Discovering Interesting Interrelationships with Undiscretized Quantitative Attributes in Large, Dense Databases

by

Shiying Huang

Thesis

Submitted by Shiying Huang

for fulfillment of the Requirements for the Degree of

Master of Information Technology (Research) (1895)

Supervisor: Professor Geoffrey I. Webb

School of Computer Science and Software Engineering
Monash University

July, 2005

To my parents, for your selfless love.
For the loving memory,
believe me, we can work it out!

Contents

List of Tables

List of Figures

Discovering Interesting Interrelationships with Undiscretized Quantitative Attributes in Large, Dense Databases

Shiying Huang
Shiyingh@csse.monash.edu.au
Monash University, 2005

Supervisor: Professor Geoffrey I. Webb

Abstract

Exploratory rule discovery is widely employed in real-world data mining, because of the flexibility in selecting applicable models. Nevertheless, two problems coexist with the merits of exploratory rule discovery. One of these drawbacks is how to limit within reasonable bounds the number of resulting models. The other problem is how to improve the efficiency of rule discovery by eliminating unnecessary computation and I/O. Techniques for tackling these issues have been studied extensively in the context of exploratory rule discovery with qualitative attributes. However, databases processed often involve quantitative attributes. Some researchers strive to introduce quantitative attributes into exploratory rule discovery by discretization, with which information loss is unavoidable. Such techniques are not optimal for mining inter-relationships between quantitative attributes and qualitative attributes. A special class of exploratory rule discovery has been proposed for mining rules with consequents being one or more undiscretized target quantitative variables. Characteristics of the selected quantitative variables are described using distributional statistics. However, previous techniques for mining exploratory rules with undiscretized quantitative targets cannot efficiently search for rules in very large, dense databases. Rule pruning techniques in this context are also limited.

The only investigation was the pruning of *insignificant quantitative association rules* proposed by Aumann and Lindell (1999). Efficiency is one of the critical issues for such techniques.

Accordingly, we propose techniques for pruning rules with undiscretized quantitative attributes. We call these techniques the *derivative extended rule filter* and the *derivative partial rule filter*. The derivative extended rule filter is an efficient variant of the existing *insignificant quantitative association rule* pruning proposed by Aumann and Lindell (1999). The derivative partial rule filter is able to remove potentially uninteresting rules that remain after the derivative extended rule filter is applied. We also discovered severe efficiency problems in existing rule pruning techniques with undiscretized quantitative attributes. The *triviality filter* is then suggested as a complement for the derivative extended rule filter, whose antimonotonicity can be utilized for more powerful search space pruning. We also propose the *difference set statistics derivation* and the *circular intersection* approaches for lessening the redundancies of data accesses and computation in our original implementation of derivative rule filters. Detailed experimental evaluations are committed to back up our arguments for desirable performance expectations with the above techniques.

Discovering Interesting Interrelationships with Undiscretized Quantitative Attributes in Large, Dense Databases

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Shiying Huang
July 13, 2005

Acknowledgments

The work leading up to this thesis was done during my one and a half years as master's candidate in the *Faculty of information Technology, Monash University.* The candidature has been very stimulating and instructive, during which I have been trained to do preliminary research. I am lucky to have so many understanding people about. The atmosphere in our group has given rise to several interesting discussions and ideas. Without the help of all these people I wouldn't be able to finish this thesis.

First of all, I want to thank my supervisor Geoffrey I. Webb for his kind supervision. I am so grateful for his support and contribution to the research presented in this thesis. He's been acting both as a teacher and a friend for me. He offered me every possible opportunity to improve myself. I am so grateful for him, because he is always smiling and understanding, even when I disappointed him.

Second, I would like to thank my parents and my brother for their selfless love and tolerance. They share both my sadness and happiness. They care about me with all their hearts, although they are thousands of miles away from me. They never walk away when I am in need of assist, even when I upset them with self-indulgence. To another person, Ning, who is as important as my family in my life, I would also like to say "thank you". He showed me the essence of love. I thus learn to peep into my own heart to see who I am, who I want to be, and who I should be. He always encourages me to enjoy the song of beatles, *We Can Work It Out*, when I am stray. This is what makes me hold on and struggle forward in difficulties. Sometimes, it hurts so much, yet I still believe he is doing everything for my wellbeing.

The following thanks go to my colleagues, Fei Zheng, for she always give me a bit of her mind; Xiaoya Lin, for his kind help in my life in Melbourne; Shane Butler and Janice Boughton, for their kind support when I got stuck in technical problems; and of course Michelle Kinsman, she is always willing to tell me how to get everything done. Thanks also go to my friends who back me up all through my way, even though they are far away from me. Especially to Yuguang Du, Ruilin Wu, Fei Wang and Yang Peng for their patience when I need somebody to confide.

It is impossible to express my gratitude for everyone. I can only deliver a "thank you" to all who care about me and whom I love.

Shiying Huang

Monash University

July 2005

Chapter 1

Introduction

This thesis focuses on two critical issues that affect exploratory discovery of rules with undiscretized quantitative consequents. The first is how to identify and discard important classes of potentially uninteresting rules. The second is how to support efficient search for interesting rules. *Exploratory rule discovery* searches for implicit patterns and regularities within given data, and presents the discovered information with models which are generally referred to as *rules.* The best known example is *association rule discovery* (Agrawal, Imielinski and Swami; 1993). Others include *implication rule discovery* (Brin, Motwani, Ullman and Tsur; 1997), *correlation set discovery* (Brin, Motwani and Silverstein; 1997), *sequential pattern discovery* (Agrawal and Srikant; 1995), *contrast set discovery* (Bay and Pazzani; 2001) and *causal structure discovery* (Silverstein, Brin, Motwani and Ullman; 2000). Exploratory rule discovery is powerful in the context of data mining where multiple models that perform equally well exist. It can present the retrieved information in a form that is understandable to the users. However, the features of exploratory rule discovery bring other problems to be addressed. First, since exploratory rule discovery generates multiple models, it is inevitable that the number of resulting rules can become too large for manual analysis. It has also been recognized that some rules discovered are potentially "uninteresting" and can be removed without jeopardizing the performance of the outcome. Second, searching for multiple models also requires expensive computation and data accesses. To make things worse, a great amount of computation can be wasted on searching for un-useful models.

Abundant research has been devoted to circumvent these dilemmas (Bayardo, Jr., Agrawal and Gunopulos; 1999; Lakshmanan, Ng, Han and Pang; 1999; Grahne, Lakshmanan and Wang; 2000; Bonchi and Geothals; 2004; Lawler and Wood; 1966; Pei et al.; 2001; Dong and Li; 1998; Agrawal and Srikant; 1994; Shenoy, Haritsa, Sundarshan, Bhalotia, Bawa and Shah; 2000; Zaki, Parthasarathy and Li; 1997; Agrawal and Shafer; 1996; Zaki, Parthasarathy, Ogihara and Li; 1997b; Park, Chen and Yu; 1995b; Toivonen; 1996; Savasere, Omiecinski and Navathe; 1995; Lin and Dunham; 1998; Yip, Loo, Kao, Cheung and Cheng; 1999; Park, Chen and Yu; 1995a; Webb and Zhang; 2005; Burdick, Calimlim and Gehrke; 2001; Agarwal, Aggarwal and Prasad; 2000). Most of such techniques were developed with qualitative attributes or discretized quantitative attributes. However, simply discretizing the quantitative attributes results in information loss, because qualitative attributes have lower levels of measurement scale than their quantitative counterparts. Research on how to efficiently remove potentially uninteresting rules with undiscretized quantitative attributes in large, dense databases is limited. Due to the fact that rules with undiscretized quantitative attributes which are described using distributions can provide richer information than can those with discretized quantitative attributes, the need for developing pruning techniques for rules with undiscretized quantitative attributes are immediate and strong.

In this chapter, a brief introduction to the background of our research is outlined. Examples are presented to argue that there is a wide scope of application for exploratory rule discovery. Next, we explain the motivations of our research. Contributions are then described, followed by the thesis organization.

1.1 Background

Many machine learning systems like classification learners, discover a single model from the available data that is expected to maximize the accuracy or some other specific measures of performance on unknown future data. Predictions or classifications are done on the basis of this single resulting model (Webb; 2005). Examples include instance-based classifiers, decision trees (Quinlan; 1993), artificial neural networks, genetic algorithms and the Naive-Bayes classifier. However, it is not always optimal to choose only one of the "best" models over others in data mining contexts. Alternative models exist that perform equally well as those which are selected. The criteria for deciding whether a model is best or not also varies with the context of application.

Rule discovery techniques are proposed to overcome this problem by searching for multiple models which satisfy a user-specified set of criteria. Thus, the users are provided with alternative choices. Better flexibility is achieved in this way. Resulting models are chosen against measures that can be readily translated for decision making. Association rule discovery is a typical example of such approaches.

Ever since the introduction of *association rule discovery* (Agrawal et al.; 1993), exploratory rule discovery has been widely employed to search for underlying interrelationships among attributes in databases. The information discovered is then modelled as rules for business or management oriented guidance.

Two classes of techniques witnessed the development of exploratory rule discovery for addressing its inherent drawbacks as has been identified, including the problem of huge numbers of resulting rules and the efficiency problem. The first class aims at effectively pruning resulting rules to control the size of discovered resulting set and provide the users with optimal outcomes, as well as at improving the rule discovery efficiency. Some of the outstanding examples are Bayardo, Jr.

et al. (1999); Lakshmanan et al. (1999); Grahne et al. (2000); Bonchi and Geothals (2004); Lawler and Wood (1966); Pei et al. (2001); Dong and Li (1998); Liu, Hsu and Ma (1999a); Brin, Motwani and Silverstein (1997); Piatetsky-Shapiro (1991); Carter, Hamilton and Cercone (1997); Aggarwal and Yu (1998); Zhong, Yao and Ohsuga (1999); Gray and Orlowska (1998); Srikant, Vu and Agrawal (1997); Han and Fu (1995); Klemettinen, Mannila, Ronkainen, Toivonen and Verkamo (1994); Baralis and Psaila (1997); Klemettinen et al. (1994); Meo, Psaila and Ceri (1996); Bayardo (1998); Lin and Kedem (1998); Gunopulos, Mannila and Saluja (1997); Pasquier, Bastide, Taouil and Lakhal (1999b); Pei, Han and Mao (2000); Zaki and Hsiao (1999); Webb and Zhang (2002); Liu, Hsu and Ma (1999b); Bay and Pazzani (2001); Aumann and Lindell (1999). The second class of techniques is developed for improving exploratory rule discovery efficiency. Examples are Agrawal and Srikant (1994); Shenoy et al. (2000); Zaki, Parthasarathy and Li (1997); Agrawal and Shafer (1996); Zaki, Parthasarathy, Ogihara and Li (1997b); Park et al. (1995b); Toivonen (1996); Savasere et al. (1995); Lin and Dunham (1998); Yip et al. (1999); Park et al. (1995a); Webb and Zhang (2005); Burdick et al. (2001); Agarwal et al. (2000); Agarwal, Aggarwal and Prasad (2001).

1.2 Applications of Exploratory Rule Discovery

Considering the outstanding characteristics of exploratory rule discovery, including easy interpretability of resulting models for decision making, excellent scalability, relatively simple algorithms, it has been extensively applied in a wide range of real world applications since its introduction. *Association rule discovery* (Agrawal et al.; 1993), which is a typical member of exploratory rule discovery, was originally developed in the context of *market basket data.* The resulting rules can express how important products or services are related to each other, and can be easily interpreted so as to suggest particular actions.

Existing applications for exploratory rule discovery include:

1. Legal domain and security issues (Cuppens and Mige; 2002; Bench-Capon, Coenen and Leng; 2000; Johnston and Governatori; 2003; Governatori and Stranieri; 2001; Lee and Stolfo; 1998; Lee, Stolfo and Mok; 1999)

2. Knowledge discovery with the web (web mining and E-commerce) (Loh, Wives and de Oliveira; 2000; Wu and Jajodia; 2004; Fonseca, Golgher, de Moura and Ziviani; 2003; Ma, Liu and Wong; 2000; Cooley; 2000; Dua, Cho and Iyengar; 2000; Lo and Ng; 1999; Abbattista, Degemmis, Licchelli, Lops, Semeraro and Zambetta; 2002)

3. Geographic analysis (Ale and Rossi; 2000; Koperski and Han; 1995; Han, Koperski and Stefanovic; 1997; Koperski, Han and Adhikary; 1998).

4. Business analysis and decision management (Brijs, Swinnen, Vanhoof and Wets; 1999) Commercial (Hipp, Güntzer and Grimmer; 2001; Chou, Grossman, Gunopulos and Kamesam; 2000)

5. Medical science and bio-informatics analysis, (Doddi S; 2001; Oyama, Kitano, Satou and Ito; 2000; Satou; 1997; Wetjen; 2002; Toivonen, Onkamo, Hintsanen, Terzi and Sevon; 2004).

6. Text mining and document analysis (Ahonen, Heinonen, Klemettinen and Verkamo; 1998; Kawano and Hasegawa; 1998; Wong, Whitney and Thomas; 1999).

7. Applications in machine learning to improve classification performance (Liu, Ma, Wong and Yu; 2003; Alhammady and Ramamohanarao; 2004)

8. Remotely sensed data (Dong, Perrizo, Ding and Zhou; 2000).
9. Library application (Michail; 1999, 2000).
10. Student management (Ma, Liu, Wong, Yu and Lee; 2000).
11. Census data (Brin, Motwani, Ullman and Tsur; 1997).

1.3 Motivations

Although intensive research has been contributed to exploratory rule discovery with qualitative attributes and discretized quantitative attributes, little has been done regarding exploratory rule discovery with undiscretized quantitative attributes to efficiently identify potentially uninteresting rules in which context status of quantitative attributes are described using distributional statistics instead of frequencies as for rules with discretized quantitative attributes. Most of existing rule pruning techniques are initially developed in exploratory rule discovery with qualitative or discretized quantitative attributes. We identified the need for designing rule techniques specifically for rules with undiscretized quantitative attributes.

