Measure.

The Rule Importance Measure (as introduced in Chapter 4) is developed to provide a diverse rank of how important the association rules are, although this approach can also be applied to rules generated by other rule discovery algorithms (such as classification rule generations).

The association rule algorithm can be applied on this transaction data set to generate rules, which have condition attributes on the antecedent part and decision attributes on the consequent part. Rules generated from different reduct sets can contain different representative information. If only one reduct set is being considered to generate rules, other important information might be omitted. Using multiple reducts, some rules will be generated more frequently than other rules. We consider the rules that are generated more frequently to be more important.

If a rule is generated more frequently across different rule sets, we say this rule is *more important* than rules generated less frequently across those same rule sets.

The Rule Importance Measure is defined as follows,

Definition 8

$$\text{Rule Importance Measure} = \frac{\begin{array}{c}\text{Number of times a rule appears in all}\\ \text{the generated rules from the reduct sets}\end{array}}{\text{Number of reduct sets}}.$$

The Rule Importance Measure can be integrated into the current rough sets based knowledge discovery system to be used during the rule evaluation process. A list of ranked important rules can therefore be presented with their rule importance measures to facilitate the understanding of the extracted knowledge.

III. Rules-As-Attributes Measure.

The method of discovering and ranking important rules by considering rules as attributes is introduced in Chapter 5. The motivation comes from the concept of reduct. A reduct of a decision table contains attributes that can fully represent the original knowledge. If a reduct is given, rules extracted based on this reduct are representative of the original decision table. Extending this concept of reduct to rule evaluations, if rules are considered as condition attributes in a new decision table, the reduct of this new decision table contains important attributes, which are the rules.

We construct a new decision table $A_{m\times(n+1)}$, where each record from the original decision table $u_0, u_1, ..., u_{m-1}$ are the rows, and the columns of this new table consists of $Rule_0, Rule_1, ..., Rule_{n-1}$ and the decision attribute. We say a rule can be applied to a record in the decision table if both the antecedent and the consequent of the rule appear together in the record. For each $Rule_j$ ($j \in \{0, ..., n-1\}$), we assign 1 to cell $A[i, j]$ ($i \in \{0, ..., m-1\}$) if the rule $Rule_j$ can be applied to the record u_i. We set 0 to $A[i, j]$ otherwise. The decision attribute $A[i, n]$ ($i \in \{0, ..., m-1\}$) remains the same as the original values of the decision attribute in the original decision table.

We further define the *Reduct Rule Set* and *Core Rule Set*.

Definition 9 Reduct Rule Set. We define a reduct generated from the new decision table A as **Reduct Rule Set**. A *Reduct Rule Set* contains *Reduct Rules*.

The *Reduct Rules* are representative rules that can fully describe the decision attribute.

Definition 10 Core Rule Set. We define the intersection of all the *Reduct Rule Sets* generated from this new decision table A as *Core Rule Set*. A *Core Rule Set* contains **Core Rules**.

The *Core Rules* are contained in every *Reduct Rule Set*.

By considering rules as attributes, reducts generated from the new decision table contain all the important attributes, which represent the important rules generated from the original data set; and it excludes the less important attributes. Core attributes from the new decision table contain the most important attributes, which represent the most important rules.

This Rules-As-Attributes Measure can be integrated into the rule evaluation stages, after all the rules are generated from the original knowledge, in order to help to understand the essential knowledge of the input data.

Discussions on Rules Importance Measure and Rules-As-Attributes Measure

The Rule Importance Measure and the Rules-As-Attributes Measure can both be applied to the knowledge discovery system individually, although they are different measures. They are not to be used together, nor in any way do they compete with each other. Both measures consider the input data as a decision table. The Rule Importance Measure is applied through the rule generation procedure, the input of this measure is the original decision table, and the output is a set of rules ranked by their importance. The Rules-As-Attributes Measure takes any sets of rules as input, and it is to be used after the rules are generated. Such rules can be generated by various learning algorithms. The output of the Rules-As-Attributes Measure is a set of important rules, which is a subset of the original rule sets generated from the original data. Therefore, in situations where the given input is only a decision table, we can use the Rule Importance Measure to generate a list of rules with their rankings of importance; in the situations when there exist a set of rules already, we can use the Rules-As-Attributes Measure to extract the important rules.

Other Enhancements.

The utilities of the three approaches discussed in this thesis have been demonstrated via the enhanced rough sets based knowledge discovery system presented in Figure 7.2. They are proposed to facilitate the evaluation of the rules. There are other techniques that can be used along with these approaches. For example, during the rule generation process, properly defined rule templates (as discussed in Chapter 3) can not only reduce the computation of rule generations, but can also ensure high quality rules, or interesting rules generated according to the application purposes. Important attributes, such as probe attributes [71], can be defined in the data preprocessing stage for generating rules containing such attributes for generating expected rules.

Our motivation is proposing approaches to enhance current knowledge discovery system, to facilitate the knowledge discovery of more interesting and higher quality rules.

7.3 Case Study

This section provides a case study [1] to illustrate how a rough sets based knowledge discovery system provides a useful mechanism for a real-world user-centric personalization system using the enhancements proposed in this thesis. We demonstrate the rule evaluation techniques proposed in the earlier chapters through a real-world application.

Personalization towards individuals recently became an important focus for business applications, such as personalized home pages and a personalized shopping cart. In an online shopping application, individuals' online purchasing patterns and online browsing experiences may be personalized as well. Such personalization is helpful to predict customers' interests and to recommend relevant advertisements of interested products to facilitate customers' online shopping experiences. However an online web user normally browses hundreds of web pages before making a purchase online, and different users visit different websites. Personalization based on other people's past histories may not be very interesting to another user. A user-centric personalization system based on an individual user's search histories is needed for precise personalization.

[1] The work shown in this section is a collaboration with the Decision Technology Department at Hewlett-Packard Labs in Palo Alto, California. The author was employed as a summer intern in Summer 2006.

The following Figure 7.3 illustrates the prototype of a potential user-centric personalization system, combining data mining and machine learning algorithms on predicting online product purchases. User-centric data is collected and stored in the databases. Features related to user-centric clickstream data are selected and the data is preprocessed for the prediction engine. The search terms users input into the search engines, and the search terms they use on the leading shopping online stores are considered as strong indications of the purchasing interests, and the terms are categorized first to classify potential users into different product purchasing categories. Classification algorithms such as decision tree [77], logistic regression [5] and Naïve Bayes [64], association rules algorithms such as apriori [3], and other prediction algorithms are applied in the following steps to further predict whether a user is an online buyer or non-buyer according to the observed browsing behaviours across multiple websites.

As part of the HP adaptive user-centric personalization project shown in Figure 7.3, one possible approach of modeling users' browsing behaviours is to study their browsing histories across multiple websites. Personalized products and advertisements through predictions generated by such models can be of great benefit from the business point of view.

We first survey the current personalization systems in Section 7.3.1. We discuss the differences between user-centric and site-centric data, and the differences between the personalization techniques deployed for these two types of data. We study the clickstream data collected from an online audience measurement company as our test data. We show through empirical experiments how the Rule Importance Measure, as an example of the rule evaluation approaches proposed in this thesis to enhance the knowledge discovery systems, can be applied towards the user-centric personalization system to extract important rules on predicting online buyers.

7.3.1 Personalization Systems

Clickstream data collected across all the different websites a user visits reflect the user's behaviours, interests, and preferences more completely than data collected from the perspective of a single website. For example, we would expect that we could better model and predict the intentions of users who we know not only searched on Google but also

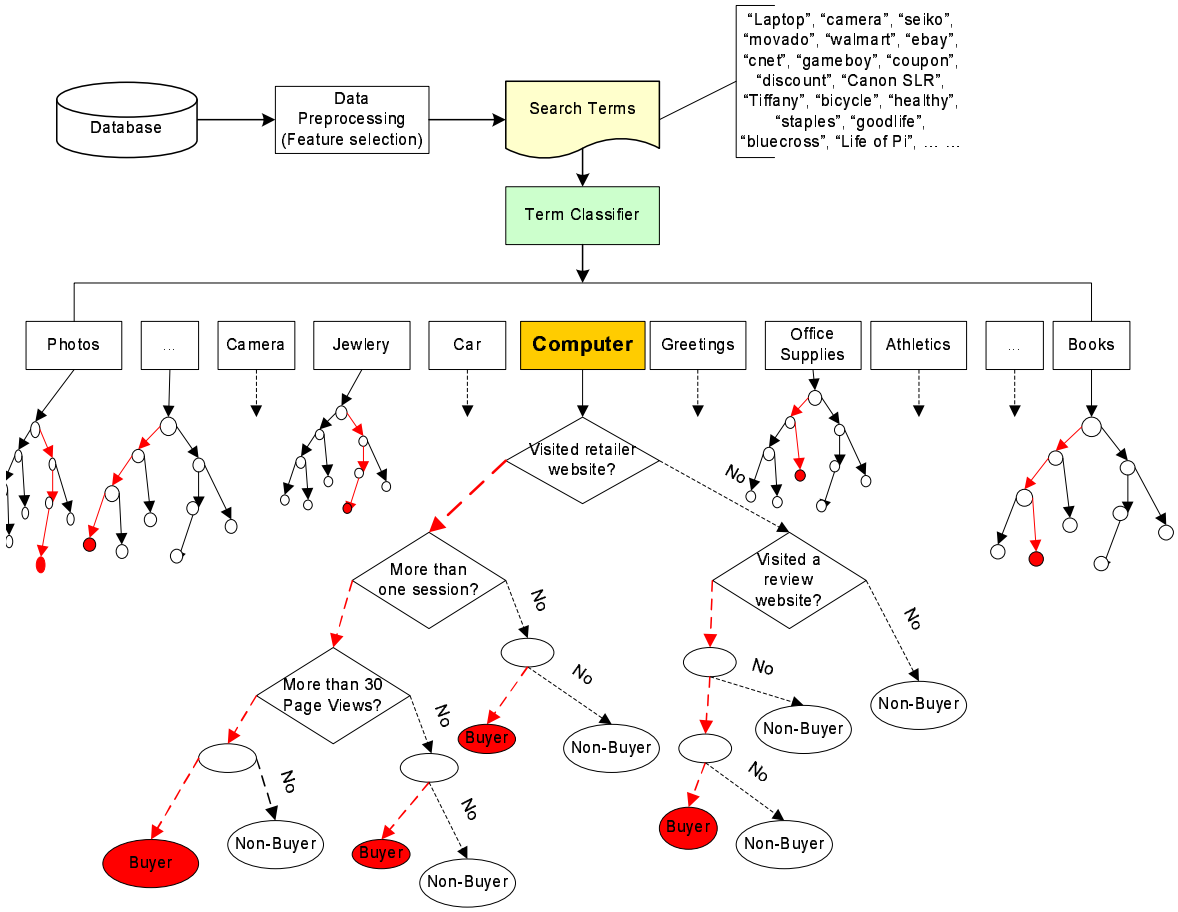

Figure 7.3: Prototype for Online Product Purchasing System

visited the HP shopping website and the Dell website, than if we know only one of those pieces of information. The complete data set is termed user-centric data, which contains site-centric data as a subset. Current research on clickstream data analysis is centered around site-centric data [70]. The site-centric personalization systems collect customers' browsing histories based on the clickstream data from the individual web site perspective, and personalizations are generated according to these clickstream data to recommend items to the internet users who browse this specific web site. Predictions can be generated for a new customer based on the profile matching of the existing customers (such as name, location, gender, occupation, IP address, operating system and browser information), browsing histories (such as the web pages the customers visited during a certain period of time, application tasks and their sequences the customers performed during a certain period of time), and the preferences of the browsed items (such as some customers expressing great interests on specific items or tasks, whereas some customers show no interest on the same items or tasks).