Moreover, although discovering rules with undiscretized attributes offers the users with more information of a quantitative attributes, this is achieved at the cost of considerable amount of additional computation and data accesses for collecting the necessary statistics and parameters. Thus, the inherent efficiency problem of exploratory rule discovery is exacerbated. With the volume of data to be processed increasing at a rapid pace nowadays, the demand for more efficient and effective rule discovery with undiscretized quantitative attributes in very large, dense databases is staring us in the face. However using the frequent item set framework, as Aumann and Lindell (1999) have done, requires excessive memory as well as formidable computation for maintaining candidates during the rule discovery. We believe that this problem can be tackled by reducing computational redundancies and unnecessary data accesses.

Motivated by these observations, we devote ourselves to designing and developing rule pruning techniques for exploratory rule discovery with undiscretized attributes. We also study methods for improving the efficiency of the rule discovery algorithms. We follow Webb (2001) by adopting a new framework: the OPUS algorithm, for rule discovery in order to speed up rule pruning with undiscretized attributes. We design algorithms that can work with very large, dense databases for which the current popular algorithms fail. We analyze the features of rule discovery with undiscretized quantitative attributes and propose techniques for effective and efficient rule pruning in this context.

1.4 Thesis Contributions

Herewith we summarize the contributions of this thesis as follows.

1. Existing exploratory rule discovery techniques are classified by us into two classes: distributional-consequent rule discovery and propositional rule discovery. We propose comprehensive descriptions of these two kinds of exploratory rule discovery. (Chapter 2)

2. We analysis the differences between distributional-consequent and propositional rule discovery and explain why the techniques for identifying rule interestingness are different for these two types of exploratory rule discovery. (Chapter 3)

3. We propose the definition of *derivative extended rules* (a further developed definition for insignificant rules) suitable for impact rules discovery. (Chapter 4)

4. We propose an algorithm that can automatically remove derivative extended impact rules during rule discovery. The algorithm is constructed on the basis of the OPUS search algorithm. (Chapter 4)

5. We present the definition of *derivative partial rules*, which is theoretically argued to be a type of potentially uninteresting rule that have not been identified by previous research. (Chapter 4)

6. A new hierarchy within different impact rules according to their interestingness is proposed and the relationship is explained. (Chapter 4)

7. We introduce the triviality impact rule filter to enable more powerful search space pruning. We argue that the triviality filter is a complement for the derivative extended rule filter. (Chapter 4)

8. We propose the difference set statistics derivation technique during the rule discovery to remove unnecessary data accesses and computation while searching for impact rules. (Chapter 4)

9. The circular intersection approach is proposed for further improving the efficiency of derivative extended rule filter of impact rule discovery. (Chapter 4)

10. We evaluate the effectiveness of all the techniques proposed in this thesis on several well selected large, dense databases to test how the techniques can effectively and efficiently discard numerous potentially uninteresting impact rules. The experimental results validate our argument that our algorithms perform much better than other existing techniques for removing insignificant distributional-consequent rules in very large, dense databases. (Chapter 5)

1.5 Thesis Organization

In the next chapter, we summarize the related terms and concepts that will be quoted in this thesis. Terminologies employed by researchers to described exploratory rule discovery tasks vary from work to work. We hence systematically clarify the connections and differences between some frequently used terms to avoid possible confusions in the later part of this work. Since the techniques that we are going to propose were mainly about how to efficiently prune rules using statistical methods, we also briefly explain some related statistical concepts. A review is presented of approaches for exploratory rule discovery. Existing techniques are classified into two categories: *propositional rule discovery* and *distributional-consequent rule discovery*. The reasons why we choose the OPUS based k-optimal rule discovery as the basis of our research are demonstrated. A few key ideas which lay the groundwork for the descriptions of our algorithm are explained. A formal description of k-optimal impact rule discovery which is utilized throughout this thesis is presented in the end of chapter 2.

Chapter 3 is primarily devoted to reviewing previous research in the exploratory rule discovery community. We discover that impressive amount of efforts have been contributed to developing rule pruning and efficiency improving algorithms for propositional rule discovery, while little has been done with distributional-consequent rule discovery. Considering the differences between these two kinds of exploratory rule discovery, it is meaningful to design techniques specially for rule pruning and fast rule discovery with distributional-consequent rules. We group existing techniques for propositional rule pruning into two families. One for incorporating constraints and the other for removing rules that are spurious or potentially uninteresting due to the presence of other rules. We also discuss the properties of constraints with respect to whether they can be pushed deep into the rule discovery processes for search space pruning. Applicability of the reviewed propositional rule pruning techniques with the distributional-consequent rule discovery is discussed. Finally, some fast algorithms that have been proposed are summarized.

In chapter 4, motivated by the reviews and discussions in previous chapters, we develop efficient algorithms for discovering interesting rules with undiscretized quantitative variables as consequents (or targets) in k-optimal impact rule discovery. Since the previous algorithms for discarding potentially uninteresting distributional-consequent rules, which are commonly referred to as the *insignificant rules*, are based on the frequent itemset generation, which leads to excessive memory and maintenance overheads, they are not optimal with very large, dense databases. We further develop such techniques into the derivative extended rule filter, and propose a new implementation for the same task.

We also argue that existing techniques alone are not enough for removing some class of potentially uninteresting rules that can be theoretically identified. We clarify the relationship among rules regarding their potential interestingness and propose and efficient techniques for filtering a new class of potentially uninteresting impact rules, which is defined as the class of *derivative partial rules.*

Several techniques are also introduced to achieve dramatic efficiency gains. The *triviality filter* is proposed as a complement and alternative to the derivative extended rule filter due to its anti-monotonicity. The *difference set statistics derivation* and the *circular intersection* approaches are also designed to reduce the computational and data access redundancies.

Chapter 5 deals with the empirical evaluations of all the techniques proposed in chapter 4. Comparisons are also done with an efficient implementation of Apriori to show the advantages of our techniques. The effectiveness of our proposed filters is analyzed and the influences of the efficiency improving techniques are also exhibited experimentally. The results support the theoretical analyses of the performance of our proposals.

Conclusions are drawn in chapter 6. We summarize the key issues presented in this thesis by highlighting our contributions. Suggestions for potential future research are made in the last stage to conclude this thesis.

Chapter 2

Concepts and Problem Settings

In the first part of this chapter we are going to present an introduction to terms and concepts that are frequently referred to in this thesis. The goal of our research is to address the pruning of rules with undiscretized attributes, which are described using distributional statistics. For such rules, statistical methods are often applied to evaluate their interestingness. Therefore, we explain some statistical terms in the first place. Then, the definitions and notations related to exploratory rule discovery are introduced. Similar terms are listed together and differences are explained.

Next, we review existing techniques for exploratory rule discovery including *association rule discovery*, *contrast set discovery* and *causal structure discovery*. We also classified existing techniques into two categories. Propositional rule discovery techniques deal with qualitative (categorical) attributes or discretized quantitative (numeric) attributes only. The resulting propositional rules are described using frequency statistics and measures. However, it is recognized that quantitative attributes are indispensable components of many databases. Propositional rule discovery is tailored for discovering inter-relationships between qualitative attributes or discretized quantitative attributes only. Researchers have studied numerous methods for optimal discretization (Srikant and Agrawal; 1996; Brin, Rastogi and Shim; 1999). Nevertheless, discretization entails information loss. Propositional rules perform poorly in summarizing associations between quantitative variables and qualitative attributes. Distributional-consequent rule discovery resolves this problem by describing such inter-relationships using distributions.

In the end of this chapter, we illuminate the merits of k-optimal rule discovery, and introduce the basis on which our k-optimal rule discovery algorithm is constructed: the OPUS algorithm. Reasons are explained for why we select k-optimal impact rule discovery as the foundation of our research, followed by its formal description.

2.1 Statistical Terms

Statistics and probabilities is one of the principles that is closely related to exploratory rule discovery. Statistical methods and terms are used throughout this thesis. Since statistical terminologies differ from work to work in the literature, a clear explanation of terms and concepts can simplify our discussions.

2.1.1 Qualitative vs. Quantitative Attributes

The data mining literature contains a huge variety of terms for describing different types of data or attributes. This has the potential to cause confusion. In this thesis we choose to classify attributes into two types: *quantitative* attributes and *qualitative* attributes.

Qualitative attributes is also known as *categorical* attributes in some other work. A distinct characteristic of such attributes is that they are only classified in categories, but not numerical measures (Bhattacharyya; 2000). Qualitative attributes come in two classes, namely, *nominal* and *ordinal*. Nominal attributes are attributes that are exhaustively divided into mutually exclusive categories with no rankings that can be applied to these categories. *City names* of a country is an example nominal attribute. Ordinal attributes are different from nominal ones in that the categories into which they are classified can be ordered, like the grades of student evaluations: fail, pass, credit, distinction, and high distinction.

Quantitative attributes, which are also referred to as *numerical* attributes, are attributes that are measured on a numerical scale and to which arithmetic operations can be applied. Quantitative attributes are classified into two types in this thesis: *discrete* and *continuous*. Discrete quantitative attributes have a measurement of scale composed of distinct numbers with gap in between. The number of students in a class is a typical example for discrete quantitative attributes. The other type of quantitative attributes is the class of *continuous* attributes which can ideally take any value. In other words, the measurement scale of continuous attributes does not have gaps. For instance, the height of one-year-old seedlings is a continuous quantitative attribute.

In most exploratory rule discovery, quantitative attributes are discretized. We treat such discretized quantitative attributes as ordinal qualitative attributes in this thesis.

2.1.2 Sample and Population

In order to retrieve useful information for further processing and analyzing in data mining, relevant data must be collected. Data flucutations are unavoidable, even if the data are collected under strictly controlled conditions of environment.

Considering that it is practically infeasible to access the exhaustive set of data, research and tests can only be performed with reference to data collected in the course of an experimental investigation. This is also the case with exploratory rule discovery. The data acquired through this process is referred to as a *sample*, while the vast amount of potential data which can be conceived in a given context is named the *population*. *Sampling* is the name for gathering data from the population through observations to generate a sample for future research or testing.

In this thesis, every database to which we employ exploratory rule discovery is a sample from the population, in whose characteristics we are actually interested, and which cannot be exhaustively accessed. By retrieving information from the sample, we are seeking models that can describe the features of the population from which the sample is drawn.

2.1.3 Hypothesis Tests and Errors

Mining information using sample data to summarize the features of a population is running the risk of discovering models (rules) which appear to be correct or *interesting* with reference to the sample, yet turns out to be incorrect or uninteresting with regard to the population. *Hypothesis tests*, which are also referred to as *significance tests* or *statistical tests* in this thesis, are usually adopted to assess whether a claim or conjecture, derived from a sample should be accepted as likely or not when generalized to the population, at a *significance level.*

However, errors are inevitable with the results. In the community of exploratory rule discovery, there are two types of errors related to hypothesis tests: the type-1 error which lead to reject a model or a rule while it should be accepted, and the type-2 error of accepting a model or a rule which is incorrect or uninteresting. The risks of exploratory rule discovery suffering from such errors are high (Webb; 2005).

2.2 Exploratory Rule Discovery

Although traditional data mining techniques can efficiently learn a model for classification or prediction, they do not transverse the solutions space completely. No guarantee is made about the predictiveness of the resulting model. What is contained in the resulting models is the discrimination ability over other potential alternatives. Potential solutions are ranked in a fixed order, so that a rule is desirable only if those in front of it fail to be. However, the ranking of rules differs from one application to another. Whether a rule is "best" or not depends and different learning algorithms, even if run with the same parameter settings, will generate different solutions. Another drawback for these techniques is the inconvenience for incorporating some useful criteria during learning process, like *minimum support*. There are risks for *over-fitting*, *under-fitting* and *induction bias* (Quinlan; 1993) in the models found.

Exploratory rule discovery is proposed to overcome these disadvantages. All models are generated, instead of one, that satisfy a specific set of criteria. The set of criteria are commonly referred to as *constraints* (for the definition of *constraints*, please see chapter 4). It guarantees that, with the same parameter settings applied, resulting solutions are complete and the same results can be yielded. Rules provides concise statement for the implicit knowledge that can easily achieve user understandability (Gunopulos et al.; 1997). Notably, some exploratory rule discovery techniques can address the discovery of interesting rules in data where no pre-specified classes exists.

Exploratory rule discovery aims at deriving the characteristics of a population, by searching for implicit patterns or regularities within sample data drawn from the population.

In this section, we give formal definitions of exploratory rule discovery and related terms after studying existing techniques, as well as different types of exploratory rule discovery.

Definition 1 (Exploratory Rule Discovery) *Exploratory rule discovery is a term for data mining techniques that retrieve all models which satisfy some user-specified* *set of criteria, called constraints, in a real world population by accessing a sample drawn from that population. The discovered rules can represent implicit knowledge in the data in a concise and human-apprehensible manner.*

2.2.1 Rules Discovered using Exploratory Rule Discovery

Easy interpretability of resulting models for decision making can enhance the power of data mining techniques in certain occasions. One of the famous examples is the famous market basket data, in which exploratory rule discovery is able to explicitly clarify the relationship among various products and services to an extent that the traditional data mining techniques cannot achieve. This results in the potential to provide better guidance in decision making.

Exploratory rule discovery techniques that are concerned with qualitative attributes or only are given the name *propositional rule discovery.* The rules generated by propositional rule discovery are composed of Boolean conditions only and are described using frequencies and propositional measures. There is another class of exploratory rule discovery to which we apply the name *distributional-consequent rule discovery.* It is motivated by the need for providing better descriptions for quantitative attributes, which are described using their distributions. Bodies of rules discovered by distributional-consequent rule discovery are composed of two sets, one set of Boolean conditions presenting the features of a subset of records and one of more quantitative attributes, called targets, which are described using distributional statistics.

The definition of rules that are discovered using exploratory rule discovery is given below.

Definition 2 (Rules discovered using exploratory rule discovery) *A* rule *discovered using exploratory rule discovery is composed of two parts: the* body *and the* description. *The* body *of a rule contains a set (subset) of Boolean conditions* *derived from the attributes in the databases, denoting the common features of a subset of data, and, for some applications, a set of variables in whose performance the users are interested. The* description *is a set of parameters or measures describing the performance or status of the subset of data.*

In some exploratory rule discovery techniques, the resulting rule bodies are divided into two parts: the *antecedent* and the *consequent.* For a rule taking the form $A \rightarrow C$, A is the antecedent representing the premises of a dataset, while C is called the consequent, which displays some common features of the dataset represented by A. The relationship between these two parts of rules are of most concern for the users. In some work, *left hand side (LHS)* is used for antecedent and *right hand side (RHS)* for consequent.