Each customer thus has his/her individual profile collected. The more customer profiles a personalization system collects, the more data becomes available for precise recommendations. These user profiles are then saved either in flat files or are loaded into the databases. After the data is collected, preprocessing of the original data is conducted, which includes tasks such as missing values processing, discretization, normalization and so on.

Then, personalizations are generated by certain rule generation algorithms, such as association rule algorithms, clustering algorithms, classifications and so on. The amount of personalization can be huge when first generated; therefore, post-processing for the generated results is performed in this stage.

The rules generated based on customers' profiles therefore serve as the available knowledge base for personalization systems. In the real world situation, when the personalization system observes a new customer whose profile is an exact match or similar match to the profiles in the databases, the recommendations from the personalization database are generated and provided to this new customer.

Figure 7.4 shows a model for a site-centric data personalization system. Data collected from different users, including the browsing histories, personal preferences and demographic data, are sent for creating the personalization engine. When a new customer comes, based

on the browsing histories and the demographic information, the engine recommends personalized interesting items (such as web pages) to this new user.

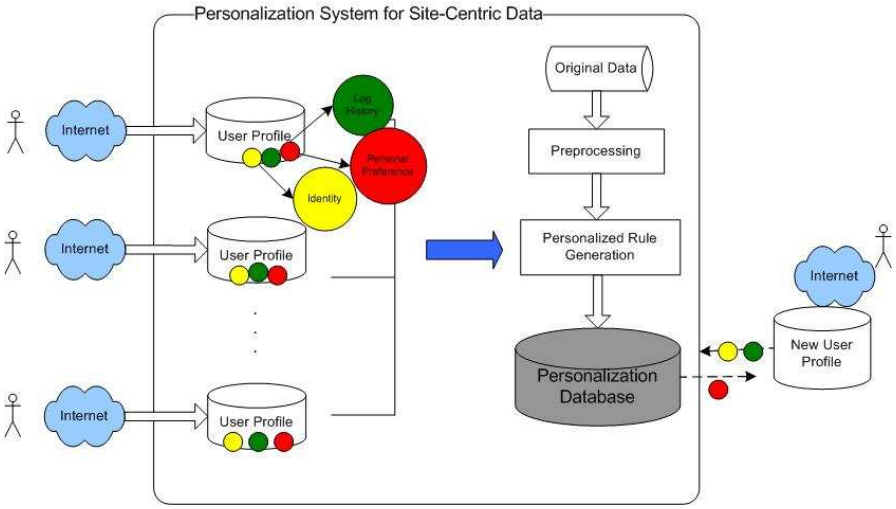

Figure 7.4: Personalization for Site-Centric Data

7.3.2 User-Centric Personalization

The personalization systems introduced above are also called "site-centric personalization systems". Such systems make predictions for "site-centric data", which is data collected on one single server [70]. Site-centric personalization systems collect customers' browsing history from the clickstream data on the web site side, and personalizations are made based on these clickstream data from the site to recommend items to the customers who browse this specific web site.

Much current research on clickstream data personalization focuses on site-centric personalization [70]. It is important to study the difference between user-centric data and site-centric data, to determine the potential value of the user-centric approach. Due to the limitations of site-centric data it is difficult to fully capture customers' online shopping behaviours for precise personalization modeling and predictions. We explain one of the difficulties in the following example.

Let us consider an online shopping and retail website such as shopping.com or Amazon.com as an example, which we call site-centric website. Consider the online clickstream data collected by this site as site-centric data. Knowledge such as customers' demographic information, the web pages the customers visited, the time the customers spent on each of the particular web pages, the incoming and outgoing URLs for each of the customers and so on are collected on the server side. Information for the previous web pages each customer visited is collected, and recommendations based mostly on the buyers'(customers who are observed to make a purchase at this site) behaviours are suggested. For those customers who visited but do not make a purchase at such sites, although later they may make a purchase elsewhere (e.g., HP shopping websites), this site captures the browsing histories for people who visited, but such browsing information is not considered to be important for making online product purchase predictions. The available information is thus not fully captured and utilized. On the other hand, demographic information for customers' background are also taken into consideration for making recommendations. Therefore, customers' privacies are not well protected.

User-centric studies are proposed for personalization based on customers' individual behaviours while greatly preserving customers' privacy issues. User-centric data is collected to capture each of the individual customers' browsing behaviours. Data containing customers' browsing histories, purchasing histories, and so on are then processed for personalization generation. In user-centric data personalization, the limitations of not effectively capturing complete information collected from only certain sites no longer exist. Users' web search histories across multiple websites are all used towards the construction of the engine. User-centric personalization has the advantage of protecting users' privacy as well. By considering more complete information collected from the users, without using their demographic information, the personalized model can fully capture the behaviours while greatly taking care of the privacy issues without using the individual's demographic information (such as users' login names, zip code, age, occupations and so on).

The personalization techniques for site-centric data are quite mature, which are techniques originating from traditional web log mining, machine learning, data mining and so on. Given the differences between site-centric data and user-centric data, it is important to study whether these site-centric personalization techniques can be applied to user-centric

data, and whether new issues (in terms of data collection, data preprocessing, user behaviour modeling and so on) and new challenges should be taken into consideration for user-centric data personalization.

User-centric data is collected for each of the individual user. The data contains users' browsing histories on all the web sites they visit and their own preferences of interested web sites. Figure 7.5 depicts a sample user-centric personalization system.

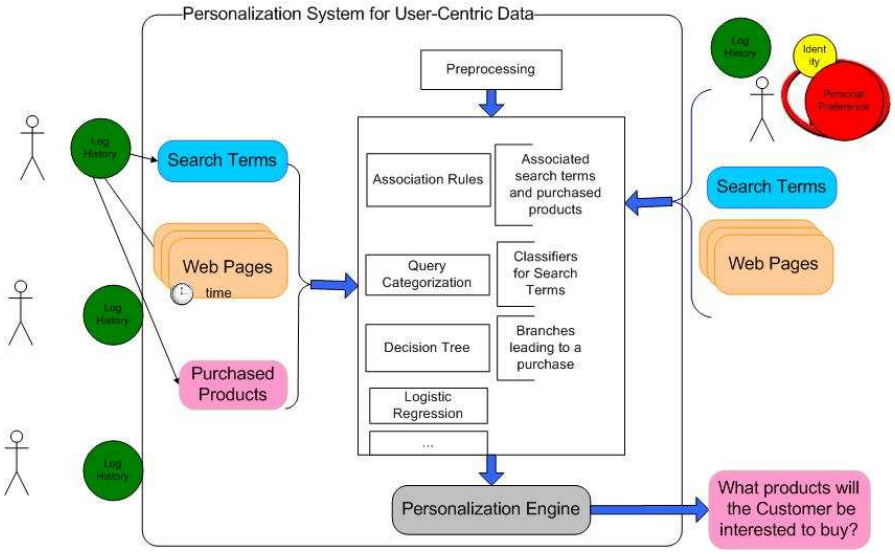

Figure 7.5: Personalization for User-Centric Data

7.3.3 Differences between User-centric and Site-centric Data

We summarize the difference between user-centric and site-centric data.

- How is the data collected? Site-centric data is collected from a particular web site due to the limitations of accessing other web sites. User-centric data is collected based on each of the individual users. The data contains clickstreams from multiple web sites that users browsed.

- What is contained in a session? Given a session containing all the users' browsing history within a limited time sequence (i.e., 30 minutes), the session data for site-

centric data contains all the webpages a user visited on one website; the session data for user-centric data contains webpages a user visited on multiple websites.

7.3.4 Related Work

Zhu *et al.* [98] recently developed a user-side web personalization system "Web-IC" to predict information content (IC) pages that a web user will be interested to visit. The motivation of this system is to help web users locate these IC pages everywhere on the web for the users themselves based on their own behaviours. The words contained in the web pages a user visits, as well as the actions (such as back pages browsing or follow-up pages) the user makes on such pages are taken into consideration as users' interests for behaviour modeling. It is shown that classifiers built from such features as extracted from user-side browsing properties can effectively predict the interested webpages for the users [99].

Although our work is similar in the fact that we both are interested in personalization towards the user-side, the purposes of our experiments and the background of the adaptive web personalization project are different from this work. We do not consider the content information (words) inside the webpages; we do not collect the user's web actions on the web pages (such as back page browsing, follow-up pages). We are interested in predicting online product purchases instead of predicting interested web pages. We are also interested in studying how site-centric algorithms can be adapted for user-centric personalization.

Other researchers studying user-centric personalization include Lieberman, who developed the Letizia web search agent for web page recommendation [62], Billsus and Pazzani, who query users to get feedback for recommending news web pages [14] and Ardissono *et al.* who customize the presentation of a website advertising a product to a user, based on a monitoring of the user's interests [8]. These researchers are more focused on user modeling and machine learning, whereas we are most interested in developing effective data mining techniques.

7.3.5 Experimental Data

Nielsen//NetRatings MegaPanel data [2] is used as our testbed for this adaptive web personalization project. Nielsen is an online audience measurement company, which is the premier provider of the media-quality internet data. The MegaPanel data offers the overall, in-depth profiles of customer behaviours. The data is collected over the complete customers' online search experiences on both leading search engines (such as Google, Yahoo) and shopping websites (such as Amazon, BestBuy). The data collection processes are designed in such a way that the average customers' online behaviours and their retention rate are consistent with the goal of representative sampling of internet users.

The data collected over 8 months (from November 2005 to June 2006) amounted to approximately 1 terabyte from more than $100,000$ households. For each URL there are time stamps for each internet user. Retailer transaction data contains more than 100 online leading shopping destinations and retailer sites. The data also contains travel transaction data, such as air plane, car and hotel reservation histories. There is also users' search terms collected in the URL data. The search terms are collected from top search engines and comparison shopping sites. In addition, additional search terms are extracted and customized by HP Labs (e.g., from Craigslist.org, which is a website for online classifiers and forums).

7.3.6 Experimental Design

The essential purpose of this adaptive web personalization project is to predict users' online purchasing behaviours based on all the websites the user visited. The clickstream data collected therefore includes not only clickstream data from one single website, but also all the other websites the users visited. The motivation of the experiment in this section is to demonstrate the usage of the proposed rule evaluation approaches in this thesis on a real-world application as a case study. The Rule Importance Measure from Chapter 4 is applied to rank the important rules which are extracted from the experimental data to predict the potential online buyers for certain products.

We first describe the features constructed from the original user-centric data used in

[2] http://www.nielsen-netratings.com/

this experiment.

Feature Construction

Features are important elements representing the experimental data. Feature construction is usually conducted in the data collection process. In our experimental data, features reflecting online purchases are not directly available from the original source of the data. Since our purpose of this section is to predict online product purchases using user-centric data, we focus on constructing features that can reflect the users' browsing and searching behaviours across multiple websites. There are 26 online product categories available in our experimental data. In this experiment, we limit the online purchasing product category to be personal computers, including desktops and laptops.

The intuition for extracting such features towards our user-centric personalization task is that, during an online purchasing event, in general, people would first search the product category in general on the leading search engines (such as Google or Yahoo); then, they would visit the retailer websites (such as Dell) who sell this product for detailed product information. After knowing more about this product, people would check how other customers consider this product in some review websites (such as CNET); when they are close to purchase, they would most likely look for coupons or discounts for a specific product.

Since site-centric data are collected as a subset of user-centric data, traditional features for site-centric clickstream analysis are considered as part of our feature sets. Features such as "the number of sessions the user spent on certain website", "the sub URLs visited", "the total time spent per session of visit" and so on are extracted.

User-centric features related to searches across multiple websites such as "search terms used across multiple search engines and websites", "whether visited retailer websites", "whether visited review websites", "whether made an online purchase" and so on are extracted.

According to the above mentioned site-centric and user-centric related features, we construct 28 features that are used in the following experiments for predicting purchase of personal computers, as shown in Table 7.1. December 2005 data is used for this experiment. In the feature descriptions, "NNR" stands for Nielsen//NetRating; HP customized sites stand for additional searches or websites extracted and customized by HP Labs (such as

Craigslist). The HP customized sites includes all the sites pre-classified by NNR.

Decision Table

December 2005 data is used for this experiment. We consider the 28 features as shown in Table 7.1 as condition attributes, we consider whether a person is a buyer or non-buyer for personal computers in December 2005 as the decision attribute. For a decision table $T = (C, D)$, $C = \{$feature sets containing 28 features$\}$, $D = \{$buyer, non-buyer$\}$. With $83,635$ users and 28 features, we create a decision table as shown in Table 7.2.

After the data is processed in the format of a decision table, we then apply the equal frequency [18] approach to discretize the data.

Rule Importance Measures

Recall the generation of the Rule Importance Measure in Figure 4.1 (Chapter 4). After the input data is preprocessed, the multiple reducts are generated. We use the genetic algorithm provided by RSES [12] for multiple reducts generation. The reducts are shown in Table 7.3.

We use apriori association rule generation to obtain prediction rules. Since the goal of this experiment is to predict whether an internet user is a potential online buyer of personal computers, our interest is to generate rules which lead to the predictions of buyers or non-buyers of computers. We specify the following two rule templates as shown in Eq. 7.1 and Eq. 7.2 that are applied during rule generations.

First, we specify that only decision attributes (buyer or non-buyer) can be on the consequent part of a rule, and there may exist more than one feature on the antecedent part of the rule. The antecedent leads to a decision (buyer or non-buyer) which is represented by the consequent part.

$$\langle Feature_1, Feature_2, \ldots, Feature_n \rangle \rightarrow \langle Decision \rangle \tag{7.1}$$

Secondly, we specify the subsumed rules using the following constraint. Given a rule represented by Eq. 7.2.

$$\langle Feature_1, Feature_2 \rangle \rightarrow \langle Decision \rangle \tag{7.2}$$

Table 7.1: 28 User-Centric Features for Online Computer Purchases in December 2005 Data

No.	Feature ID	Feature Description	No. of Users who satisfy the feature	Value range
1	G1a	Whether searched "laptop" before purchasing on Google	279	{Yes, No}
2	G1b	# of sessions this user searched "laptop" before purchasing on Google	279	{0, . . . , N}
3	G1c	# of sessions this user searched "laptop" before purchasing on all NNR	647	{0, . . . , N}
4	G1d	# of sessions this user searched "laptop" before purchasing on all NNR & HP customized search	1,012	{0, . . . , N}
5	G2a	# of page views on Google before purchasing	41, 778	{0, . . . , N}
6	G2b	# of page views on all NNR before purchasing	69, 219	{0, . . . , N}
7	G2c	# of page views on all HP customized search before purchasing	70, 192	{0, . . . , N}
8	G3a	# of sessions on Google before purchasing	41, 778	{0, . . . , N}
9	G3b	# of sessions on all NNR before purchasing	69, 219	{0, . . . , N}
10	G3c	# of sessions on all HP customized sites before purchasing	70, 192	{0, . . . , N}
11	G5a	# of page views per user who searched "laptop" on Google before purchasing	279	{0, . . . , N}
12	G5b	# of page views per user who searched "laptop" on all NNR websites before purchasing	647	{0, . . . , N}
13	G5c	# of page views per user who searched "laptop" on HP customized websites before purchasing	1,012	{0, . . . , N}
14	G6c1	# of sessions a user visited a hardware manufacturers or multi-category computers/consumer electronics sites and NNR sites before purchasing	48, 130	{0, . . . , N}
15	G6c2	# of sessions a user visited a hardware manufacturers or multi-category computers/consumer electronics sites and HP customized sites before purchasing	48, 627	{0, . . . , N}
16	G6d1	# of page views a user visited a hardware manufacturers or multi-category computers/consumer electronics sites and NNR sites before purchasing	48, 130	{0, . . . , N}
17	G6d2	# of page views a user visited a hardware manufacturers or multi-category computers/consumer electronics sites and HP customized sites before purchasing	48, 627	{0, . . . , N}
18	G15	# of sessions the user searched "coupon" or "review" before purchasing	3,208	{0, . . . , N}
19	G6a	Whether this user visited the hardware manufacturers or multi-category computers/consumer electronics sites and HP customized sites before purchasing	48, 627	{Yes, No}
20	G6b	Whether this user visited the hardware manufacturers or multi-category computers/consumer electronics sites and NNR sites before purchasing	48, 130	{Yes, No}
21	G20a	Whether this user visited the hardware manufacturers or multi-category computers/consumer electronics sites before purchasing	50, 041	{Yes, No}
22	G20c	# of sessions this user visited the hardware manufacturers or multi-category computers/consumer electronics sites before purchasing	50, 041	{0, . . . , N}
23	G20d	# of page views this user visited the hardware manufacturers or multi-category computers/consumer electronics sites before purchasing	50, 041	{0, . . . , N}
24	G14a	Whether this user made a purchase (of any product category) in the past month (November)	25, 029	{0, . . . , N}
25	G14b	Whether this user made a purchase of computer hardware, or computer software, or consumer electronics categories in the past month (November)	5, 400	{Yes, No}
26	G14c	# of purchases of computer hardware, computer software, or consumer electronics category the user made in the past month (November)	5, 400	{0, . . . , N}
27	G11	# of time (seconds) this user spent to visit the hardware manufacturers or multi-category computers/consumer electronics sites before purchasing	50, 041	{0, . . . , N}
28	G16	Whether this user visited a review site before purchasing (In the URL table, pag_addres contain %cnet%)	12, 323	{Yes, No}

Table 7.2: Decision Table for Classifications

User ID	Condition Attributes 28 Features								Decision Attribute {buyer, non-buyer}
ID	G1a	G1b	G1c	G1d	...	G14c	G11	G16	{buyer, non-buyer}
1	Yes	2	0	2	...	7	5200	No	buyer
2	Yes	5	1	7	...	2	413	Yes	non-buyer
3	No	0	0	1	...	0	622	No	buyer
...	...	...	...	...	...	...	...	...	...
83,635	Yes	1	0	3	...	0	342	No	buyer

Table 7.3: Reducts Generated by Genetic Algorithm for Decision Table 7.2

No.	Reduct Sets
1	{G2c, G3a, G3b, G14a, G11, G6b, G16}
2	{G2a, G2c, G3b, G6d1, G14a, G11, G16}
3	{G2a, G2c, G3b, G6c2, G14a, G11, G16}
4	{G2a, G2b, G2c, G6d1, G14a, G11, G16}
5	{G2a, G2c, G3b, G14a, G11, G6a, G16}
6	{G2a, G2c, G3b, G20a, G14a, G11, G16}
7	{G2c, G3a, G3b, G6c2, G14a, G11, G16}
8	{G2b, G2c, G3a, G20c, G14a, G11, G16}
9	{G2c, G3a, G3b, G20d, G14a, G11, G16}
10	{G2c, G3a, G3b, G20a, G14a, G11, G16}

the following rules

$$\langle Feature_1, Feature_2, Feature_3 \rangle \rightarrow \langle Decision \rangle \tag{7.3}$$

$$\langle Feature_1, Feature_2, Feature_6 \rangle \rightarrow \langle Decision \rangle \tag{7.4}$$

can be removed because they are subsumed by Eq. 7.2.

The classes of online buyers and non-buyers are very imbalanced in this data set. Among the $83,635$ number of users, only 449 are buyers, which take 0.53% of the total number of users. It is trivial to obtain higher confidence rules by simply generating rules to predict the non-buyers based on the features. However, this will not satisfy the purpose of such research of predicting online buyers. We therefore set the values of support and confidence to be lower in order to generate rules that can be used to predict both buyer and non-buyers. We generate the rule sets based on these 10 reduct sets with $support = 0.01\%, confidence = 5\%$.