2.2.2 Terms of Exploratory Rule Discovery

Here are some exploratory rule discovery related notions.

Attributes and Conditions

An attribute is a property or characteristic for an entity. A *condition* is a Boolean predicate which characterizes a qualitative attribute taking a certain categorical value, or a quantitative attribute taking a value in a given range. A condition in this thesis corresponds to an *item* in many other works Agrawal et al. (1993).

Records

A record is a row in a database. For propositional rule discovery, a *record* is an element to which we apply conditions, while for distributional-consequent rule discovery, a record is a *pair* $< c, v >$, where c is the nonempty set of Boolean conditions, and v is a set of values for the quantitative variables in whose distribution the users are interested. In association rule discovery, records are often referred to as a *transaction.*

Database

A *database* is a finite set of *records*, which is also called a *dataset*.

Relations among Rules

A rule r_1 is a *parent* of rule r_2 if the antecedent of r_1 is a subset of that of r_2 and the consequents of both rules are identical. By contraries, r_2 is a *child* of r_1, if the antecedent of r_2 is a superset of the antecedent of r_1. If the cardinality of the antecedent of r_1 is smaller than the body of r_2, r_2 is defined as a *direct parent* of the r_1, otherwise, it is a *non-direct ancestor* of the first rule.

A parent rule of rule r is also referred to as a *generalization* or a *simplification* of r. A child rule can also be called a *specialization* of its parent.

Coverset and Coverage

We use the notation $coverset(A)$, where A is a conjunction of conditions, to represent the set of records that satisfy the condition (or set of conditions) A. If a record x is contained in $coverset(A)$, we say that x is *covered* by A. $Coverset(\emptyset)$ includes all the records in the given database. $Coverage(A)$ is the number of records covered by A: $coverage(A) = |coverset(A)|$. A set of conditions or items are often referred to as an *itemset*, which represents a set of records covered by the itemset as well.

2.2.3 Review of Propositional Rule Discovery

Most existing exploratory rule discovery techniques discover rules with qualitative or discretized quantitative attributes only. Bodies of rules generated by these techniques include only Boolean conditions. One typical example is *association rule discovery*, firstly proposed by Agrawal et al. (1993) to discover underlying associations in market basket data. Their approach was constructed in two separate steps. In the first step, *Frequent itemsets*, which are itemsets that contain at least a given number of transactions (or records), are generated using the given data sample. After the frequent itemset generation stage, association rules that satisfy user-specified constraints are derived from the resulting frequent itemsets and useful information of the resulting association rules are collected. The frequent itemset approach is widely employed in exploratory rule discovery. The body of an association rule is separated into two disjunct parts: the *antecedent* and the *consequent*. Both the antecedent and the consequent of an association rule are allowed to have arbitrary number of conditions.

Brin, Motwani and Silverstein (1997) proposed a technique which applies a chi-square test to sets of items in a database to discover *correlation rules*. The body of a correlation rule is a set of correlated items. Instead of dividing the rule body into an antecedent and a consequent, correlation rule discovery treats all conditions in the rule body symmetrically. The generated correlated sets consist of only items that are positively correlated with each other. By contrast, with association rules, the items in discovered rules are not assured to be correlated. It is argued by them that correlation rule discovery is able to yield results that are more in accordance with prior knowledge of the structure of data. Brin, Motwani, Ullman and Tsur (1997) further developed the definition of correlated set discovery into *implication rule discovery*, which is primarily composed of discovery processes in which rules with implications between the antecedent and the consequent can be found, as opposed to correlation sets which measures the co-occurrence only. Implication rules, like the association rules are divided into rule antecedents and consequents. An arbitrary number of Boolean conditions are allowed on both rule antecedents and consequents.

Contrarily, *emerging pattern discovery* (Dong and Li; 1999) and *contrast set* (Bay and Pazzani; 2001) endeavour to find rules with a single *target variable*, or *target* for short, as rule consequent. The measure for contrast sets and emerging patterns interestingness is how greatly the frequency of the target values for the sets of records which are covered by rule antecedents differ from each other. *Emerging pattern discovery* looks for itemsets which cover two datasets whose support ratio is greater than a given threshold, called minimum growth rate.

$$growthrate = \frac{support(dataset_1)}{support(dataset_2)} > min\_growthrate$$

Emerging patterns are presented using itemset borders composed of the minimum set of records and the maximal set of records. Such emerging patterns enable the description of contrasts between two groups. However, the techniques were restricted to only two groups at a time. *Contrast set discovery* (Bay and Pazzani; 2001) was developed for automatically detecting all differences between contrasting groups from observational multivariate data. It is different from the previous techniques in that contrast set discovery is concerned with multiple target groups for one attribute and the resulting models can highlight the dissimilarities between all these groups. It was later argued by Webb, Butler and Newlands (2003) that the features of contrast set discovery for deriving contrasts can also be achieved using association rules.

Classical exploratory rule discovery can only discovery itemsets that imply statistical relationships instead of causal relations between rule antecedent and consequent. It may also be the case that the rule $A \rightarrow C$ is a strong rule only by virtue of other conditions. *Causal structure discovery* was suggested by Silverstein et al. (2000) for mining causal relations. The authors identify the causality among conditions assuming the *Markov condition.* The causal rules found can illuminate both the existence and the lack of causality. Identifying the lack of causality can effectively reduce erroneous decisions.

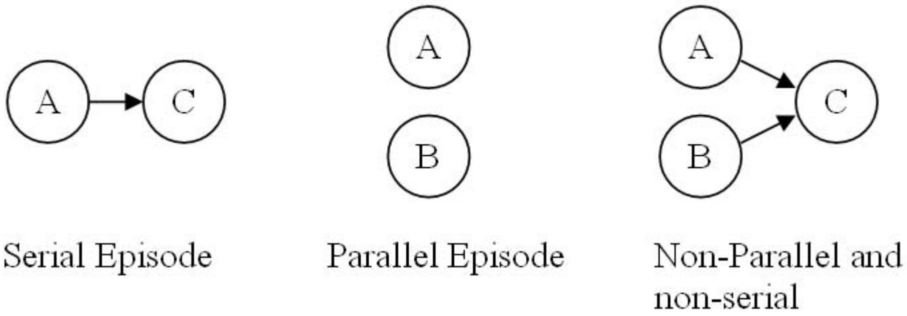


Figure 2.1: Different episodes

Exploratory rule discovery has been extended to mine implicit regularities and patterns in sequential or time-related databases. In such databases, sequential relationships exist among records and time series have to be taken into account when searching for useful regularities. Emerging patterns can be applied to this research area, while there are plenty of other techniques. *Sequential pattern* discovery mines user buying patterns in time-stamped market basket data. Three algorithms were proposed by Agrawal and Srikant (1995) for mining sequential patterns. A set of ordered itemsets were generated as discovered rules. Variants of support and confidence are defined to describe status of the discovered patterns. Resulting rules are represented as a sequence of events that happens in a fixed order, but not necessarily consecutively. *Episode discovery* which was proposed by Mannila, Toivonen and Verkamo (1997) is also a form of exploratory rule discovery for mining time related inter-relationships in databases. Rules generated consist of items that do not exist in the same time but have associated time of occurrence. Actually, the discovered episodes are acyclic graphs of events whose edges specify the temporal relationship without timing intervals restrictions as shown in figure 2.1.

For example, with the last episode in figure 2.1, A & $B \rightarrow C$ means if A and B both happen (regardless of the order), C will occur soon.

Traditional exploratory rule discovery results in rules that are *ample* (with at least minimum support) in all the given data regardless of the time. However, some interesting regularities only exist in restricted time intervals, such rules may be discarded due to a very low value for support. For example, a rule meaning that people are inclined to buy coffee and donuts during breakfast time may be discarded based on the traditional structure. however, they may be of little interest for decision maker, since this is a well-known fact. *Cyclic association rule discovery* (Ozden, Ramaswamy and Silberschatz; 1998) and *partial periodic pattern discovery* (Han, Dong and Yin; 1999) both seek rules that represent periodic behaviours in a sequential database. *Cyclic association rule discovery* can successfully capture periodicities, but discover only rules that are true in every cycle. Resulting cyclic rules display cyclic variations in a database over time. *Partial periodic pattern discovery* was proposed to discover models like "somebody is apt to do something at sometime during the working days" which cyclic association rule discovery may fail to find.

Inter-transaction rules Previous techniques mine rules which only describe the characteristics within the same transaction or record. These are called *intra-transaction* rules. *Inter-transaction rules* can result in rules like: "after company A open a branch in a certain area, company B will also open a branch in a month within a mile" (Lu, Han and Feng; 1998; Bettini, Wang and Jajodia; 1998). Inter-transactional techniques discover rules whose antecedents and consequents are both episodes that happen in accordance to the restrictions in rule descriptions.

2.2.4 Coping with Quantitative Attributes and Distributional-consequent Rule Discovery

Previously mentioned methods for exploratory rule discovery are all about discovering rules with qualitative attributes only. Considering the fact that a great number of quantitative attributes exist in real world databases, relatively little has been done to effectively process quantitative attributes. Motivated by this, several techniques for mining rules with quantitative attributes are proposed, most of which are constructed on bucketing or discretization. Srikant and Agrawal (1996) proposed an extended definition of association rules, named *quantitative association rules*, by considering the intervals of quantitative values. Quantitative attributes are first discretized using equi-depth partitioning with a partial completeness measure for determining the intervals . Then, consecutive intervals are merged until the minimum support is satisfied. They identified several difficulties in determining the number of intervals for quantitative attributes.

1. If the intervals are too small, a large portion of rules generated may turn out with low supports. Setting the minimum support too high can lead to removal of interesting rules, while a low support may probably result in computational infeasibility.

2. If the intervals are too large, many rules generated may suffer from low confidence, this, in turn can lead to discarding of interesting rules.

3. If the number of intervals are too large, the resulting number of rules together with the execution time may increase unacceptably. As a by product, many of the resulting rules may be uninteresting.

A greater-than-expected-value criterion was introduced to identify the interestingness of the output rules by Srikant and Agrawal (1996). A rule is regarded "interesting" if and only if its support or confidence is R times higher than the expected value, where R is ratio specified by the user.

Fukuda, Morimoto, Morishita and Tokuyama (1996b) proposed approaches for mining two kinds of optimized rules with only one quantitative attribute in rule antecedents and a qualitative attribute as consequents. They searched for *confident rules* (rules that satisfy the minimum confidence constraint) with optimized support, and *ample rules* (rules that satisfy the minimum support constraint) with optimized confidence. Randomized bucketing is used to discretized the quantitative attribute before the merging of intervals takes place. Only one optimized interval can be generated using their techniques. Rastogi and Shim (2001) extended the above approach to mine rules with several disjunctions of optimized intervals. Fukuda, Morimoto, Morishita and Tokuyama (1996a) considered occasions where two quantitative attributes are allowed on rule antecedents. They find optimized 2-dimensional rectangular or admissible regions that optimized an interestingness measure called *gain* as well as support and confidence.

$$Gain(r) = support(r) \times (confidence(r) - minmum\_confidence)$$

In this formula, r is a rule. Brin et al. (1999) further developed this approach to mine optimized gain rules in which multiple optimized regions can be identified efficiently.

Wang, Tay and Liu (1998) also proposed a technique for merging adjacent intervals in a bottom up manner to maximize the interestingness of a set of rules, based on a modified B-tree. The *J-measure* is used for measuring the interestingness of merged rules.

$$J(A \rightarrow C)P(A)[P(C|A)\log_2 \frac{P(C|A)}{P(C)} + (1 - P(C|A))\log_2 \frac{1 - P(C|A)}{1 - P(C)}]$$

The operation that minimizes information loss due to merging is chosen. Their approach works well with skewed data.

Even though these discretize-and-merge techniques can reduce the information loss to a moderate degree, they cannot effectively describe the information regarding the influence of qualitative attributes on quantitative variables. Due to the fact that discretized qualitative attributes have lower levels of measurement scale than their undiscretized counterparts, it follows that distributions are the best descriptions for quantitative attributes.

Grounding on the preceding arguments, Aumann and Lindell (1999) propose a different type of *quantitative association rule discovery* with an additional quantitative variable, which is nominated by the users, added to the body of discovered rules. This quantitative variable, which is referred to by us as the *target*, constitutes the consequent of the quantitative association rules and the rule descriptions are distributional statistics, *mean* as an example, for describing the status of the target. Therefore, the technique they proposed belongs to the distributional consequent rule discovery.

Notice that using *quantitative association rule discovery*, which is the same as that adopted by Srikant and Agrawal (1996) for their technique, is confusing. On this observation, Webb (2001) extended the technique of Aumann and Lindell (1999), and applied the name *impact rule discovery*, which is the name adopted by us in this thesis.

2.2.5 K-Optimal Rule Discovery and The OPUS Algorithm

Like most exploratory rule discovery techniques, the approach of quantitative association rule discovery put forward by Aumann and Lindell (1999) is based on the *frequent itemset framework*. It is well known that, when coping with very dense databases, some of the frequent itemset related approaches are expensive in terms of memory usage and data maintenance expenses during the course of rule discovery. The situation deteriorates when the databases are very large and dense.

Moreover, most exploratory rule discovery makes use of a minimum support constraint. The minimum support constraint is usually enforced to prune the NP-hard search space and make it computationally feasible. There are two reasons why this framework is not always appropriate. First, it is elusive to subjectively choose a threshold for support which is an objective measure. The case is often that there is a narrow range of possible minimum support value below which the number of resulting rules can become unwieldy. Second, a support is not always desirable as a measure of interestingness. People are sometimes interested in a strong feature manifested by a small part of the data, yet a rule with a high support only result in rules which represent the behaviours of the majority. Rules with high support sometimes turn out to represent well known knowledge. An example is the famous *bread and milk* association in the market basket data. These items are commonly consumed by almost every household, but they yield relatively much lower profit than items like *vodka and caviar* (Cohen, Datar, Fujiwara, Gionis, Indyk, Motwani, Ullman and Yang; 2001) which are also highly correlated with an extremely low support. Given a specific minimum support higher than the support of vodka and caviar, exploratory rule discovery removes such rules automatically. One will never be able to tell what the minimum support should be in order to discover all rules that interest the users.