There are 75 rules generated by using the Rule Importance Measures and rule templates. We rank their rule importance, as shown in Table 7.4. In comparison, without the rule templates or using reducts, $16,178,963$ rules are generated.

How to Interpret the Rules

Let us take two rules from Table 7.4 as examples.

Rule No.1: *If an online user has not searched on any of the HP customized search sites, but this user made an online purchase (of any product category) in the previous month, and this user spent more than 622 seconds visiting a hardware manufacturer or multi-category computers/consumer electronics sites, then this user may be a potential online buyer of personal computers.*

Rule No.3: *If an online user did not visit any hardware manufacturer or multi-category computers/consumer electronics sites, then this user may not be a potential online buyer of personal computers.*

Discussions

The Rule Importance measures provide an efficient view for important and representative knowledge contained in this user-centric clickstream data. Such extracted rules are useful to

Table 7.4: The Rule Importance for Decision Table 7.2

No.	Selected Rules	Rule Importance
1	G2c=0, G14a=1, G11 $\geq$ 622 $\rightarrow$ buyer	100%
2	G16=0 $\rightarrow$ non-buyer	100%
3	G11=0 $\rightarrow$ non-buyer	100%
4	G3b=0, G11 $\geq$ 622 $\rightarrow$ buyer	80%
5	G2c<13, G3a=0, G14a=0, G11 $\geq$ 622 $\rightarrow$ buyer	50%
6	G2a=0, G2c<13, G14a=0, G11 $\geq$ 622 $\rightarrow$ buyer	50%
7	G2b $<$ 10 $\rightarrow$ non-buyer	20%
8	G2c $<$ 13, G20a = 1, G14a = 1, G16=0 $\rightarrow$ buyer	20%
9	G2b $<$ 10, 1 $\leq$ G20c $<$ 4, G14a=1 $\rightarrow$ buyer	10%
10	G2a=0, G2c $<$ 13, G11 $\geq$ 622, G6a=1 $\rightarrow$ buyer	10%
...	...	...

predict whether an online purchase will happen for certain users according to the observed online searching and browsing behaviours.

In order to generate rules for possible online buyer prediction, the value of the support is quite low. The following study explains this situation. According to a study published by comScore [3] in December 2004 about the results for internet users' potential on purchasing electronics and computer products, the results indicate that the 92% of the internet users purchase the products offline after searching on the internet. Only a small percentage of internet users would make an online purchase eventually, although 85% of such purchases happen after 5 or 12 weeks of the initial search. The current experimental data we consider contains users' online browsing behaviours occurring within one month. Therefore the occurrence of online purchasing in our data is low.

Through our case study, we also found certain user-centric features are more important than others on predicting online purchases, namely the features that arise in the Rule Importance Measure. For example, the number of page views an internet user spent on

[3]http://www.comscore.com/press/release.asp?press=526

search websites is an indication of this person's interests. The fact that some user made a purchase online previously indicates such a user is more likely to conduct online purchases.

Other Experiments

For this user-centric web personalization application, in addition to the case study of using the Rule Importance Measure for evaluating important rules for online purchases, we also conducted related experiments on using classification algorithms including decision trees, logistic regression and Naïve Bayes for online product purchasing prediction based on this user-centric experiment data. We discuss briefly the experimental design and the classification algorithms as well as the experimental results. They serve as preliminary work on applying classification algorithms for user-centric web personalization purchasing predictions.

We use cross validation through the rest of the experiments. For the complete data in the form of a decision table 83635×29 as shown in Table 7.2, we performed 2-fold cross validation (with 50% for training, 50% as testing data, and iterating the process to average the performance results). The experiments on decision tree classification are conducted on Suse Linux v9.2. (Intel Xeon(TM) CPU 2.80GHz, 2 processor with 1.5G RAM). The experiments on logistic regression and Naïve Bayes are conducted on a PC with Pentium M CPU 1.86GHz, 1.5G RAM.

Given a confusion matrix as shown in Table 7.5, we use the following evaluation metrics [20] to evaluate classification performance. Below T stands for "true", F stands for "false", P stands for "positive" and N stands for "negative".

Table 7.5: Confusion Matrix

	Actual buyer	Actual non-buyer
Predicted buyer	TP	FP
Predicted non-buyer	FN	TN

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$
$$TruePositiveRate = \frac{TP}{TP + FN}$$
$$FalsePositiveRate = \frac{FP}{FP + TN}$$

Decision Tree Decision tree learning is an approach to approximate discrete-valued functions that can be represented by a decision tree [77]. The trees can then be used to construct classifiers for predictions. In a decision tree, the interior nodes stand for the condition attributes, the leaves stand for the classes, the path from the interior nodes leading to a leaf stands for a rule which satisfies the conjunction of condition attributes on this path and leads to a decision. A decision tree algorithm uses information gain to select the attributes from which to start branching. The attribute with the highest information gain is chosen as the splitting attribute for the current node. For a given decision attribute C (we assume only buyer or non-buyer as the two classes in our discussion), the information gain is

$$I(buyer, non - buyer) = -\sum_{i=1}^{2} p_i log_2 p_i$$

There are different decision tree implementations available. We use C4.5 decision tree [77] implementation for classification rule generation [4]. This implementation avoids overfitting of the data, performs rule post-pruning and is able to handle continuous attribute values as well.

The experimental results are shown in Table 7.6.

Table 7.6: Decision Tree Classifier

	DecisionTree
Precision	26.76%
Recall	8.48%

[4]Source code is downloaded from http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html

The result indicates that a decision tree can be used as a classifier for online product purchasing prediction [5]. Classifiers can be created based on user-centric features to predict the potential buyers. The branching node in the decision tree splitting a potential buyer and non-buyer can be detected and be used for suggesting personalized/recommended items such as potentially interesting products.

Logistic Regression We use Weka's [6] logistic regression implementation for creating the classifier based on logistic regression. This statistical regression model can be used for binary dependent variable predication. By measuring the capabilities of each of the independent variables, we can estimate the probability of a buyer or non-buyer occurrence. The coefficients are usually estimated by maximum likelihood, and the logarithm of the odds $\lg(\frac{p}{1-p})$ is modeled as a linear function of the explanatory variables [5], which are the 28 features from our data. The probability of buyer can be estimated by

$$P = \frac{\epsilon^{\alpha+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n}}{1+\epsilon^{\alpha+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n}}$$

The default cutoff threshold of predicting a buyer is p = 0.5. The classification results are shown in Table 7.7. By adjusting different cutoff thresholds, we can obtain various sets of classification performance according to different thresholds.

Table 7.7: Logistic Regression Classifier

	Logistic Regression (p=0.5)
Precision	18.52%
Recall	2.23%

By varying the cutoff threshold, we plot the following comparison curves. Figure 7.6 shows the precision and recall curve for the user-centric classifier generated by logistic regression. Figure 7.7 shows the ROC curve [76] (False Positive Rate vs. True Positive

[5] These turn out to be respectable figures according to marketing executives at HP, compared to other non-data-mining methods such as using single SQL aggregation to calculate precision and recall.

[6] Downloaded from http://www.cs.waikato.ac.nz/ml/weka/

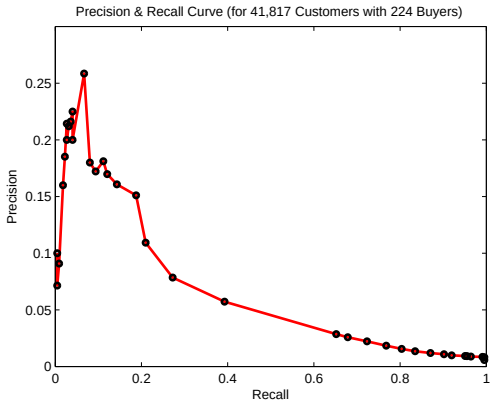

Figure 7.6: Precision vs. Recall

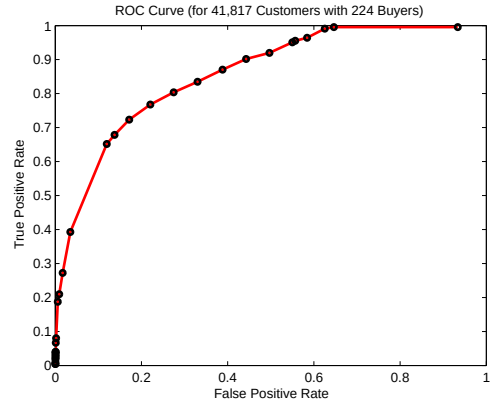

Figure 7.7: ROC Curve

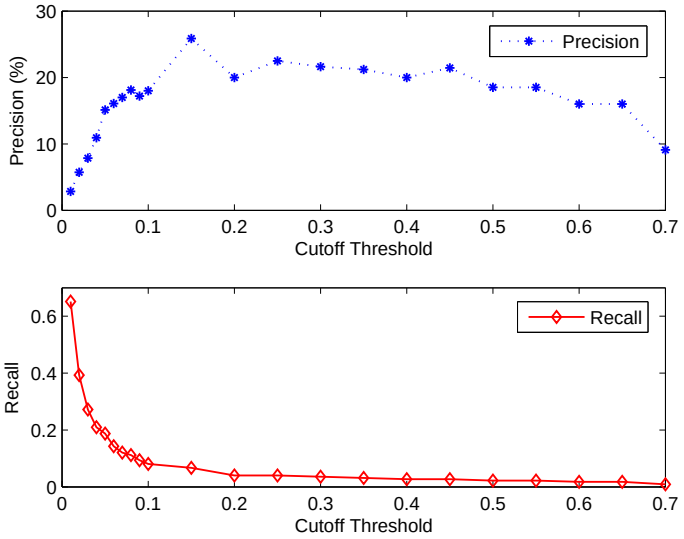

Figure 7.8: Cutoff Threshold vs. Precision and Recall

Rate) for the user-centric classifier generated by logistic regression.

Figure 7.8 shows the tradeoff between the cutoff threshold and precision/recall for the user-centric classifier generated by logistic regression. This plot can be used for determining the suggested cutoff threshold in order to reach a satisfied precision and recall towards certain classification applications.