One avenue of attack is to use the *k-optimal* rule discovery for which no minimum support for the resulting rules is necessary. Instead, the users are requested to specify the number of rules they prefer. The rule discovery system automatically ranks the resulting models against a user-designated interestingness measure and the top k rules with the highest value for that measure are presented to the users. In this way, the number of resulting rules can be kept under control and the post-rule-mining processing is eased. Resulting rules are guaranteed to be the most “interesting” ones.

Our k-optimal rule discovery is constructed based on the OPUS (Optimized Pruning for Unordered Search) algorithm. OPUS is an admissible algorithm in which all the desirable solutions are guaranteed to be discovered, as opposed to heuristic algorithms in which no such guarantee is made. OPUS systematically searches through a tree-style search space as is illustrated in figure 2.2. In this figure each node is associated with a potential target. It is recognized that such a search space is exponential in size. For many rule discovery tasks, the number of nodes to be explored is extremely large (Webb; 2001). Only pruning can make the search space exploitable. A minimum support (or coverage) constraint is usually utilized for the purpose of search space pruning in the context of support-confidence-based rule discovery. Sometimes the minimum support has to be set very high in order to make the computation feasible. However, the OPUS algorithm enables efficient, dynamic space pruning during the course of rule discovery with a very low support, or even without a specific minimum support. In this way, the risks of discarding interesting rules are minimized. *Branch and bound* pruning techniques are introduced to effectively prune the search space that contains no solutions. It has been demonstrated that this approach can achieve dramatic reduction in computation expenses in comparison with previous algorithms applied to such a search space (Webb; 1995).

By enforcing the k-optimal constraint, the bounds for search space pruning change with time resulting in the gradual reduction of space to be explored. Contrarily, with the support and confidence framework only, the size of space to be searched is fixed once the minimum support is specified.

2.2.6 K-Optimal Impact Rule Discovery

With regard to the previous arguments, we choose to carry out our research using the k-optimal impact rule discovery proposed by Webb (2001) as a basis. In this

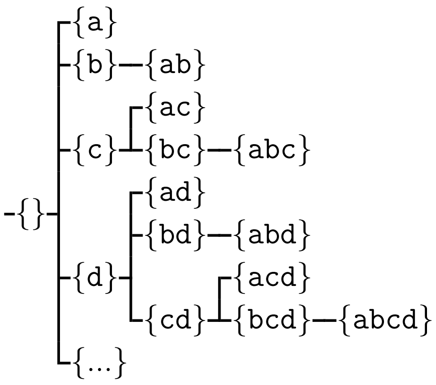


Figure 2.2: Fixed structure search space for OPUS

section, we give a former definition of k-optimal impact rule discovery, which is an extended version of that proposed by Webb (2001). We formalized our definition and characterized the terminology of k-optimal impact rule discovery to be used in this paper as follows:

1. An impact rule generated using our algorithm takes the form: $A \rightarrow target$. A is a conjunction of Boolean conditions, while *target* is described by the following statistics and measures: *coverage*, *mean*, *variance*, *maximum*, *minimum*, *sum* and *impact*.

2. *Impact* is an interestingness measure suggested by Webb (2001):

   $$impact(A \rightarrow target) = (tarmean(A \rightarrow target) - tarmean(\emptyset \rightarrow target)) \times coverage(A)$$

   In this formula, $tarmean(A \rightarrow target)$ denotes the mean of the *target* variable covered by A.

3. A k-optimal impact rule discovery task is a 6-tuple: $KMIIRD(\mathcal{C}, \mathcal{T}, \mathcal{D}, \mathcal{M}, \lambda, k)$.

   $\mathcal{C}$: is a nonempty set of Boolean conditions derived from the database, $\mathcal{C}$ the set of available conditions for resulting impact rule antecedents.

   $\mathcal{T}$: is a nonempty set of variables in whose distribution we are interested. In this thesis, we confine the number of target variables to 1.

   $\mathcal{D}$: is the database on which the k-optimal impact rule discovery is performed.

```
Algorithm: OPUS_IR(Current, Available)
  1. SoFar := $\emptyset$
  2. FOR EACH P in Available
     2.1 New := Current $\cup$ P
     2.2 IF it cannot be determined that $\forall x \subseteq Available : \neg solution(x \cup New)$ THEN
       2.2.1 generate the distribution statistics for the target
             variable
       2.2.2 record New $\rightarrow$ target
       2.2.3 OPUS_IR(New, SoFar)
       2.2.4 SoFar := SoFar $\cup$ P
     2.5 END IF
  3. END FOR
```

Table 2.1: Basic OPUS_IR algorithm

$\mathcal{M}$: is a set of constraints[1]. For most constraints, useful bounds for search space pruning can be derived, either tight or loose, and completeness of information is still sustained. Classes of constraints include anti-monotone, succinct (Han and Kamber; 2001), and monotone constraints (Pei et al.; 2001).

λ: $\{X \rightarrow Y\} \times \{D\} \rightarrow \mathcal{R}$ is a function from rules and databases to values, and defines a interestingness metric such that the greater the value of $\lambda(X \rightarrow Y, \mathcal{D})$ the more interesting this rule is given the database.

k: is a user specified integer number denoting the number of rules in the ultimate set of solutions for this task.

Pseudo code for a simple OPUS based algorithm for impact rule discovery (OPUS_IR) is displayed in table 2.1. In this table, *current* is the node in the search space whose children nodes are currently being explored. *Available* is the set of conditions that can be added to the conjunction of conditions in *current* to generate the antecedent of a potential impact rules: $New \rightarrow target$ for examination. A depth-first search is thus performed.

[1] Constraints will be discussed in more detailed in chapter 3.

tid	target	cat1	num	cat2
0	-1.5	A	13	C
1	0	B	4	D
2	2.3	B	10	C
3	11	A	4	C
4	0	A	15	D
5	-1	A	11	C
6	12.4	A	7	D
7	-2.7	A	11	C
8	6.8	B	7	D
9	12	A	3	C
10	-3	B	4	C
11	0	B	5	D
12	-1.2	A	11	D
13	1.6	A	14	C
14	0.5	B	8	C

Table 2.2: Database for algorithms illustration: mean=2.48, variance=28.7017

Database for Illustration

We contrived a fictitious database for better explanations of our algorithms in this thesis. This database contains 4 attributes: *target, cat1, num, cat2*. *Target* is the quantitative variable in whose distribution we are interested; *num* is a numeric variable which is discretized into two ranges: greater than 10 and smaller than or equal to 10.

The fixed search space for OPUS_IR with this database is shown in figure 2.3. Before the search algorithm operates, the data is loaded to the memory in a vertical layout[2] with each condition followed by a list of numbers for the records for which this condition is true. Then the program is run with the initial *current* set to $\emptyset$ and the initial *available* composed of all the available conditions. Every combination of conditions is guaranteed to be explored once and only once, regardless of the order in which conditions in the initial *available* are ranked.

[2] Please refer to the next chapter for explanation of database layouts.

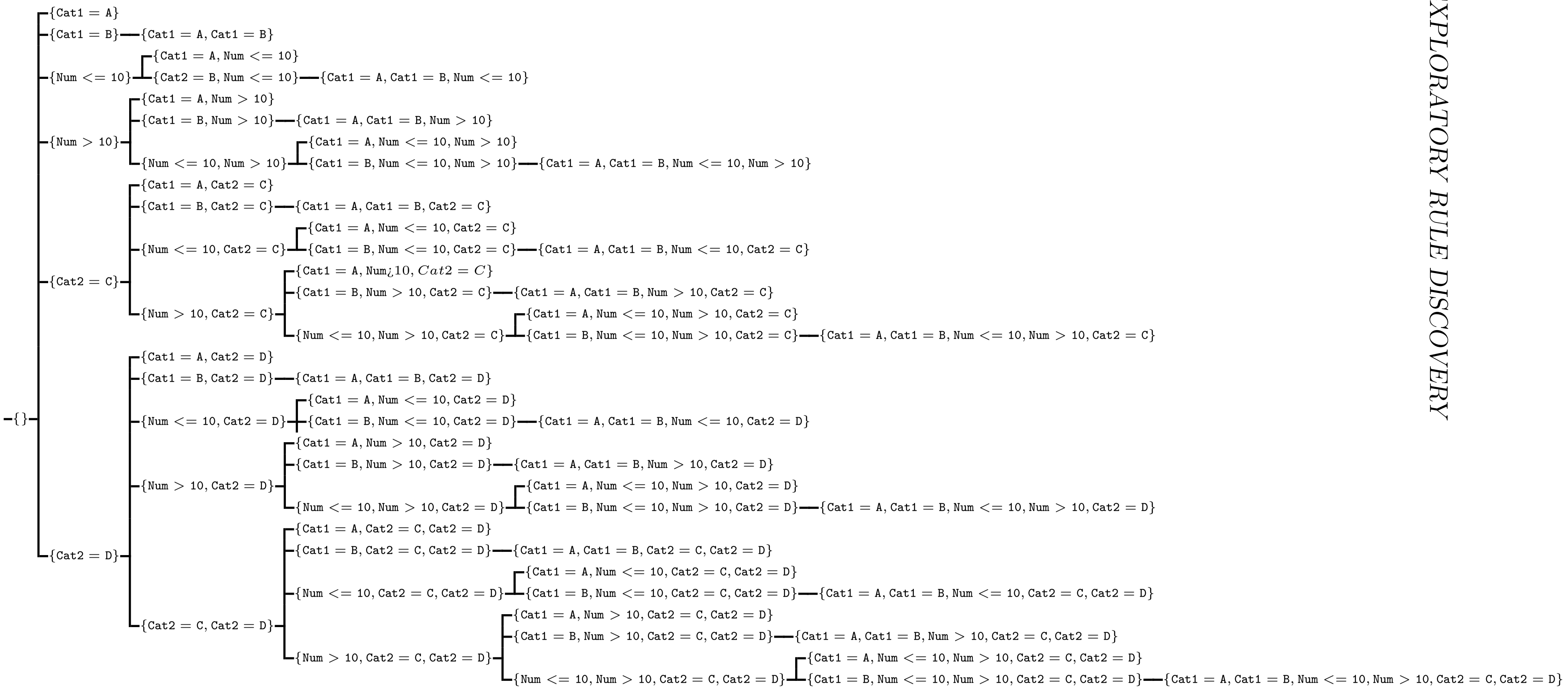

{}
{Cat1 = A}
{Cat1 = B}
{Cat1 = A, Cat1 = B}
{Num <= 10}
{Cat1 = A, Num <= 10}
{Cat2 = B, Num <= 10}
{Cat1 = A, Cat1 = B, Num <= 10}
{Num > 10}
{Cat1 = A, Num > 10}
{Cat1 = B, Num > 10}
{Cat1 = A, Cat1 = B, Num > 10}
{Num <= 10, Num > 10}
{Cat1 = A, Num <= 10, Num > 10}
{Cat1 = B, Num <= 10, Num > 10}
{Cat1 = A, Cat1 = B, Num <= 10, Num > 10}
{Cat2 = C}
{Cat1 = A, Cat2 = C}
{Cat1 = B, Cat2 = C}
{Cat1 = A, Cat1 = B, Cat2 = C}
{Num <= 10, Cat2 = C}
{Cat1 = A, Num <= 10, Cat2 = C}
{Cat1 = B, Num <= 10, Cat2 = C}
{Cat1 = A, Cat1 = B, Num <= 10, Cat2 = C}
{Num > 10, Cat2 = C}
{Cat1 = A, Num¿10, $Cat2 = C$}
{Cat1 = B, Num > 10, Cat2 = C}
{Cat1 = A, Cat1 = B, Num > 10, Cat2 = C}
{Num <= 10, Num > 10, Cat2 = C}
{Cat1 = A, Num <= 10, Num > 10, Cat2 = C}
{Cat1 = B, Num <= 10, Num > 10, Cat2 = C}
{Cat1 = A, Cat1 = B, Num <= 10, Num > 10, Cat2 = C}
{Cat2 = D}
{Cat1 = A, Cat2 = D}
{Cat1 = B, Cat2 = D}
{Cat1 = A, Cat1 = B, Cat2 = D}
{Num <= 10, Cat2 = D}
{Cat1 = A, Num <= 10, Cat2 = D}
{Cat1 = B, Num <= 10, Cat2 = D}
{Cat1 = A, Cat1 = B, Num <= 10, Cat2 = D}
{Num > 10, Cat2 = D}
{Cat1 = A, Num > 10, Cat2 = D}
{Cat1 = B, Num > 10, Cat2 = D}
{Cat1 = A, Cat1 = B, Num > 10, Cat2 = D}
{Num <= 10, Num > 10, Cat2 = D}
{Cat1 = A, Num <= 10, Num > 10, Cat2 = D}
{Cat1 = B, Num <= 10, Num > 10, Cat2 = D}
{Cat1 = A, Cat1 = B, Num <= 10, Num > 10, Cat2 = D}
{Cat2 = C, Cat2 = D}
{Cat1 = A, Cat2 = C, Cat2 = D}
{Cat1 = B, Cat2 = C, Cat2 = D}
{Cat1 = A, Cat1 = B, Cat2 = C, Cat2 = D}
{Num <= 10, Cat2 = C, Cat2 = D}
{Cat1 = A, Num <= 10, Cat2 = C, Cat2 = D}
{Cat1 = B, Num <= 10, Cat2 = C, Cat2 = D}
{Cat1 = A, Cat1 = B, Num <= 10, Cat2 = C, Cat2 = D}
{Num > 10, Cat2 = C, Cat2 = D}
{Cat1 = A, Num > 10, Cat2 = C, Cat2 = D}
{Cat1 = B, Num > 10, Cat2 = C, Cat2 = D}
{Cat1 = A, Cat1 = B, Num > 10, Cat2 = C, Cat2 = D}
{Num <= 10, Num > 10, Cat2 = C, Cat2 = D}
{Cat1 = A, Num <= 10, Num > 10, Cat2 = C, Cat2 = D}
{Cat1 = B, Num <= 10, Num > 10, Cat2 = C, Cat2 = D}
{Cat1 = A, Cat1 = B, Num <= 10, Num > 10, Cat2 = C, Cat2 = D}

Figure 2.3: Fixed structure search space for OPUS_IR with the fictitious database

2.3 Summary

So far, we have explained the concepts and terms related to exploratory rule discovery that are to be frequently used in the thesis. We have also introduced useful terminology of statistics with high relevancy to our research. Previous research on exploratory rule discovery was summarized and classified into two categories, namely, *propositional exploratory rule discovery*, and *distributional-consequent exploratory rule discovery*. The features and differences of these two types of exploratory rule discovery are illuminated. Moreover, techniques of existing exploratory rule discovery and those for dealing with quantitative attributes are reviewed.

We argued that although existing techniques for discovering inter-relationships with quantitative attributes based on the discretize-and-merge paradigm can achieve excellent approximation, information loss is unavoidable. Distributions are asserted to be a better description for quantitative attributes. We introduced the distributional rule discovery which generates rules described using distributions. K-optimal rule discovery is constructed on the basis of OPUS which is an efficient admissible algorithm for unordered search. The merits of using the k-optimal rule discovery were explained, which accounts for our decision on selecting the k-optimal impact rule discovery as the basis of our research.

Finally, formal description of the k-optimal impact rule discovery and the rule discovery algorithm OPUS_IR was introduced.

In the next chapter, the previous research will be discussed and the techniques are considered in two categories: the rule pruning techniques and the fast algorithms for rule discovery. The applicability of techniques developed for propositional rule discovery in distributional-consequent rule discovery will also be discussed.

Chapter 3

Previous Research

As has been mentioned before, different from the traditional data mining techniques, exploratory rule discovery techniques search for all models that satisfies user needs for the purpose of providing alternative models which perform equally or similarly well given a context. It is recognized that two fundamental problems lie with the present-day models of rules (Ng, Lakshmanan, Han and Pang; 1998): (1) Lack of user exploration and guidance. (2) Lack of focus in resulting models. These problems lead to excessive numbers of resulting rules and expensive unnecessary computation and data accesses. Accordingly, research has followed two trends. The first trend addresses how to automatically remove uninteresting resulting models. This means to automatically discard rules which are not interesting or useful in a specific context of application, or are redundant with regard to the set of resulting rules. The second trend intends to reduce unnecessary computation and data access as much as possible. Considerable research has been devoted to both topics.

In this chapter, we first examine the similarities and differences between techniques for distributional-consequent rule and propositional rule pruning and optimizations. The necessities for developing rule pruning techniques specially for distributional-consequent rule discovery are illuminated.

Next, we review existing techniques for rule pruning and optimizations. We consider existing techniques for rule pruning and optimization in two categories. The first category includes techniques that strive towards reducing the number of resulting rules via incorporating constraints specified by the user or derived from the background knowledge. They are commonly known as constraint-based techniques. The second category focuses on how to automatically generate a set of "interesting" rules that are not derivative from other interesting rules in the resulting set. This category is in turn classified into two types: the non-redundant techniques and the productive and statistically productive rule generation. Models removed using the non-redundant techniques can sustain information completeness; while the second class of techniques can induce unavoidable information loss. Nevertheless, such techniques are powerful in many applications. The applicability of the previously mentioned technique in the context of distributional-consequent rule discovery is also discussed.

After reviewing rule pruning techniques, we turn to techniques that address how to improve rule discovery efficiency both before and after rule pruning techniques are applied. A great number of fast exploratory rule discovery algorithms have been developed based on the frequent itemset framework. However, many others have attempted novel search structures for rule discovery other than the frequent itemset lattice.

3.1 Differences in Exploratory Rule Discovery

Rule pruning techniques for propositional rule discovery are extensively studied by researchers. Nevertheless, developing efficient pruning techniques more suitable for distributional-consequent rules is an essential research topic. Since antecedents of both distributional-consequent and propositional rules are composed of conjunctions of Boolean conditions, the techniques for pruning distributional-consequent rules are in a way similar as those for propositional rules. However, the fundamental differences between the descriptive natures of consequents for distributional-consequent rules and propositional rules mean that while some analytic techniques for propositional rules can be modified for use with distributional consequent rues, others cannot.

As a result of distributional-consequent rules being described using distributional statistics of an undiscretized target variable, computation of the required information can induce very large numbers of data accesses which are computationally expensive! Let us compare *association rule discovery* and *quantitative association rule discovery* as was proposed by Aumann and Lindell (1999) which are both developed on the basis of the frequent itemset framework. These approaches undertake the same frequent itemsets generation in the first phase, which is followed by different rule discovering processes. Association rules can be generated with few additional accesses to the records, while quantitative association rule discovery has to go through the database numerous times, in order to derive necessary statistics for rule descriptions. When the amount of data is huge, computational and data access costs can grow to be unmanageable. Hence, the need for developing efficient techniques is pressing.

3.2 Previous Techniques for Rule Pruning

Propositional rule pruning and optimization has been extensively studied. Two basic trends can be found in the literature. The first one is to introduce a set of user-specified criteria, which resulting rules must satisfy to be regarded as interesting. The other trend concentrates on developing algorithms for automatically removing rules that are uninteresting with the presence of other rules. Both classes of techniques are well studied. Applicability of existing propositional rule pruning techniques in distributional-consequent rule discovery is also discussed.

3.2.1 Constraint-based rule discovery

Users are required to specify some constraints on the rules to be found before searching for constraint-based techniques. The constraints we discuss at length in this section are constraints on interestingness measures, on available conditions (or combinations of conditions), or on the patterns to be considered. In this way, we automatically regard rules that do not satisfy any of these constraints as *uninteresting*.

Constraints

A constraint is a predicate of the power set of the set of records covered by a conjunction of conditions (itemset) that composes of the rule body. It is a criterion on the characteristics of a rule or an itemset that must be satisfied to be considered as interesting.

There are various constraints that can be introduced into data mining process at different stages. Figure 3.1 concisely illustrates the relationship of the data mining process and the introduction of different constraints.

In this thesis, only two types of constraint-based rule pruning techniques are to be discussed. The first class of techniques introduces some user-specified constraints on interestingness measures or statistics into rule discovery process. The resulting rules must have a value for a specific interestingness measure greater than a threshold, or in a fixed range. The second class incorporates rule constraints. Items that may or may not appear in a resulting rule, or rule structures that are acceptable

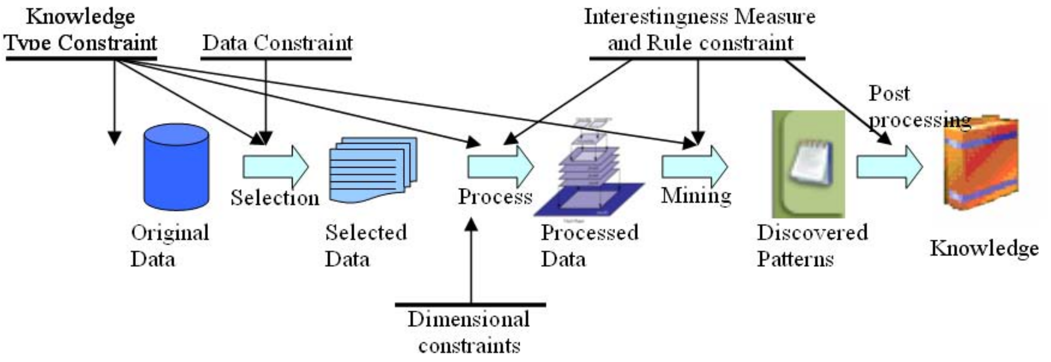


Figure 3.1: Constraints and data mining

or unacceptable are pre-defined before rule discovery. Commonly, item *taxonomies* or *hierarchies* are utilized to achieve better understandability.

Many researchers have investigated how constraints can enhance search space pruning during rule discovery. We classify constraints into two categories according to their properties about whether or not they can be pushed deep into the rule discovery process and still ensures the completeness of resulting solutions. For most constraints, either bounds can be found for the search space be derived for pruning (e.g. anti-monotone and monotone constraints), or all and only those sets of items that are guaranteed to satisfy the constraints can be enumerated (succinct constraints).

The most powerful and widely applied type of constraints for search space pruning are the *anti-monotone (or downward closed)* ones. If the set of records covered by set of conditions does not satisfy an anti-monotone constraint, neither can the coverset of any of its supersets. For a tree-style search space as that for the OPUS algorithm, if a node violates an anti-monotone constraint, the whole branch can be pruned without jeopardizing the completeness of resulting rules. The bound derived from an anti-monotone constraint is *tight*, because no more computation is required to calculate it. For example, a minimum support for resulting rules of 0.1

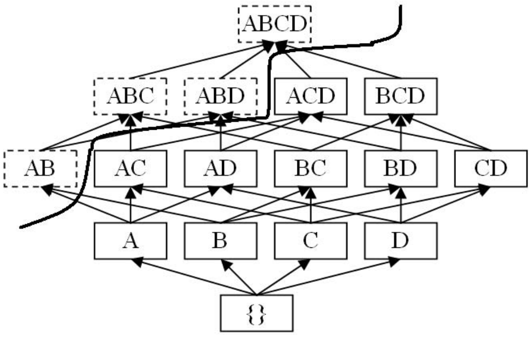


Figure 3.2: Frequent itemset lattice pruning using anti-monotone constraints

is an anti-monotone constraint. In the solution space, anti-monotone constraints make up an upper bound. Using the frequent itemset lattice as an example, in figure 3.2, if node AB has a lower support than 0.1, all the itemsets that are derived from AB should be pruned, because no solutions can be found among them. Anti-monotone constraints were first introduced into association rule discovery by Agrawal and Srikant (1994) and are formally summarized by Ng et al. (1998), who proposed an architecture for supporting constraint-based, human-centered exploratory mining of various kinds of rules.

Scientists have also been studying how to introduce *monotone (or upward closed)* constraints into exploratory rule discovery for search space pruning. Monotone constraints are those that if failed by an itemset will also be failed by all its subsets. Monotone constraints create tight lower bounds in the solution space for a rule discovery task. For example, that a rule must contain a certain item is a monotone constraint. By incorporating monotone constraints into the frequent itemset based algorithms, we are able to save a great deal of data accesses and computation. For example, if A must be included in the antecedent of a desirable rule, no data access or computation is required for assessing the validity of the itemsets in dashed boxes in figure 3.3.

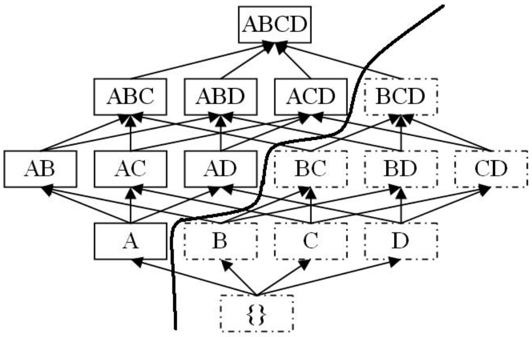


Figure 3.3: Frequent itemset lattice pruning using monotone constraints

A typical technique for combining both anti-monotone constraints and monotone constraints to achieve superior performance in search space pruning was suggested by Grahne et al. (2000) for correlated set discovery. The authors made use of the *CT-support* constraint, which is anti-monotone, and the *correlated* constraint, which is monotone, as well as other user-specified constraints to calculate the bounds of solution space in correlated set discovery. How the synergy between anti-monotone and monotone can be exploited is also studied by Bonchi and Geothals (2004) in a rule discovery algorithm using a *B-tree*.

Loose bounds can be built by using constraints other than the anti-monotone or monotone ones. Further calculations are required before the bounds can be derived. Such properties were studied by Lawler and Wood (1966) and summarized as the *branch and bound* methods.

Another well studied class of constraints is the class of succinct constraints. For constraints with succinctness, we can always name all and only those itemsets that are validate. $Min_J < minval$[1] is an example of succinct constraints, as we are able to enumerate all the records that have a value smaller than *minval*. Only a conjunction of conditions that covers at least one record among these enumerated

[1] Min_J stands for the minimum value for attribute J.

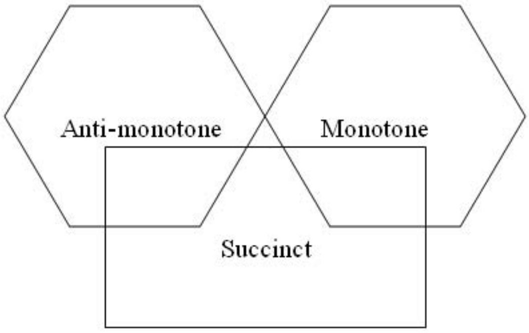


Figure 3.4: Relationship of different constraints (Pei et al.; 2001)

ones can be a solution. Succinct constraints are *pre-counting prunable*, because pruning can be done before accessing the database.

The relationships between the above mentioned types of constraints are described in figure 3.4. This figure show that the class of succinct constraints overlaps with both classes of anti-monotone and monotone constraints.

All the previous discussions were devoted to constraints with only 1-variable. Only one measure is changing in these 1-variable constraints. However, constraints with multiple variables are useful in some contexts of application. For instance, if users are only interested in rules in which the average price of an item is greater than the average of another. We need to evaluate the constraints with two variables. Lakshmanan et al. (1999) studied the properties of 2-variable constraints and argued that the *anti-monotonicity* and *quasi-succinctness* of 2-variable constraints can be introduced into the rule discovery for better rule pruning. They defined that, a 2-variable constraint is anti-monotone if and only if an itemset violates the constraint, then so does all its supersets. Whereas, a 2-variable constraint $C(S,T)$, where S and T are sets of items, is quasi-succinct if it can be reduced to two 1-variable succinct constraints of the form $C_S(S, qc_S)$ and $C_T(T, qc_T)$, where qc_S, qc_T are constants such that the set of all valid S-sets and the set of all valid T-sets are preserved after the reduction.

In this thesis, we restrict ourselves to 1-variable constraints only.

Interestingness Measure Constraints

Rule interestingness measures are metrics against which the interestingness of rules is measured. *Subjective interestingness measures* involve user interpretations. Usually, if a rule is outside users' expectation, it is subjectively "interesting". By contrast, *objective interestingness measures* are statistics or metrics that are derived from data.