Naïve Bayes Previous studies have shown that a simple Naïve Bayesian classifier has comparable classification performance with decision tree classifiers [42]. Naïve Bayes classifiers [64] assume that the effect of an individual attribute on a given class is independent of the values of the other attributes. Let H be the hypothesis that a person is a buyer C, the probability that H holds given the data set X is $P(H|X)$, the Bayes theorem is

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Given a set of condition attributes $\{a_1, a_2, \ldots, a_n\} \in X$,

$$P(C|a_1, a_2, \ldots, a_n) = arg\ max_{a_i \in X} P(a_i|a_1, a_2, \ldots, a_n)$$

$$= arg\ max_{a_i \in X} \frac{P(a_1, a_2, \ldots, a_n|a_i)P(a_i)}{P(a_1, a_2, \ldots, a_n)} = arg\ max_{a_i \in X} P(a_1, a_2, \ldots, a_n|a_i)P(a_i)$$

Since the Naïve Bayes classifier assumes that the attribute values are conditionally independent given the class value C,

$$P(a_1, a_2, \ldots, a_n|a_i) = \prod_j P(a_j|a_i)$$

therefore

$$P(C|a_1, a_2, \ldots, a_n) = arg\ max_{a_i} P(a_i) \prod_j P(a_j|a_i)$$

Based on the frequencies of the variables over the training data, the estimation corresponds to the learned hypothesis, which is then used to classify a new instance into either buyer or non-buyer of certain product categories.

We use Weka's Naïve Bayes classifier implementation for our experiments [90]. We list the classification results in Table 7.8. In comparison to logistic regression as a classifier, for the same recall, Naïve Bayes has a lower precision.

Table 7.8: Naïve Bayes Classifier

	Naïve Bayes	Logistic Regression (P=0.035)
Precision	3.52%	9.15%
Recall	23.2%	24.1%

Discussions

From our experiments, we observed that logistic regression provides a lower precision than the Decision Tree, although it provides a flexible option to adjust the precision and recall for the classifiers. Naïve Bayes assumes the independence between each of the features. It is a simple classification model, although the precision is lower than logistic regression. The classification experimental results we have obtained on this user-centric clickstream data demonstrate effective product level prediction.

7.4 Conclusions

In this chapter, we demonstrate how the proposed techniques in this thesis can be used to enhance current knowledge discovery systems. The work on exploring the rule templates for choosing interesting rules in Chapter 3 can be applied to the system for limiting the number of recommendations and generating recommendations towards the special requirement of the applications. The Rule Importance Measures introduced in Chapter 4 are useful method for providing users various selections of recommendations in different degrees of importance and interestingness. The Rules-As-Attributes Measure discussed in Chapter 5 can be used to choose representative and important recommendations from large recommendation databases. The privacy concerns for users in the context of a knowledge discovery system are one of the common issues in the data collections of the real-world systems. In the situation when the users are not willing to release their personal information, the system will have many occurrences of missing values in the original data. Simply ignoring such user profiles having missing values is not a good strategy for utilizing the available information; therefore, techniques on how to handle large data sets containing missing at-

tributes values are very important. The proposed ItemRSFit approach in Chapter 6 are shown to be well applied to large data sets for processing missing values.

Through a case study of using Rule Importance Measures on generating important rules from a user-centric personalization system, we show empirically how the Rule Importance Measure, as a sample of proposed approaches in this thesis, can be adapted into a real world application. Important rules ranked by the Rule Importance Measure are effective on discovering potential online buyers. The other proposed approaches can also be applied to this user-centric personalization system. For example, the rule templates introduced in Chapter 3 can be used through the rule generation process to extract domain related rules. Given a set of rules generated by existing learning algorithms, the Rules-As-Attributes Measure in Chapter 5 can be applied for extracting important rules. In the situation when there exist missing attribute values from the user-centric clickstream data, we can use the ItemRSFit approach in Chapter 6 to assign the missing values in the data preprocessing stage. We also have interesting experimental results on discovering prominent features for user-centric personalization applications. The feature construction problem for user-centric applications is still a new area. Features that can better describe an online buyer or non-buyer's intention still needs to be studied. In addition, we have some preliminary results on a related topic in the area of online product purchasing prediction, namely how to make use of classification algorithms in order to make these predictions.

Chapter 8

Conclusion and Future Work

8.1 Conclusions

We proposed rule evaluation measures to facilitate the knowledge understanding process and data mining applications. We first studied rule templates as one of the subjective interestingness measures, and explained its application towards extracting interesting recommendations in a movie recommendation task (Chapter 3). We then introduced rough sets theory, based on which two rule evaluation measures were proposed, the Rule Importance Measure (Chapter 4), and the Rules-As-Attributes Measure (Chapter 5). The Rule Importance Measure was proposed to provide a ranking of how important the association rules are. Since reducts contain the representative attributes of the original data, rules generated based on reducts are therefore representative of the original knowledge. Multiple reducts generated multiple rules, and the more important rules were generated more frequently across these rule sets. The Rule Importance Measure can be applied to evaluate association rules. It is an easy and objective measure. The Rules-As-Attributes Measure is also designed to extract important rules, although it does not provide a list of rules ranked by their importance. This measure instead provides a set of important rules, which are from the reduct of a decision table that is constructed by considering the rules generated from original data as the condition attributes. Empirical studies for both these rough set based rule evaluation measures demonstrate their effectiveness on extracting important rules. These measures can be applied towards various applications.

In addition to the proposed rule evaluation measures, we also studied a data preprocessing approach on handling missing attribute values (Chapter 6). The ItemRSFit approach integrates both the association rule algorithms and the rough sets theory on assigning the missing attribute values. We believe such studies on the preprocessing of the knowledge discovery system can contribute to the rule generations, thus the rule evaluation process can benefit from more complete and better quality data. We propose this data preprocessing approach be done prior to applying either our proposed Rule Importance Measure or the Rules-As-Attributes Measure.

Finally, through a case study of an user-centric web personalization system, we show how the proposed techniques can be utilized and adapted to an actual system (Chapter 7). The Rule Importance Measures are demonstrated throughout the personalization system on how to extract important rules for purchasing predictions. The end results indicate the extracted important rules are useful to predict whether an online purchase will happen for certain users according to the observed online searching and browsing behaviours.

Our contribution lies in a series of rule evaluation measures and the explorations of their empirical applications. We summarize our contributions as follows:

- What's new? We have proposed two rough set based rule evaluation measures, the Rule Importance Measure and the Rules-As-Attributes Measure; a new use of rule templates to generate interesting rules, in the context of recommender systems; and a new approach ItemRSFit for preprocessing missing attribute values. Such rule evaluation techniques and missing attribute processing techniques proposed in this thesis are novel in the area of knowledge discovery in databases and data mining. We also explored the usage of the proposed new techniques on a new user-centric web personalization system.

- What's different? The Rule Importance Measures are different from the rule interestingness measure or the rule quality measures proposed by other researchers in the field [17, 35]. The Rule Importance Measure is used to evaluate association rules. It is an objective measure, which provides a more direct and obvious view for the rules. The Rule Importance Measure can reduce the amount of data required for the rule generations by selecting only important attributes from the original data. The number of rules generated is thus greatly reduced. As an example, for the geriatric

care data set in Chapter 4, 218 rules are generated and ranked using the Rule Importance Measure, however, $2,626,392$ rules are generated from the original data set without considering the reduct sets or rule templates. Our method efficiently extracts important rules, and at the same time provides a ranking for important rules.

- What's significant? The two rough set based rule evaluation measures, the Rule Importance Measure and the Rules-As-Attributes Measure, are our most significant contributions of this thesis. They provide an automatic, effective, and straightforward way of extracting important knowledge. They can also be used jointly with other measures to facilitate the rule evaluations. These two measures incorporate domain related information into the rule evaluations. The significance of the Rule Importance Measure is also reinforced in the case study presented in Chapter 7.

- What is better? The proposed techniques in this thesis can help people better understand the discovered knowledge from the original data. They help people automatically select important and significant knowledge from a huge amount of data. In case of incomplete knowledge, the proposed missing attribute processing approach can be used to provide a complete data source for knowledge discovery. Therefore rules generated from such complete data better represent the original knowledge than rules generated based on data with missing information.

8.2 Future Work

We plan to investigate the following future work.

- Rule Evaluation Extensions. The rule evaluation approaches discussed in Chapter 4 and 5 are proposed and demonstrated in experiments based on association rule generation. We believe such measures can be widely applied towards other rule generations such as classification rules and sequential patterns [4]. It would be interesting to conduct logic analysis on the Rule Importance Measure to make this measure more fundamental and useful. The Rule Importance Measure and the Rules-As-Attribute Measure can also be integrated together to further evaluate rules. For example, it

would be interesting to consider the Rules-As-Attribute measure as a transformation of rules into a decision table. In this decision table, we may also find multiple reducts. Some rules would appear more frequently in some reducts than others. The Rule Importance Measure can be again applied. We can rank rules according to the values of the Rule Importance Measure. This may provide further insight into the inherent merit of the Rule Importance Measure.

We would also like to extend the proposed measures towards broader knowledge discovery domains, such as survival analysis [11] in the medical research field. Patients' survival status and survival time (such as days of survival after a disease is diagnosed) are the two main objectives for predictions. Such predictions also require rule generations based on the medical data. The rule evaluation measures discussed in this thesis can be applied to evaluate such prediction rules to facilitate the doctors' diagnosis. In addition to extending the proposed evaluations to more application domains, we are also interested in exploring their values in a general rule evaluation framework. Yao proposed a three-level framework for the theoretical foundations of measuring and quantifying discovered knowledge based on utility theory [93, 94]. We would like to explore the value of our proposed rule evaluation measures in this framework and to compare with other rule evaluation measures within that framework.

- Cost-Sensitive Learning. The ItemRSFit approach proposed in Chapter 6 uses the RSFit approach to predict the non-compatible records. We would like to experiment with other techniques on predicting missing values for the non-compatible records, such as the method of assigning the common attribute values, to improve the overall accuracies of the integrated ItemRSFit prediction. In our research, we also adopt the strategies used by Zhu and Wu [100] on balancing the computational cost and the prediction accuracy. A lower support value can bring a higher prediction accuracy; however, frequent itemsets with lower support require more time for computation than frequent itemsets with higher support. In the future, we are interested in exploring a satisfactory balance between the support value and the prediction accuracy, in order to obtain a satisfactory accuracy efficiently. Given the available computational cost and the affordable computation time, it is interesting to explore what percentage of the missing attributes can be predicted, and what are the most effec-

tive attributes to be predicted. In case of a higher prediction cost, the idea of giving more important attributes higher priorities for predictions may be applied as heuristics. Consider Table 7.1 in Chapter 7 as an example. Through the case study, we found feature G14a (i.e., whether the user made a purchase in the previous month) is a more important attribute than G14c (i.e., the number of purchases the user made in the previous month) for predicting an online buyer of certain product. In the case of missing attribute values existing in both these two features, we can predict the missing value of G14a prior to G14c to see whether the overall learning task is satisfactory.

- New Challenges Facing the User-Centric Personalization System. In this thesis, most of the empirical experiments conducted in multiple chapters are towards the applications of personalization systems. In the case study of the user-centric web personalization systems (Chapter 7), we experienced several challenges through our experiment. For future work, we would like to expand on this experimentation in its own right, towards improved personalization for users. First, extracting a terabyte of data from a database is a long process. Secondly, classifying imbalanced data is a challenging process. Since most people are not online buyers, in our data set, the majority class belongs to non-buyers, and a very small percentage are buyers. Out of $83,635$ number of users, the two classes of buyers and non-buyers are divided as 449 vs. $83,186$ users. Without a controlling method (such as forcing the decision tree to branch), the decision tree classifies all the users as non-buyers. We would like to use the techniques [81, 83] from recent research on classifying imbalanced data to help solve the classification difficulties. Thirdly, feature constructions require a mix of domain knowledge and a data miner's expertise. Features that can better describe an online buyer or non-buyer's intentions still need to be studied and brought into the experiment. Fourthly, the search terms used by different websites have different indications, which made the extraction task difficult. Fifthly, the standard evaluation for the user-centric personalization task is still yet to be discovered. As an emerging research area, we are interested in exploring different problems in this area such as developing richer user models, investigating techniques for predicting approximate purchasing time for user online purchases and exploring latent models for user in-

tentions and predicting demographic information based on users' online searching behaviours.

- Evaluations with Human Users. For the rule evaluation measures we proposed in Chapter 4 and 5, we compared the differences between the Rule Importance Measures and the confidence measures as an example of current interestingness measures. We also introduced some intuitive evaluations of "more interesting" rules. In the future, we would like to study how effective these measures are by performing experiments with human users who are experts in the domain. Certain user satisfaction studies may be conducted for real people's evaluations with appropriate measures from across a sufficiently large sample of users in a restricted domain.

Appendix A

Other Related Concepts in Rough Sets Theory

We list a walk-through example to explain other related concepts in rough sets theory that might be of interest. The following discussions are based on [72].

A data set can be represented as a decision table, which is used to specify what conditions lead to decisions. A decision table can be defined as $T = (U, C, D)$, where U is the set of objects in the table, C is the set of the condition attributes and D is the set of the decision attributes. Table A.1 gives an example of the decision table. {a,b,c} is the set of condition attributes, and {d} is the set of decision attributes.

Here we only look at the situation when the value of the decision attributes is either 0, or 1. And we will not discuss the situation when the condition attributes have missing values.

U is the set of objects we are interested in, where $U \neq \phi$. Let R be an equivalence relation over U, then the family of all equivalence classes of R is represented by U/R. $[x]_R$ means a category in R containing an element $x \in U$. Suppose $P \subseteq R$, and $P \neq \phi$, $IND(P)$ is an equivalence relation over U. For any $x \in U$, the equivalence class of x of the relation $IND(P)$ is denoted as $[x]_P$. X is a subset of U, R is an equivalence relation, the lower approximation of X and the upper approximation of X is defined as:

$$\underline{R}X = \cup\{x \in U | [x]_R \subseteq X\}$$

Table A.1: An Example of Decision Table

U	a	b	c	d
1	1	1	0	0
2	1	2	0	1
3	1	3	0	1
4	0	1	0	0
5	0	2	0	0
6	0	3	0	1
7	0	2	0	1
8	0	3	0	0

$$\overline{R}X = \cup\{x \in U | [x]_R \cap X \neq \phi\}$$

respectively.

Reduct and core are further defined as follows [72]. R is an equivalence relation and let $S \in R$. We say, S is dispensable in R, if $IND(R) = IND(R - \{S\})$; S is indispensable in R if $IND(R) \neq IND(R - \{S\})$. We say R is independent if each $S \in R$ is indispensable in R.

Q is a reduct of P if Q is independent, $Q \subseteq P$, and $IND(Q) = IND(P)$. An equivalence relation over a knowledge base can have many reducts. The intersection of all the reducts of an equivalence relation P is defined to be the *Core*, where

$$Core(P) = \cap \text{All Reducts of P.}$$

The reduct and the core are important concepts in rough sets theory. Reduct sets contain all the representative attributes from the original data set. They are often used in attribute selection process. The core is contained in all the reduct sets, and it is the necessity of the whole data. Any reduct generated from the original data set cannot exclude the core attributes.

Let $T = (U, C, D)$ be a decision table, the C-positive region of D is defined to be the

set of all objects of U which can be classified into U/D using attributes from C, which is,

$$POS_C(D) = \cup\{\underline{C}X | X \in IND(D)\}.$$

An attribute $f \in C$ is dispensable if $POS_{C-\{f\}}(D) = POS_C(D)$. All the core attributes are indispensable.

The degree of dependency between the equivalent class R and the decision attribute D is defined as

$$\tau_R(D) = \frac{\text{cardinality of } POS_R(D)}{\text{cardinality of } U}.$$

We use Table A.1 as an example to show how to calculate the degree of dependency.

Example 8 In Table A.1, $U = \{1, 2, 3, ..., 8\}$ is a set of objects. $C = \{a, b, c\}$, $D = \{d\}$. Suppose $IND = \{b, c\}$. We have the equivalence classes of IND, $E_1 = \{1, 4\}$, $E_2 = \{2, 5, 7\}$, $E_3 = \{3, 6, 8\}$. The decision attribute d consists of two classes, $D_1 = \{2, 3, 6, 7\}$, $D_0 = \{1, 4, 5, 8\}$. The lower and upper approximation of D are,

$$\begin{aligned}
\underline{R}D_1 &= \phi \\
\overline{R}D_1 &= E_2 \cup E_3 = \{2, 3, 5, 6, 7, 8\} \\
\underline{R}D_0 &= E_1 = \{1, 4\} \\
\overline{R}D_0 &= E_1 \cup E_2 \cup E_3 = \{1, 2, 3, 4, 5, 6, 7, 8\}
\end{aligned}$$

Because $IND(\{b, c\}) = IND(\{b, c\} - \{c\})$, we say c is dispensable. For $P = \{a, b, c, d\}$, $Q \subseteq P$, $Q = \{a, b\}$. Because $IND(Q) = IND(P)$, $Q = \{a, b\}$ is a reduct of P.

$IND(D) = \{\{2, 3, 6, 7\}, \{1, 4, 5, 8\}\}$, $IND(\{b, c\}) = \{\{1, 4\}, \{2, 5, 7\}, \{3, 6, 8\}\}$, therefore $POS_{\{b,c\}}(D) = \{1, 4\}$.

Because $POS_{\{b,c\}-\{b\}}(D) = \phi \neq POS_{\{b,c\}}(D)$, b is indispensable.

$$\tau_{\{b,c\}}(D) = \frac{\text{cardinality of } POS_{\{b,c\}}(D)}{\text{cardinality of } U} = \frac{2}{4} = \frac{1}{2}.$$

This dependency evaluation is often used as the stopping condition for the reduct generation algorithm.

Appendix B

Geriatric Care Data Set

We list the geriatric care data set used in Chapter 4, Chapter 5, and Chapter 6 in Table B.1.

Table B.1: Attributes for the Geriatric Care Data Set

Order	Name	Question
1	edulevel	Education level
2	eyesight	How is your eyesight?
3	hearing	How is your hearing?
4	eat	Can you eat?
5	dress	Can you dress and undress yourself?
6	takecare	Can you take care of your appearance?
7	walk	Can you walk?
8	getbed	Can you get in and out of bed?
9	shower	Can you take a bath or shower?
10	bathroom	Can you go to the bathroom commode?

11	phoneuse	Can you use the telephone?
12	walkout	Can you get places out of walking dist.?
13	shopping	Can you go shopping for groceries etc.?
14	meal	Can you prepare your own meals?
15	housewk	Can you do your housework?
16	takemed	Can you take your own medicine?
17	money	Can you handle your own money?
18	health	How is your health these days?
19	trouble	Trouble with life?
20	livealone	Do you live here alone?
21	cough	Often cough?
22	tired	Easy feel tired?
23	sneeze	Often sneeze?
24	hbp	High blood pressure?
25	heart	Heart problem?
26	stroke	Stroke or effects of stroke?
27	arthriti	Arthritis or rheumatism?
28	parkinso	Parkinson's disease?
29	eyetroub	Eye trouble not relieved by glasses?
30	eartroub	Ear trouble?
31	dental	Dental Problems?
32	chest	Chest problems?
33	stomach	Stomach or digestive problems?
34	kidney	Kidney Problems?
35	bladder	Lose control of your bladder?
36	bowels	Lose control of you bowels?
37	diabetes	Ever been diagnosed with diabetes?
38	feet	Feet problems?
39	nerves	Nerve problems?
40	skin	Skin problem?
41	fracture	Any fractures?
42	age6	Age group by 5-year
43	studyage	Age at investigation
44	sex	Sex
45	livedead	Survival status

Appendix C

Data Sets Used in Chapter 4 and Chapter 5

We list selected UCI data sets [21] A through M that are used in Chapter 4 and Chapter 5 as follows.

A. Abalone Data This data set is used to predict the age of abalone from physical measurements. There are $4,177$ instances and 8 condition attributes in this data set. There are no missing attribute values or inconsistent data instances in the data set.

B. Breast Cancer Data This data set contains 9 condition attributes and 286 instances. The date is used to diagnose the breast cancer disease. There are missing attributes existing in the data set. We ignore all the missing attribute values, and remove 9 records, we have 277 instances in the data. There are 12 inconsistent data records removed from the data as well.

C. Car Data The car data set contains 6 condition attributes, and $1,728$ instances. We apply association rules algorithm with rule templates, and there are 9 rules generated. We first use core algorithm to generate core attributes, and all the condition attributes are the core attributes. There is only one reduct generated for this data set, and the reduct contains all the core attributes.

D. Glass Data This data set is used for the study of classification of types of glass by criminological investigation. At the scene of the crime, the glass left can be used as evidence. There are 214 instances and 9 condition attributes. There are no missing attribute values or inconsistent data instances.

E. Heart Data This data set is related to heart disease diagnosis. There are 4 databases in this data set, we use cleveland clinic foundation data in our experiment because this is the only one well processed and used by most researchers. This cleveland data contains 303 instances, and 13 condition attributes. We remove 6 missing attribute values. There is no inconsistent data existing.