One of the central problems in knowledge discovery is to develop good objective measures of interestingness for discovered models (Dong and Li; 1998), which can be used for rule pruning or ranking. The most intensively adopted measures are *support and confidence.*

$$support(A \rightarrow C) = P(A, C)$$

$$confidence(A \rightarrow C) = \frac{P(A, C)}{P(C)}$$

Setting a minimum support for resulting rules is to specify a minimum degree of generality. Rules that are not *general* enough is not interesting enough to attract user considerations. Contrarily, confidence is a measure of predictive ability which present the probability of co-occurrence. *Coverage* is an alternative for support which measures the frequency of records in the databases that satisfy the rule antecedent.

$$coverage(A \rightarrow C) = P(A)$$

However, support and confidence alone is not enough for capturing whether a resulting model is interesting or not. A rule may have very high support and confidence, but is still not surprising. Things go worse when the database is dense and great redundancies exist among the data. We have presented examples of spurious rules with high support in last chapter.

Techniques were proposed to alleviate this problem by mining associations with multiple minimum supports for different items (Liu et al.; 1999a), or by decreasing the value for minimum support as the size of discovered frequent itemsets increases (Seno and Karypis; 2001). An approach using weighted support was also proposed by Tao, Murtagh and Farid (2003) in order to address the uncertainty of support thresholds. Some developed techniques for rule mining without support, examples including the correlation set and implication rule discovery (Brin, Motwani and Silverstein; 1997; Brin, Motwani, Ullman and Tsur; 1997), and jumping emerging patterns (Li, Zhang, Dong, Ramamohanarao and Sun; 1999).

Many other interestingness measures are proposed as complements for support and confidence in exploratory rule discovery. Some of the frequently applied ones are introduced below.

1. **Lift (IBM; 1996):** This measure was also referred to as *interest* by Brin, Motwani and Silverstein (1997). The measure is defined in the following formula:

   $$lift(A \rightarrow C) = \frac{P(A, C)}{P(A)P(C)}$$

   This is a measure for the degree of dependency between rule antecedent and consequent as well as a ratio of confidence over a prior expectation. Lift treats all the conditions in a rule symmetrically. With a lift of 1, the rule antecedent and consequent are independent from each other. A lift greater than 1 implies positive dependency between A and C, while a lift between 0 and 1 implies negative dependency. The further the value for lift deviates from 1, the greater the antecedent and consequent are correlated, either positively or negatively.

2. **Conviction:** Brin, Motwani, Ullman and Tsur (1997) proposed the interestingness measure *conviction* for *implication rule discovery.* Conviction is also a measure of predictive ability. However, it is the measure of actual implication as opposed to confidence which measures co-occurrence only. It is also different from *lift (interest)*, which is essentially a measure of departure from independence. Conviction is defined as follows:

   $$conviction(A \rightarrow C) = \frac{P(A)P(\neg C)}{P(A, \neg C)}.$$

   This interestingness measure is declared to be advantageous over confidence and lift in measuring the predictive ability in many ways. If conviction is 1, the rule antecedent and consequent are completely uncorrelated. A rule that holds all the time (rule confidence is 100%), has an infinite conviction.

3. **Leverage (Piatetsky-Shapiro; 1991):**

   $$leverage(A \rightarrow C) = P(A, C) - P(A)P(C)$$

   This interestingness measure denotes the distance between the observed frequency of A and C and the frequency that is expected if A and C are independent. Setting a minimum leverage means setting a lower bound for support. High leverage implies high support. This property is useful in respect that a rule with low lift but with unexpected high frequency may also be interesting.

4. **Share (Carter et al.; 1997):** This is a measure proposed as an alternative to support. Share measures importance of an itemset in a specific way which is designed in accordance with the features of market basket data. Since transaction data not only contain information of the existence of a certain item but also quantities, costs, profits etc. More insight into an item can be exploited using such information. By introducing share, accurate financial

calculations or comparisons can be done. Item share is defined to be the ratio of actual sum of an attribute for a user-specified share measure (for example, market share) in a certain itemset to the global sum of that attribute.

5. **Collective Strength (Aggarwal and Yu; 1998):** is defined to be the product of actual ratio of good events to bad events and expected ratio of bad events to good events.

   $$collective\_strength(A \rightarrow C) = \frac{P(A,C)}{1 - P(A,C)} \times \frac{expected(1 - P(A,C))}{expected(P(A,C))}$$

   The expected ratio is calculated by assuming independence among items. Items within a rule that are completely positively correlated can generate an infinite collective strength, while being perfectly negatively correlated produces a 0 collective strength. A collective strength of 1 implies independence. This measure is monotone in that if a collective strength of an itemset is v, any subset of this itemset will have a collective strength greater than v.

6. **Improvement:** Bayardo, Jr. et al. (1999) argued that a rule should be regarded as interesting only if it exhibits a minimum improvement in confidence, which is greater than a user-specified value, comparing to its parents. Their approach took the relationship of different rules into account and will be discussed in detail in section 3.2.2.

7. **Other measures:** Dong and Li (1998) proposed an interestingness measure in terms of neighborhood-based unexpectedness. The neighbourhood of an association rule consists of all association rules within a given distance. Two types of unexpectedness are defined, namely, *unexpected confidence* and *isolated*. Zhong et al. (1999) used *peculiarity* to determine the extend to which one data object differs from other similar data objects. Gray and Orlowska

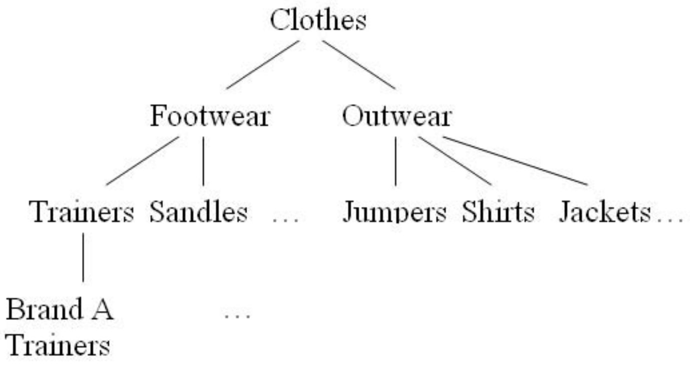


Figure 3.5: A taxonomy for clothes

(1998) defined an interestingness measure to evaluate the strength of association rules which contains a weighted discrimination and a weighted support component. *Laplace preference function* was used to determine the goal of rule searching, providing a conservative estimation of the predictive accuracy of a class description. This approach trade-off accuracy to achieve better generality (Webb; 1995). A summary of interesting measures developed in the context of propositional exploratory rule discovery is provided in table 3.1.

Rule Constraints

Even after interestingness constraints are introduced, the resulting rules may still turn out to be uninteresting in many ways. A rule may correspond to background knowledge, or user expectations, or may contain uninteresting attributes. Rule constraints are specified to address these problems. Taxonomies and hierarchies are often incorporated in constraint-based rule discovery to achieve better understandability.

Taxonomy is a hierarchy in the form of a tree as shown in figure 3.5. In this taxonomy, *clothes* is the root from which the following nodes can be derived. *Brand A trainers* is a leaf. Rule constraints can be specified with any levels of concept.

measure	Definition	Range	Bibliography
support	$P(A,C)$	[0,∞)	Agrawal et al. (1993)
confidence	$\frac{P(A,C)}{P(C)}$	[0,1]	Agrawal et al. (1993)
lift/interest	$\frac{P(A,C)}{P(A)P(C)}$	[0,∞)	Brin, Motwani and Silverstein (1997); IBM (1996)
conviction	$\frac{P(A)P(\neg C)}{P(A,C)}$	[0,∞)	Brin, Motwani, Ullman and Tsur (1997)
leverage	$P(A,C) - P(A)P(C)$	[-0.25,0.25]	Piatetsky-Shapiro (1991)
share	$\frac{Sum_S A(A,C)}{global\_sum}$	-	Carter et al. (1997)
collective strength	$\frac{1-P(A,C)}{1-P(A)P(C)} \times \frac{P(A)P(C)}{P(A,C)}$	[0, ∞)	Aggarwal and Yu (1998)
laplace	$\frac{P(A,C)+1}{P(A)+k^{a}}$	[0,1]	Webb (1995); Roberto J. Bayardo and Agrawal (1999)
ϕ-coefficient	$\frac{P(A,C)-p(A)P(C)}{\sqrt{P(A)P(C)(1-P(C)(1-P(C)))}}$	[-1,1]	Piatetsky-Shapiro (1991)
Odds ratio	$\frac{P(A,C)P(\neg A,\neg C)}{P(A,\neg C)P(\neg A,C)}$	[0,∞)	-
Yule's Q	$\frac{P(A,C)P(\neg A,\neg C)-P(A,\neg C)P(\neg A,C)}{P(A,C)P(\neg A,\neg C)+P(A,\neg C)P(\neg A,C)}$	[-1,1]	-
Yule's Y	$\frac{\sqrt{P(A,C)P(\neg A,\neg C)}-\sqrt{P(A,\neg C)P(\neg A,C)}}{\sqrt{P(A,C)P(\neg A,\neg C)}+\sqrt{P(A,\neg C)P(\neg A,C)}}$	[-1,1]	-
Cohen's kappa	$\frac{P(A,C)+P(\neg A,\neg C)-P(A)P(C)-P(\neg A)P(\neg C)}{1-p(A)P(C)-P(\neg A)P(\neg C)}$	[-1,1]	-
Gini	$1 - [P(C)^2 + P(\neg C)^2] - P(A)[1 - \frac{P(A,C)^2}{P(A)^2} - (1 - \frac{P(A,C)}{P(A)})^2] - P(\neg A)[1 - \frac{P(\neg A,C)^2}{P(\neg A)} - (1 - \frac{P(\neg A,C)}{P(\neg A)})^2]$	-	Morimoto et al. (1998);Roberto J. Bayardo and Agrawal (1999)
Entropy	$-[P(C)\log P(C) + P(\neg C)\log P(\neg C)] - P(A)[conf(A \rightarrow C)\log conf(A \rightarrow C) + (1 - conf(A \rightarrow C))\log 1 - conf(A \rightarrow C)] - P(\neg A)[conf(\neg A \rightarrow C)\log conf(\neg A \rightarrow C) + (1 - conf(\neg A \rightarrow C))\log 1 - conf(\neg A \rightarrow C)]$	-	Roberto J. Bayardo and Agrawal (1999)
Gain	$P(A,C) \times (\frac{P(A,C)}{P(A)} - min\_conf)$	-	Fukuda et al. (1996a)
J-Measure	$P(A)[P(C \mid A)\log_2 \frac{P(C \mid A)}{P(C)} + (1 - P(C \mid Y))\log_2 \frac{1-P(C \mid A)}{1-P(C)}]$	-	Wang et al. (1998)
Cosine similarity	$\frac{P(A,C)}{\sqrt{P(A)P(C)}}$	[0,1]	-
Certainty factor	$max(\frac{P(C \mid A)-P(C)}{1-P(C)}, \frac{P(A \mid C)-P(A)}{1-P(A)})$	[-1,1]	-
Jacard	$\frac{P(A,C)}{P(A)+P(C)-P(A,C)}$	[0,1]	-

Table 3.1: Summary of interestingness measures

[a] The k is an integer number greater than 1.

Users may only be interested in rules that contain certain items or contain children of a specific node in a hierarchy. A typical method for combining item constraints in rule discovery was suggested by Srikant et al. (1997). They argued that such rule constraints enable effective pruning of search space, because of the anti-monotonicity of item constraints.

However, mining rules with taxonomies brings several practical problems (Srikant and Agrawal; 1997). To discover rules at higher concept levels generates rules with generally high support which often happen to be within user expectations. Contrarily, rules at primitive concept levels often have a low support but are more specific or concrete. Such rules are interesting for the users even though they are "weak". This makes it harder to specify a minimum support for all the resulting rules in a task.

Han and Fu (1995) proposed a rule-constraint-based technique using multiple support thresholds. They gradually reduce the minimum support as the concept level becomes lower, until reaching the primitive level. Srikant and Agrawal (1997) tried to mine rule with *one* taxonomy which is interesting only when its support or confidence is more than R times the expected value. They define the expected value for itemset A given its parent C in the taxonomy to be:

$$expected_C(Pc(A)) = \frac{Pc(a_1)}{Pc(c_1)} \times \frac{Pc(a_2)}{Pc(c_2)} \times ... \times \frac{Pc(a_n)}{Pc(c_n)} \times Pc(C)$$

where $a_1 ... a_n$ and $c_1 ... c_n$ are elements in A and C respectively. Graaf, Kosters and Witteman (2000) proposed a similar scheme as that suggested by Srikant and Agrawal (1997) for incorporating several taxonomies in a same task. Their definition of expected value differs from the above definition.

Several papers have also been contributed to introduce rule templates during the course of rule discovery. Rule templates enable easy descriptions of interesting rules structures (Klemettinen et al.; 1994). Acceptable or unacceptable patterns should be nominated by the users beforehand, so as to "focus" the users' analysis and exploration efforts only on the set of rules that is of a specific interest to them. One of the important issues to be addressed for these techniques is how to present the discovered patterns in an understandable manner both for the users and the machine. They proposed algorithms that use Boolean expressions to filter uninteresting rules during rule discovery. Such expressions are disjunctions of conjunctions of predicates representing the acceptable status of items. Moreover, Baralis and Psaila (1997) introduced predefined templates as a means to capture the user specifications for rule mining processes. They proposed a general language to design templates for the extraction of arbitrary association rule types as apposed to Klemettinen et al. (1994)'s definition. Their rule templates go in two classes: the *inclusive templates* for the rules that are of most interest and the *restrictive templates* for those that are known uninteresting beforehand.

Meo et al. (1996) designed a SQL-like operator `MINE RULE` for mining rules, combining the interestingness constraints, the taxonomies and the rule constraints. Their approach enables a universal description of the exploratory rule discovery problem.

A notable feature of the previously mentioned techniques is that they can efficiently incorporate background knowledge for guidance and control of the mining process. However, the resulting rules found using these techniques may turn out to be incomplete, and is less surprising, which is often the case when incorporating subjective information. Rules found are often those users expect to exist or to not exist.

Applicability in Impact Rule Discovery

Constraint-based techniques can effectively reduce the number of resulting rules, circumventing the dilemma that data mining itself may generate too much information to be analysed. Constraint-based techniques for propositional rule discovery using interestingness measures are applicable in impact rule discovery. However, interestingness measures that can be used for impact rule discovery are different from propositional rule discovery. Since the consequent of an impact rule is described using distributions instead of Boolean predicates, many of the interestingness measures devised for propositional rules, which are concerned only about the presence of both the antecedent and the consequent of the rules are not applicable for impact rule discovery. For instance, *support* of the propositional rules cannot be used as a measure of generality for impact rules, because the consequent of an impact rule do not have *support*. Hence, *coverage*, which is the *support* of rule the antecedent, is adopted instead.

Considering mining impact rules from the fictitious database in table 2.2, if we are interested only in rules for which the target means are greater than 3. The following rule:

```
10<num & cat2=D → target (coverage:3 mean:2.6 variance:  5.88
min:0 max:4.8 sum:7.8 impact:-7.77)
```

should be removed as uninteresting.

Distributional statistics are preferred for describing distributional-consequent rules. Webb (2001) also proposed a measure called *impact* for describing interestingness of impact rules.

Constraints on items and rule templates can also be directly introduced into impact rule discovery with minute adaptation. For instance, the users are aware

```
Algorithm:  OPUS_IR(Current, Available, $\mathcal{M}$)
  1. SoFar := $\emptyset$
  2. FOR EACH P in Available
      2.1 New := Current $\cup$ P
      2.2 IF current rule $New \rightarrow target$ does not satisfy any of the
          prunable constraints in $\mathcal{M}$
              THEN go to step 2.
      2.3 END IF
      2.4 IF current rule $New \rightarrow target$ satisfies all the nonprunable
          constraints in $\mathcal{M}$
              Record New $\rightarrow$ target in the rule_list;
      2.5 END IF
      2.6 OPUS_IR(New, SoFar, $(M)$);
      2.7 SoFar := SoFar $\cup$ P
  3. END FOR
```

Table 3.2: The OPUS_IR algorithm with constraints

that if the value of `Num` is determined by the value of `Cat1`, and `Cat2`, we can specify that rules with combinations of these 3 attributes should not be considered as interesting.

The OPUS_IR algorithm that we designed to introduce different constraints during rule discovery is described in table 3.2. This algorithm can efficiently search through a tree-style search space for potential impact rules and is designed on the basis the OPUS algorithm (Webb; 1995).

3.2.2 Compact Representations of Resulting Rules

An objective criterion has to be specified by users subjectively before the constraint-based techniques can be applied. There is another class of rule pruning techniques for which little user guidance and analysis before or during rule discovery is required. In such approaches, resulting rules are compared with each other so as to decide whether they are "redundant" or not. Thus, a more compact statement for the discovered information is provided.

Association rule discovery, which is a typical type of exploratory rule discovery, was initially developed for market basket data, which is generally sparse. However, with more and more research devoted to exploratory rule discovery, attempts were made to extend the application of exploratory rule discovery to dense and large databases like the census databases in Blake and Merz (1998). Too many resulting rules becomes a problem of concern. In the majority of cases, real-life database may generate thousands of "strong" rules among which numerous are redundant or uninteresting. A more compact manner for presenting and summarizing rules are necessary for users to more efficiently analyzed and made use of the discovered knowledge.

Existing rule pruning techniques that belong to this class are the discovery of *maximal frequent itemsets*, the *non-redundant techniques*, and the *productive and statistical significant rule discovery.*

Mining Maximal Frequent Itemsets

Mining maximal frequent itesmets, which are also referred to as *the most specific sentences* (Gunopulos et al.; 1997), is one of the frequently adopted methods to tackle the problem of too many resulting rules. In such techniques, implicit knowledge in the database is expressed using only a set of maximal frequent itemsets, whose size is orders of magnitude smaller than the size of all frequent itemsets. A *maximal frequent itemset* is defined to be the most specific set of frequent itemset being no subset of other frequent itemsets. It is asserted that in many situations, to know only the set of maximal frequent itemsets is sufficient, from which all other frequent candidates are implied according to the anti-monotonicity of the support constraint. The set of maximal frequent itemsets can implicitly and concisely represent the discovered knowledge.

Representative research in the literature includes the *MaxMiner* (Bayardo; 1998) which implemented a heuristic search for frequent candidates, as soon as possible, right after all the subsets of a potential candidate are identified as frequent. Thus linear scalability with the size of longest maximal frequent itemset is achieved. By contrast, Apriori undertakes the generation of frequent itemset of $k + 1$ items only after all candidates of size k are found.

Pincer-Search (Lin and Kedem; 1998) is also one of the famous algorithms for mining maximal frequent itemsets, which is an NP-hard search space reduction algorithm. The authors integrated both bottom-up and top-down searches for mining maximal frequent itemsets. Gunopulos et al. (1997) also proposed a randomized algorithm for computing sets of most specific sentences in relational database with binary data.

Although maximal frequent itemset approaches can generate a set of candidates whose size is even smaller than that yielded by the closed itemset techniques, which are to be discussed next, rules cannot be generated without further data accesses on the basis of discovered maximal frequent itemsets. Even if further browsing of database is performed to derive necessary descriptions for resulting rules, no rule redundancies can be eliminated in this way.

Non-Redundant Rule Discovery

It is a fact that different degree of redundancies exist in resulting set of rules discovered using exploratory rule discovery. For example, a rule might convey exactly the same information as another one which is more general. We called this rule a redundant rule. By identifying and removing redundant rules that convey no further information with the presence of others, a more compact set of rules or frequent itemsets can be generated. *Closed sets, rule covers* and the *trivial rule filtering* are all techniques for discarding such redundant rules.

The closed set techniques have attracted a great deal of attention in propositional exploratory rue discovery. An itemset I is *closed* if and only if there exists no itemset I' where $I \subset I'$ and $coverset(I) = coverset(I')$. Many algorithms are developed targeting at implementing frequent closed set discovery and generation of a compact set of resulting rules with discovered frequent closed sets. Typical examples of closed set related algorithms are *A-Closed* by Pasquier et al. (1999b), *Apriori-Closed* by Pasquier, Bastide, Taouil and Lakhal (1999a), *Closet* by Pei et al. (2000), and *CHARM* by Zaki and Hsiao (1999).

Closed set related techniques generally undergo a two step process. First, *Galois closure operators* or *Galois connections* are adopted for the closed set generation, which successfully maintain the completeness of information. The number of resulting closed sets is exponentially smaller than that of the traditional frequent items sets. Efficiency is improved by some using the *Galois connection lattice* pruning. Second, different schemes are used to generate non-redundant rules. Apriori-Closure (Pasquier et al.; 1999b) discovers *Duquenne-Guigues* Basis for Exact Rules (rules with confidence = 1) and the *proper and structural bases* for approximate rules (rules whose confidence is less than 1). Bastide, Pasquier, Taouil, Stumme and Lakhal (2000) suggested two new bases for association rules whose union is a generating set for all valid models. The bases are composed of closed itemsets and their generators. However, their approach mainly concentrates on the discovery of frequent closed itemsets instead on rules discovery. CHARM (Zaki; 2000) used Galois lattice of concepts and frequent closed tidsets (sets of transaction IDs) to generate non-redundant association rules.

Closed itemset related techniques prune rules in a similar manner as is described in the following example:

Example 1 *Consider the following exact rules*

$$A \Rightarrow B$$

$$B \Rightarrow C$$

$$A \Rightarrow C$$

The third rule can be discarded without sacrificing completeness, for it can be deduced from the first two.

For approximate rules, if rule

$$A \rightarrow C$$

and

$$A' \rightarrow C'$$

have the same support and confidence, and $A \subset A'$ *or (and)* $C' \subseteq C$ *then the second rule is regarded as uninteresting.*

Non-redundant rules generated using the closed set techniques are defined to be the most general rules that are not implied by any of their parents.

Toivonen, Klemettinen, Ronkainen, Hgtijnen and Mannila (1995) proposed a domain independent method for reducing the number of resulting rules without information loss. They defined a *rule cover* as a subset of all rules that, for every record, if there is an applicable rule in the original rule set there must be a rule in the discovered rule cover that is applicable for this record.

$$\cup_{r \in allrule} coverset(r) = \cup_{r \in rulecover} coverset(r)$$

In their approach, after the rule cover is generated, a clustering technique is applied to group the rules in the rule cover according to the distance between rules. However, for some applications a minimal set of rules may not be sufficient. Although it is theoretically possible to infer all interesting rules from the minimal rule set, the user might not be able to identify the most interesting rules in a straightforward manner because they are not explicitly presented.

Webb and Zhang (2002) defined *trivial rules* as redundant rules whose coverage is the same as the coverage of any subset of their antecedents. They proposed a novel technique for efficiently discarding trivial rules with fixed consequents during k-optimal rule discovery based on the OPUS search algorithm. In this way, redundant rules that can be eliminated using closed set techniques can be removed without generating frequent closed itemsets.

Rule Improvements and Significance

Most exploratory rule discovery techniques seek rules $A \to C$ for which there is a correlation between the antecedent A and the consequent C. However, whenever one such rule is found, there is a risk that many derivative and potentially uninteresting rules $A' \to C'$ will also be found. These rules are those for which there is a correlation between the antecedent and the consequent only by virtue of there being a correlation between A and C. For example, if A and C are correlated then for any term B that is unrelated to either A or C, A & B will also turn out to be correlated with C. The rules pruned using the *non-redundant techniques* are a special form of derivative rules.

Rule pruning techniques (or filters) concerning the improvements and significance of resulting rules examine the relations between rules, and remove those within expectation (or without enough surprisingness). The results suffer from information loss. However, in some applications, it is worth trading off information completeness against efficiency in post-discovery processing.

Confidence Improvement and Unproductive Rules: Bayardo, Jr. et al. (1999) defined a *minimum improvement* in confidence that a rule must exhibit in order to be consdered interesting. The minimum confidence improvement used by them is defined as below:

$$imp(A \rightarrow C) = argmin_{A' \subset A}(confidence(A \rightarrow C) - confidence(A' \rightarrow C))$$

They argued that a minimum improvement greater than or equal to 0 is a desirable constraint in most applications of association rule mining. Webb (2003) referred to the rules with a *minimum improvement* greater than 0 as *productive rules*. The non-redundant techniques are able to prune some of the rules that have 0 improvement compared with any of its parents', accepting those with either negative or positive improvement as non-redundant.

Since a minimum improvement is an objective measure of interestingness, it is hard for users to subjectively decide on a proper minimum improvement for all the rules, which is also a potential problem with other objective measures. Setting the minimum improvement too high leads to discarding of rules that are actually interesting; while setting it too low retains potentially uninteresting rules. Truth has that by performing exploratory rule discovery, users are trying to retrieve implicit knowledge within a population with reference to a sample, which is only a subset of that population. Sampling inevitably produces data fluctuations. Hence, it happens that a rule with a desirable minimum improvement may actually be uninteresting in terms of the population.

Statistically Unproductive Rules: We start reviewing techniques regarding statistically unproductive rules with the following example.

Example 2 *Suppose the following propositional rules are generated by a rule discovery systems:*

$$A \rightarrow C[support = 60\%, confidence = 90\%]$$

$$A\&B \rightarrow C[support = 45\%, confidence = 91\%]$$

$$A\&D \rightarrow C[support = 46\%, confidence = 70\%]$$

There is a strong possibility that the conditions A and B are independent and the second rule conveys little interesting information given the first one. For many rule discovery tasks, users are interested only in rules whose antecedents and consequents are positively related. Therefore, the third rule is regarded as statistically unproductive (or insignificant as is referred to by many other researchers (Bay and Pazzani; 2001; Liu et al.; 1999b; Aumann and Lindell; 1999)) and should also be discarded, because the condition D is negatively correlated to condition A concerning the consequent C.

Statistical tests are applied to reduce the influence of sampling on resulting rules. The Chi-square test is widely employed for testing independence with Boolean conditions. Liu et al. (1999b) did research on association rules with fixed consequents. They used a chi-square test to assess whether the antecedent of a rule is independent from its consequent, accepting only rules whose antecedent and consequent are positively correlated. Hence, rules that happen to appear "interesting" by chance can be discarded in this way. The rules that are identified as productive but are discarded by using a statistical test with a given significance level, are referred to as *statistically unproductive rules.*

The chi-square test is also used in other contexts including contrast set discovery (Bay and Pazzani; 2001) for identifying significance of discovered contrast sets.

However, the chi-square test is not suitable for small samples and as the number of rules increases so does the risk of type-1 error (discarding rules which are actually significant). Webb (2005) proposed a statistically sound technique for filtering derivative extended rules, using the Fisher exact test and a hold out set. Webb's technique successfully controls errors induced by multiple comparisons.

Applicability in Distributional-Consequent Rule Discovery

The maximal frequent itemset approaches can be introduced to the distributional-consequent rule discovery directly, however, the resulting set of most specific distributional-consequent rules can not successfully provide information that are sought by the users who choose distributional-consequent rules over the propositional rules.

Galois connection is only applicable to qualitative attributes as far as we know. It is not applicable with distributional-consequent rule discovery theoretically. A naive adaption of the closed set and rule cover techniques for distributional-consequent rule discovery would be impossible (Webb; 2005). However, the trivial rule filter proposed by Webb and Zhang (2002) can be modified for pruning distributional-consequent rules. We discuss details of how this can be implemented chapter 4.

Setting a minimum improvement for distributional statistics for resulting impact rules is not desirable. The interactions regarding undiscretized quantitative variable are more subtle and even harder to capture using only thresholds of objective interestingness measures. Statistical tests should be applied. Aumann and Lindell (1999) proposed a technique for removing insignificant quantitative association rules using a standard z test based on the frequent itemset framework. However, their approach is not optimal for rule discovery in very large, dense databases, since their approach was based on frequent itemset generation, which requires excessive computation for maintaining candidates in memory during rule discovery.

3.3 Efficiency Consideration

Originally developed in the context of market basket data, which is generally sparse with relatively smaller data volume, the traditional frequent itemset techniques are computationally infeasible to discover rules in very dense real world databases. Two drawbacks are identified with the traditional frequent itemset approaches. First, it is too expensive to handle huge numbers of candidate itemsets generated during rule discovery, due to the prohibitive maintenance overhead. Second, it is tedious to repeatedly access the database and check the candidates by pattern matching which is especially true for mining long patterns (Han, Pei and Yin; 2000). Due to the features of rules discovered using exploratory rule discovery, which we have mentioned before, rapid progresses have found their ways in real world exploratory rule applications. It is necessary to design algorithms for rule discovery in databases with huge redundancies, such as census databases. How to improve the efficiency of exploratory rule discovery becomes a critical issue of concern.