F. Iris Data This data set concerns plants. For the Iris data set, there are 4 condition attributes, 150 instances. There is no inconsistent data existing in the data. We first use core algorithm to generate core attributes, but the result is empty. This means none of the attributes is indispensable. There are 4 reducts generated. We apply association rules algorithm with rule templates, and there are 50 rules generated.

G. Lymphography Data The data set contains 148 instances and 18 condition attributes. There are no missing attribute values in this data. We check that there is no inconsistent data. The core is empty for this data set. 147 reducts are generated from this data set.

H. Pendigits Data This is a pen-based recognition of handwritten digits data set. There are 10 classes with 16 condition attributes in the data, and $7,494$ training instances and $3,498$ testing instances are in the data. We use training data to conduct our experiments. Each instance represents a hand-written digit with 16 attributes, which are coordinates information. There is no reference on the 16 condition attributes. We use Ci $(1 \leq i \leq 16)$ to represent these attributes in our experiments. There are no missing attribute values, or inconsistent data in this data.

I. Pima Indians Diabetes Data The data comes from all female patients who are at least 21 years old of the pima Indian heritage. The data is used to diagnose whether

patients show signs of diabetes according to a list of criteria. There are 768 instances and 8 condition attributes in this data set. There are no missing attribute values, and no inconsistent data.

J. Spambase Data This data set originally contains $4,601$ instances and 57 condition attributes. It is used to classify spam and non-spam emails. Most of the attributes indicate whether a certain word (such as, order, report) or character (such as !, #) appears frequently in the emails. There are no missing attribute values. There are 6 inconsistent data instances that are removed. After removing redundant data instances as well, there are $4,204$ left in this data set. There are 110 reducts and 7 core attributes generated from this data set. It is interesting to notice that, the core attributes, which are essential to determine whether an email is not a spam email, are, the word frequency of "george", "meeting", 're", "you", "edu", "!", and the total number of capital letters in the email. In addition, it is interesting to pay attention to the reducts as well. They are important information on identifying the possible spam emails.[1]

K. Wine Recognition Data This data is about using chemical analysis to determine the origin of wines. There are 13 attributes, 178 instances, and 3 classes in the data. There are no missing attribute values or inconsistent data. The core is empty.

L. Yeast Data This data set is used to predict the cellular localization sites of proteins. There are $1,484$ instances with 8 condition attributes in the data, and no missing attribute values. We remove 31 redundant instances.

M. Zoo Data This artificial data set contains 7 classes of animals, 17 condition attributes, 101 data instances, and there are no missing attribute values in this data set. Since the first condition attribute "animal name" is unique for each instance, and we consider each instance a unique itemset, we do not consider this attribute in our experiment. There are no inconsistent data in this data set.

[1]For the apriori association rule generation, we set the maximum number of item per set to be 6. Without this limitation, the rule generation gives an error of "out of memory".

Bibliography

[1] Eachmovie collaborative filtering data set, 1997.

[2] Sun microsystems: Sun fire v880 specifications. sun microsystems products and services, 2003.

[3] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

[4] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *ICDE '95: Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society.

[5] Alan Agresti. *An introduction to Categorical Data Analysis*. John Wiley & Sons, 1996.

[6] Aijun An and Nick Cercone. Elem2: A learning system for more accurate classifications. In *Canadian Conference on AI*, pages 426–441, 1998.

[7] Aijun An and Nick Cercone. Rule quality measures for rule induction systems: Description and evaluation. *Computational Intelligence*, 17(3):409–424, 2001.

[8] Liliana Ardissono, Anna Goy, Giovanna Petrone, and Marino Segnan. A multi-agent infrastructure for developing personalized web-based systems. *ACM Trans. Inter. Tech.*, 5(1):47–69, 2005.

[9] M. Balabanovic and Y. Shoham. Fab: Content-based, collaborative recommendation. *Communication of the ACM, 40(3)*, pages 66–72, 1997.

[10] Jan G. Bazan, Hung Son Nguyen, Sinh Hoa Nguyen, Piotr Synak, and Jakub Wróblewski. Rough set algorithms in classification problem. pages 49–88, 2000.

[11] Jan G. Bazan, Antoni Osmólski, Andrzej Skowron, Dominik Slezak, Marcin S. Szczuka, and Jakub Wroblewski. Rough set approach to the survival analysis. In *RSCTC '02: Proceedings of the Third International Conference on Rough Sets and Current Trends in Computing*, pages 522–529, London, UK, 2002. Springer-Verlag.

[12] Jan G. Bazan, Marcin S. Szczuka, and Jakub Wróblewski. A new version of rough set exploration system. In *RSCTC '02: Proceedings of the Third International Conference on Rough Sets and Current Trends in Computing*, pages 397–404, London, UK, 2002. Springer-Verlag.

[13] Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proc. 15th International Conf. on Machine Learning*, pages 46–54. Morgan Kaufmann, San Francisco, CA, 1998.

[14] Daniel Billsus and Michael J. Pazzani. A hybrid user model for news story classification. In *UM '99: Proceedings of the seventh international conference on User modeling*, pages 99–108, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc.

[15] C. Borgelt. Efficient implementations of apriori and eclat. In *Proceedings of the FIMI'03 Workshop on Frequent Itemset Mining Implementations, Melbourne, Florida, USA, November, CEUR Workshop Proceedings*, 2003.

[16] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. In Joan Peckham, editor, *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*, pages 265–276. ACM Press, 1997.

[17] Ivan Bruha. Quality of decision rules: Definitions and classification schemes for multiple rules. In *Machine Learning and Statistics, The Interface, Edited by G. Nakhaeizadeh and C. C. Taylor.*, pages 107 – 131. John Wiley & Sons, Inc., 1997.

[18] David K. Y. Chiu, Andrew K. C. Wong, and B. Cheung. Information discovery through hierarchical maximum entropy discretization and synthesis. In *Knowledge Discovery in Databases*, pages 125–140. 1991.

[19] A. Chouchoulas and Q. Shen. Rough set-aided keyword reduction for text categorization. *Applied Artificial Intelligence*, 15:843–873, 2001.

[20] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 233–240, New York, NY, USA, 2006. ACM Press.

[21] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[22] Guozhu Dong and Jinyan Li. Interestingness of discovered association rules in terms of neighborhood-based unexpectedness. In R. Kotagiri X. Wu and K. Korb, editors, *Proceedings of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'98)*, pages 72–86, Melbourne, Australia, April 1998.

[23] Piatesky-Shapiro G. Fayyad, U. and P. Smyth. From data mining to knowledge discovery: an overview. *Advances in Knowledge discovery and data mining (AAAI/MIT Press)*, pages 1–34, 1996.

[24] Rachel L. Freeman, Jerzy W. Grzymala-Busse, Laura A. Riffel, and Stephen R. Schroeder. Analyzing the relation between heart rate, problem behavior, and environmental events using data mining system lers. In *CBMS '01: Proceedings of the Fourteenth IEEE Symposium on Computer-Based Medical Systems*, page 11, Washington, DC, USA, 2001. IEEE Computer Society.

[25] B. Gray and M.E. Orlowska. Ccaiia: Clustering categorical attributes into interesting association rules. In *Proceedings of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'98)*, pages 132–143, Melbourne, Australia, April 1998.

[26] Jerzy W. Grzymala-Busse. Incomplete data and generalization of indiscernibility relation, definability, and approximations. In Dominik Slezak, Guoyin Wang, Marcin S. Szczuka, Ivo Düntsch, and Yiyu Yao, editors, *RSFDGrC (1)*, volume 3641 of *Lecture Notes in Computer Science*, pages 244–253. Springer, 2005.

[27] Jerzy W. Grzymala-Busse, Witold J. Grzymala-Busse, and Linda K. Goodwin. Coping with missing attribute values based on closest fit in preterm birth data: A rough set approach. *Computational Intelligence*, 17(3):425–434, 2001.

[28] Jerzy W. Grzymala-Busse and Ming Hu. A comparison of several approaches to missing attribute values in data mining. In *Rough Sets and Current Trends in Computing*, pages 378–385, 2000.

[29] L. Feng H. Liu, H. Lu and F. Hussain. Efficient search of reliable exceptions. In N. Zhong and L. Zhou, editors, *Proceedings of the Third Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'99)*, pages 194–203, Beijing, China, April 1999.

[30] Jiawei Han, Yandong Cai, and Nick Cercone. Knowledge discovery in databases: An attribute-oriented approach. In Li-Yan Yuan, editor, *Proceedings of the 18th International Conference on Very Large Databases*, pages 547–559, San Francisco, U.S.A., 1992. Morgan Kaufmann Publishers.

[31] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2000.

[32] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In Weidong Chen, Jeffrey Naughton, and Philip A. Bernstein, editors, *2000 ACM SIGMOD Intl. Conference on Management of Data*, pages 1–12. ACM Press, May 2000.

[33] Aboul-Ella Hassanien. Rough set approach for attribute reduction and rule generation: a case of patients with suspected breast cancer. *J. Am. Soc. Inf. Sci. Technol.*, 55(11):954–962, 2004.

[34] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[35] R. Hilderman and H. Hamilton. Knowledge discovery and interestingness measures: A survey. Technical report, Department of Computer Science, University of Regina, October 1999.

[36] R.J. Hilderman. *Mining Summaries From Databases Using Domain Generalization Graphs and Objective Measures of Interestingness.* PhD thesis, University of Regina, 2000.

[37] Thomas Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 259–266, New York, NY, USA, 2003. ACM Press.

[38] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 688–693, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[39] Xiaohua Hu. *Knowledge Discovery in Databases: an Attribute-Oriented Rough Set Approach.* PhD thesis, University of Regina, 1995.

[40] Xiaohua Hu, T. Y. Lin, and Jianchao Han. A new rough sets model based on database systems. *Fundam. Inf.*, 59(2-3):135–152, 2004.

[41] Xiaohua Hu, Ning Shan, Nick Cercone, and Wojciech Ziarko. Dbrough: A rough set based knowledge discovery system. In *ISMIS '94: Proceedings of the 8th International Symposium on Methodologies for Intelligent Systems*, pages 386–395, London, UK, 1994. Springer-Verlag.

[42] Jin Huang, Jingjing Lu, and Charles X. Ling. Comparing naive bayes, decision trees, and svm with auc and accuracy. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 553, Washington, DC, USA, 2003. IEEE Computer Society.

[43] Zhe Huang and Yun-Quan Hu. Applying AI technology and rough set theory to mine association rules for supporting knowledge management. In *International Conference on Machine Learning and Cybernetics*, volume 3, pages 1820–1825, 2003.

[44] D. Ivo and G. Gunther. The rough set engine grobian. In *Proceedings of the 15th IMACS World Congress*, volume 4, Berlin, 1997.

[45] Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo. Finding interesting rules from large sets of discovered association rules. In Nabil R. Adam, Bharat K. Bhargava, and Yelena Yesha, editors, *Third International Conference on Information and Knowledge Management (CIKM'94)*, pages 401–407. ACM Press, 1994.

[46] B. Krulwich and C. Burkey. Learning user information interests through extraction of semantically significant phrases. *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access, Stanford, California. March*, 1996.

[47] Marzena Kryszkiewicz. Association rules in incomplete databases. In *PAKDD*, volume 1574 of *Lecture Notes in Computer Science*, pages 84–93. Springer, 1999.

[48] Marzena Kryszkiewicz and Henryk Rybinski. Finding reducts in composed information systems. In *RSKD*, pages 261–273, 1993.

[49] K. Lang. Newsweeder: Learning to filter netnews. *Proceedings of the 12nd International Conference on Machine Learning, Tahoe City, California*, 1995.

[50] Jiuyong Li, Rodney Topor, and Hong Shen. Construct robust rule sets for classification. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 564–569, New York, NY, USA, 2002. ACM Press.

[51] Jiye Li and Nick Cercone. Applying association rules for interesting recommendations using rule templates. Technical Report CS-2004-0415, School of Computer Science, University of Waterloo, April 2004.

[52] Jiye Li and Nick Cercone. Discovering and ranking important rules. In *IEEE Granular Computing*, volume 2, pages 506–511, Beijing, China, July 2005.

[53] Jiye Li and Nick Cercone. Empirical analysis on the geriatric care data set using rough sets theory. Technical report, School of Computer Science, University of Waterloo, March 2005.

[54] Jiye Li and Nick Cercone. A rough set based model to rank the importance of association rules. In Dominik Slezak, Jingtao Yao, James F. Peters, Wojciech Ziarko, and Xiaohua Hu, editors, *RSFDGrC (2)*, volume 3642 of *Lecture Notes in Computer Science*, pages 109–118. Springer, 2005.

[55] Jiye Li and Nick Cercone. Assigning missing attribute values based on rough sets theory. In *IEEE Granular Computing*, pages 607–610. IEEE Computer Society, 2006.

[56] Jiye Li and Nick Cercone. Comparisons on different approaches to assign missing attribute values. Technical Report CS-2006-04, School of Computer Science, University of Waterloo, January 2006.

[57] Jiye Li and Nick Cercone. Introducing a rule importance measure. In James F. Peters, Andrzej Skowron, Didier Dubois, Jerzy W. Grzymala-Busse, Masahiro Inuiguchi, and Lech Polkowski, editors, *T. Rough Sets*, volume 4100 of *Lecture Notes in Computer Science*, pages 167–189. Springer, 2006.

[58] Jiye Li and Nick Cercone. Predicting missing attribute values based on frequent itemset and rsfit. Technical Report CS-2006-13, School of Computer Science, University of Waterloo, April 2006.

[59] Jiye Li and Nick Cercone. A method of discovering important rules using rules as attributes. *International Journal of Intelligent System, Special Issues on Granular Computing*, 2007.

[60] Jiye Li, Puntip Pattaraintakorn, and Nick Cercone. Rule evaluations, attributes, and rough sets: Extension and a case study. In *Transactions on Rough Sets VI, the First Commemorative Issue*, volume 4374 of *Lecture Notes in Computer Science*. Springer, 2007.

[61] Jiye Li, Bin Tang, and Nick Cercone. Applying association rules for interesting recommendations using rule templates. In *Proceedings of the Eighth Pacific-Asia Conference, PAKDD2004*, pages 166–170, 2004.

[62] Henry Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, Montreal, Quebec, Canada, 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[63] Weiyang Lin, Sergio A. Alvarez, and Carolina Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data Min. Knowl. Discov.*, 6(1):83–105, 2002.

[64] M. E. Maron. Automatic indexing: An experimental inquiry. *J. ACM*, 8(3):404–417, 1961.

[65] P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering. In *Proceedings of the SIGIR-2001 Workshop on Recommender Systems.*, 9 2001.

[66] Bradley N. Miller, Istvan Albert, Shyong K. Lam, Joseph A. Konstan, and John Riedl. Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of ACM 2003 Conference on Intelligent User Interfaces (IUI'03)* , Chapel Hill, North Carolina, 2003. ACM.

[67] Y.Y. Yao N. Zhong and S. Ohsuga. Peculiarity-oriented multi-database mining. In J. Zytkow and J. Rauch, editors, *Proceedings of the Third European Conference on the Principles of Data Mining and Knowledge Discovery (PKDD'99)*, pages 136–146, Prague, Czech Republic, September 1999.

[68] Aleksander Øhrn. Rosetta technical reference manual. Department ofComputer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway. May 25, 2001.

[69] Aleksander Øhrn. *Discernibility and Rough Sets in Medicine: Tools and Applications.* PhD thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim Norway, 1999.

[70] Balaji Padmanabhan, Zhiqiang Zheng, and Steven O. Kimbrough. Personalization from incomplete data: what you don't know can hurt. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 154–163, New York, NY, USA, 2001. ACM Press.

[71] Puntip Pattaraintakorn, Nick Cercone, and Kanlaya Naruedomkul. Hybrid intelligent systems: Selecting attributes for soft-computing analysis. In *COMPSAC (1)*, pages 319–325. IEEE Computer Society, 2005.

[72] Zdzislaw Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[73] Zdzislaw Pawlak, Jerzy Grzymala-Busse, Roman Slowinski, and Wojciech Ziarko. Rough sets. *Commun. ACM*, 38(11):88–95, 1995.

[74] Jian Pei, Jiawei Han, and Runying Mao. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[75] Bartlomiej Predki and Szymon Wilk. Rough set based data exploration using rose system. In *ISMIS '99: Proceedings of the 11th International Symposium on Foundations of Intelligent Systems*, pages 172–180, London, UK, 1999. Springer-Verlag.

[76] Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

[77] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[78] V. J. Rayward-Smith, J. C. W. Debuse, and B. de la Iglesia. The use of modern heuristic algorithms for mining insurance data. In *Handbook of data mining and knowledge discovery*, pages 849–856, New York, NY, USA, 2002. Oxford University Press, Inc.

[79] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM*

1994 Conference on Computer Supported Cooperative Work, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[80] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth. *Conference on Human Factors in Computing Systems-CHI'95, Denver*, 1995.

[81] Victor S. Sheng and Charles X. Ling. Thresholding for making classifiers cost-sensitive. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, Boston, Massachusetts, USA, 07 2006.

[82] Padhraic Smyth and Rodney M. Goodman. Rule induction using information theory. In *Knowledge Discovery in Databases*, pages 159–176. 1991.

[83] Yanmin Sun, Mohamed S. Kamel, and Yang Wang. Boosting for learning multiple classes with imbalanced class distribution. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006)*, Hong Kong, China, 2006.

[84] Marcin S. Szczuka. Rules as attributes in classifier construction. In Ning Zhong, Andrzej Skowron, and Setsuo Ohsuga, editors, *RSFDGrC*, volume 1711 of *Lecture Notes in Computer Science*, pages 492–499. Springer, 1999.

[85] Pang-Ning Tan and Vipin Kumar. Interestingness measures for association patterns: A perspective. *KDD 2000 Workshop on Postprocessing in Machine Learning and Data Mining, Boston, MA.*, 2000.

[86] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *KDD*, pages 32–41. ACM, 2002.

[87] S. Vinterbo and A. Øhrn. Minimal approximate hitting sets and rule templates. *International Journal of Approximate Reasoning*, 25(2):123–143, 2000.

[88] Staal A. Vinterbo and Aleksander Øhrn. Minimal approximate hitting sets and rule templates. *Int. J. Approx. Reasoning*, 25(2):123–143, 2000.

[89] T. Wakaki, H. Itakura, and M. Tamura. Rough set-aided feature selection for automatic web-page classification. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, pages 70–76, 2004.

[90] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann Publishers Inc., 2005.

[91] Chih-Hung Wu, Chian-Huei Wun, and Hung-Ju Chou. Using association rules for completing missing data. In *Fourth International Conference on Hybrid Intelligent Systems*, pages 236–241. IEEE Computer Society, 2004.

[92] K. Kobayashi Y. Yoshida, Y. Ohta and N. Yugami. Mining interesting patterns using estimated frequencies from subpatterns and superpatterns. In G. Grieser et al., editor, *DS 2003*, pages 494–501. Springer-Verlag Berlin Heidelberg, 2003.

[93] Y.Y. Yao. A step towards the foundations of data mining. In *Data Mining and Knowledge Discovery: Theory, Tools, and Technology V, The International Society for Optical Engineering*, pages 254–263, 2003.

[94] Y.Y. Yao, Y.H. Chen, and X.D. Yang. A measurement-theoretic foundations of rule interestingness evaluation. In *Foundations and Novel Approaches in Data Mining Series*, pages 41–59. Springer-Verlag, Berlin, 2006.

[95] Y.Y. Yao, Y. Zhao, and J. Wang. On reduct construction algorithms. In *Proceedings of RSKT'06*, volume 4062 of *Lecture Notes in Artificial Intelligence*, pages 297–304. Springer, 2006.

[96] M. Zaki and C. Hsian. Charm: an efficient algorithm for closed association rule mining. *Technical Report, RPI, Troy NY*, 1999.

[97] Chun Zeng, Chun-Xiao Xing, and Li-Zhu Zhou. Similarity measure and instance selection for collaborative filtering. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 652–658, New York, NY, USA, 2003. ACM Press.

[98] Tingshao Zhu. *Goal-Directed Complete-Web Recommendation.* PhD thesis, University of Alberta, 2006.

[99] Tingshao Zhu, Russell Greiner, and Gerald Häubl. Learning a model of a web user's interests. In Peter Brusilovsky, Albert T. Corbett, and Fiorella de Rosis, editors, *User Modeling*, volume 2702 of *Lecture Notes in Computer Science*, pages 65–75. Springer, 2003.

[100] Xingquan Zhu and Xindong Wu. Cost-constrained data acquisition for intelligent data preparation. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1542–1556, 2005.