Some researchers tried to develop efficient algorithms on the basis of the frequent itemset framework, while others strived towards novel implementations using different rule discovery frameworks. In this section, we provide a brief review of existing fast algorithms for exploratory rule discovery and highlight the efficiency problem of distributional-rule discovery which is even worse comparing with that of propositional rule discovery.

3.3.1 Efficient Improvements with the Frequent Itemsets Framework

The frequent itemset approaches, a typical example of which is the Apriori algorithm, normally undertake two phases: 1. The candidate generation phase in which frequent itemsets are generated by accessing the database numerous times, 2. In the second phase, rules are derived from the discovered candidates, meanwhile, support and confidence are counted by further passes through the database. With respect to these two phases, different efficient techniques are studied. Since the first candidate generation phase is much more computationally expensive than the rule generation phase, more research is devoted to improve the frequent itemset generation efficiency.

Examples of fast algorithms based on the frequent itemset generation are summarized in the following list:

1. **Efficiency improvement using search space and rule pruning:** Most of the pruning techniques reviewed in the previous section can help to achieve better efficiency for the candidate generation process. Agrawal and Srikant (1994) were the first to use the pruning trick with anti-monotone constraints. Closed set related techniques only access a suborder of the original frequent itemset lattice and thus the running time is reduced.

2. **Vertical database layout:** Traditionally, a *horizontal database layout* is utilized for organizing data. In the horizontal layout, data is arranged as a set of rows, representing records, which consist of identification numbers, called TID (Transaction Identification), and sets of values for attributes. When such a structure is used, extra computational overheads are required for searching, maintaining, and computing of candidates. The entire database has to be accessed even if only a small subset of the records are useful. The *vertical*

database layout is designed to circumvent these problems. In the vertical database layout, conditions (attributes taking a value or a range of values) are followed by a list of TIDs. All necessary information for rule discovery is contained in the vertical layout which can speed up the rule discovery process (Zaki, Parthasarathy, Ogihara and Li; 1997a). Examples of algorithms using a vertical database layout for enhancing rule discovery efficiency are *VIPER* proposed by Shenoy et al. (2000), *Eclat* by Zaki, Parthasarathy and Li (1997) and *MaxEclat* and *MaxClique* by Zaki, Parthasarathy, Ogihara and Li (1997a).

3. **Parallel algorithms:** Parallel algorithms for exploratory rule discovery are extensively studied. Agrawal and Shafer (1996) first proposed three parallel rule mining algorithms on shared-nothing multi-processors. These algorithms explored different extents of tradeoff between communication and memory costs to achieve an optimal balancing point. The *count distribution* algorithm turned out to be the best of all, and delivered far superior performance to previous work. Zaki, Parthasarathy, Ogihara and Li (1997b) proposed the CCPD algorithm for association rule mining in shared-memory multi-processors. In CCPD, frequent itemsets are parallelly generated into a hash structure shared among different processors. Many additional optimizations are introduced to enhance the performance. Zaki, Parthasarathy and Li (1997) proposed the algorithm, *Eclat*, for efficiently parallel mining of association rules by clustering frequent itemsets into *equivalences classes* and then distribute them to different processors, reducing the database scans to at most three. Park et al. (1995b) suggested a parallel algorithm PDM as an extended study for the hash-based rule discovery algorithm DHP (Park et al.; 1995a).

4. **Sampling and partitioning :** These techniques are for minimizing the I/O expenses during rule discovery by reducing the size of data to be processed at a time. For sampling techniques, only a small subset of the database is drawn for rule mining. Since the volume of sample data is small enough to be stored in the memory for processing, rules can be found using as few scans as possible. Toivonen (1996) suggested a two-phase rule discovery algorithm in which rules are discovered in one data pass using sample data. However, sampling may lead to data skew and the resulting rules found may suffer from inaccuracy. The authors verified the rules found from the sample data using the rest of the database after rule discovery, so that rules missing from the sample can be identified. This algorithm requires at least one pass through database and two passes in the worst case. Savasere et al. (1995) used the algorithm *partition* to minimize the number of necessary scans of database, resulting in the same number of scans as that for the sampling algorithm by Toivonen (1996). At most one scan is required for candidate generation and another is for counting support. In this algorithm, the databases are divided into small partitions, and local frequent itemsets are found which can then be used to derive the global frequent itemsets and rules. Since no information is shared among different partitions, this algorithm can be run on parallel systems. *Partition* also suffers from data skew. *SPINC*, which is proposed by Mueller (1995), can reduce the maximum number of scans to only $\frac{2n-1}{n}$. Lin and Dunham (1998) also proposed an anti-skew algorithm which performs as well as SPINC.

5. **Other techniques:** L-Gen, which was proposed by Yip et al. (1999) minimizes the I/O costs of frequent-itemset-based exploratory rule mining by

generating candidates of multiple sizes (instead of one), based on lattice theory, during each database scan. Such a structure provides scopes for the algorithm to make use of prior knowledge collected during previous scans to prune un-useful candidates in early stage. In the best case, only two scans of database are necessary. Brin, Motwani, Ullman and Tsur (1997) proposed *DIC (Dynamic Itemset Counting)* for fast generation of candidates. DIC performs an eager search for frequent itemsets. Candidates are generated as soon as all its subsets are known to cover more records than the minimum requirement. Usually two scans are necessary for discovering candidates using homogenous data. A *hash-based* algorithm DHP (Direct Hashing and Pruning) is studied by Park et al. (1995a). The authors showed that a hash technique can be used to accelerate the candidate generation of 2-itemset. However, later studies showed that a hash structure can add to the overheads and slow the discovery efficiency down in later iterations for bigger itemsets.

Other Efficient Implementations

Frequent itemset based techniques, most of which employ a generate-and-test paradigm, play a crucial part in the context of exploratory discovery. However, since Apriori-based techniques utilize a level-wise breadth-first manner for lattice transverse, a great amount of computation is required for memory and data maintenance. A minimum support constraint is usually adopted to prune the NP-hard search space. The number of iterations required for itemset generation depends on the size of longest frequent itemsets. computational expenses are even more challenging on occasions where the size of longest patterns to be found is very large or the support threshold is very low.

Set enumeration approaches (Agarwal et al.; 2001) employed the set enumeration structure to discovery association rules, To reduce the redundancies in database access and candidate maintenance costs. A lexicographic tree of itemsets is constructed successively to generate the candidates. Such a structure makes sure that any itemset will only be assessed once during the rule discovery process. After the frequent itemsets are generated, support and confidence are counted against a metric structure in Agarwal et al. (2001)'s proposal. They proposed algorithms for breadth-first, depth-first as well as a combined approach to iterate through the search space.

One of the efficient depth-first set enumeration algorithms for rule discovery is the *OPUS* based algorithm for k-optimal rule discovery (Webb; 1995). They introduce branch and bound techniques for effective search spaces pruning and thus improve the efficiency of rule discovery.

Depth-first algorithms are also introduced with other techniques for efficiency improvement. Agarwal et al. (2000) and Burdick et al. (2001) both used a depth-first structure to efficiently mine maximal frequent itemsets.

Bay and Pazzani (2001) used a breadth-first set-enumeration approach for contrast set discovery. Bayardo (1998)'s algorithm for mining maximal frequent itemsets is also constructed on set enumeration.

Another algorithm based on tree formulation was designed by Han et al. (2000). They proposed the FP-Growth (Frequent Pattern growth) algorithm which employed a novel, compact data structure called the FP-Tree. This is an extended prefix tree structure bearing all necessary information for frequent pattern generation. A pattern fragment growth method is adopted to reduce the expenses on candidate generation and maintenance, after which a partition-based method is utilized to reduce the search space size. Then, a recursive divide-and-conquer technique is applied for mining frequent itemsets. Their approach is efficient and scalable, leading to dramatic reduction in running time compared with the frequent itemset based algorithms.

Bonchi and Geothals (2004) and Bonchi, Giannotti, Mazzanti and Pedreschi (2003) proposed techniques for integrating anti-monotone and monotone constraints on the basis of extended FP-Growth algorithms.

3.4 Summary

In this chapter, we have extensively reviewed previous techniques for exploratory rule pruning and optimizations, as well as techniques for fast rule discovery. We first identified the drawbacks inherent in exploratory rule discovery and highlighted the necessities for developing rule pruning algorithms. We then demonstrated the fact that techniques for propositional and distributional-consequent rule discovery are different in many ways. However, it is noticeable that although extensive studies have been devoted to pruning propositional rules, comparatively little has been done in the context of distributional-consequent rule discovery. Techniques that are designed specifically for propositional rule discovery are not all directly applicable in distributional-consequent rule discovery. We summarized the work related to propositional rule discovery which comes in two categories.

The first category includes the constraint-based rule discovery techniques in which the users are required to define a concrete set of criteria, called constraints, either on interestingness measures or on rule structures. Although this helps to accomplish superior performance in rule discovery, introducing user guidance in this way brings some disadvantages. It is hard to subjectively specify a proper threshold or range for an objective interestingness measure and ensure that all the “interesting” rules remain. Setting constraints for rule structures can result in reduction of overall interestingness of resulting rules in some applications, since many of the resulting rules happen to coincide with users' background knowledge.

The second category of rule pruning technique focuses dominantly on how to remove rules that are potentially uninteresting because it can somehow be "derived" from other rules. Three representational subclasses of techniques are reviewed, including the maximal frequent itemsets discovery, the non-redundant rule discovery and the productive and statistically significant (unproductive) rule discovery.

Applicability of the techniques with propositional rule discovery in the context of distributional-consequent rule discovery was discussed and possible extensions and adaptions were suggested.

Finally, we examined previous research regarding efficient discovery of rules. Many fast algorithms are developed on the basis of frequent itemset generation: generating all candidates and then test their interestingness after which rules are discovered. Efforts were delivered primarily to how to reduce the I/O costs. While others suggested novel approaches for rule discovery using frameworks other than frequent itemset generation, examples are the set-enumeration and the FP-Growth algorithms.

In the next chapter, we are going to propose several techniques for improving interesting impact rule discovery efficiency. Another type of derivative rules that has not been studied by previous research is also defined and a new algorithm is designed for removing such rules.

Chapter 4

Effective Impact Rule Pruning

In previous chapters, we have reviewed existing research regarding exploratory rule discovery, classifying them into two main classes: propositional rule discovery and the distributional-consequent rule discovery. We have also given a survey for different techniques for rule pruning and rule discovery. It has been discussed that research in the area of distributional-consequent rule discovery is limited. However, efficiency problems in distributional-consequent rule discovery are more remarkable since extra computational and I/O costs are required for collecting necessary distributional statistics for describing resulting rules. Furthermore, the problem of too many resulting rules also exist with distributional-consequent rule discovery. Since previous techniques are mainly designed for propositional rule discovery, there is an urge for developing fast and effective rule pruning techniques for distributional-consequent rule discovery.

As far as we know, the technique for removing quantitative association rules proposed by Aumann and Lindell (1999) is one of the few approaches for distributional-consequent rule pruning. However, their approach is not optimal for rule discovery in very large, dense databases, since it was devised based on the Apriori algorithm. When working on very dense databases, Apriori requires prohibitively expensive computational costs and memory storage for storing and maintaining the candidates, for the number of candidates generated during rule discovery can be unwieldy.

In this chapter, we argue that existing rule pruning techniques are not sufficient for removing all the potentially uninteresting rules that can be theoretically identified. We propose the definition of *derivative partial impact rules*, and analyze their differences from the previously identified uninteresting distributional-consequent rules. We also explain the relationship among different kinds of rules. We propose an efficient implementation for pruning derivative partial rules and *derivative extended rules*, which is similar to the *insignificant* quantitative association rules proposed by Aumann and Lindell (1999). After this, three techniques for improving the efficiency of impact rule pruning are proposed. These are the *triviality filter*, which acts as an alternative to as well as a complement for the derivative extended rule filters; the *difference set statistic derivation* approach, which aims at reducing the data access redundancies during rule generation, and the *circular intersection approach* which improves the efficiency by eliminating redundant intersection operations.

4.1 Derivative Impact Rules

It has been repeatedly mentioned in this thesis that the problem of too many resulting rules is among the typical problems of exploratory rule discovery. Identifying and removing potentially uninteresting or spurious rules has always been a focus of data mining research. In section 3.2.2, we have reviewed the techniques for removing rules that are uninteresting because of the existence of some or one of their generalizations. Examples are the *closed set related techniques*, the techniques for removing *trivial rules*, and *unproductive rules.* We applied the term *derivative extended rules* to describe the uninteresting rules removed using such techniques. The set of derivative extended rules is only a subset of *derivative rules.* We have also given the description of *derivative rules*, which, just as the name implies, *derivative rules* are those which convey redundant information that can somehow be derived from other rules. Rules that are not "derivative" from any other rules are referred to as *fundamental rules.* In this section, we are going to propose two derivative rule filters, and explain how these filters can effectively reduce the number of resulting rules.

4.1.1 Derivative Extended Rules

We applied the constraint based OPUS_IR in table 3.2 to the fictitious database in table 2.2 with the minimum coverage set to 0.2, 19 rules are generated as outcome as shown in figure 4.1. In this figure, the italic nodes are those identified as "strong" with a coverage over 0.2, the × nodes correspond to the nodes or branches that are pruned without additional data accesses, according to the anti-monotonicity of the minimum coverage constraint; while others are nodes that are found "uninteresting" after accessing the data. As the size and density of database on which OPUS_IR is run increase, the number of resulting rules increases exponentially.

Derivative Extended Rule Pruning

Aumann and Lindell (1999) introduced insignificant quantitative association rule pruning. They defined a rule with a significantly different mean from all its parents' as *significant (desired).* An uninteresting impact rule whose mean is not significantly improved comparing with any of its parents' may happen to be interesting by chance due to sampling fluctuations. The seemingly interesting information conveyed by such rules is implied by their fundamental counterparts. In other words, these rules are derivative comparing with its parents (ancestors). Using Aumann and Lindell's definition, many rules whose performance is not significantly improved

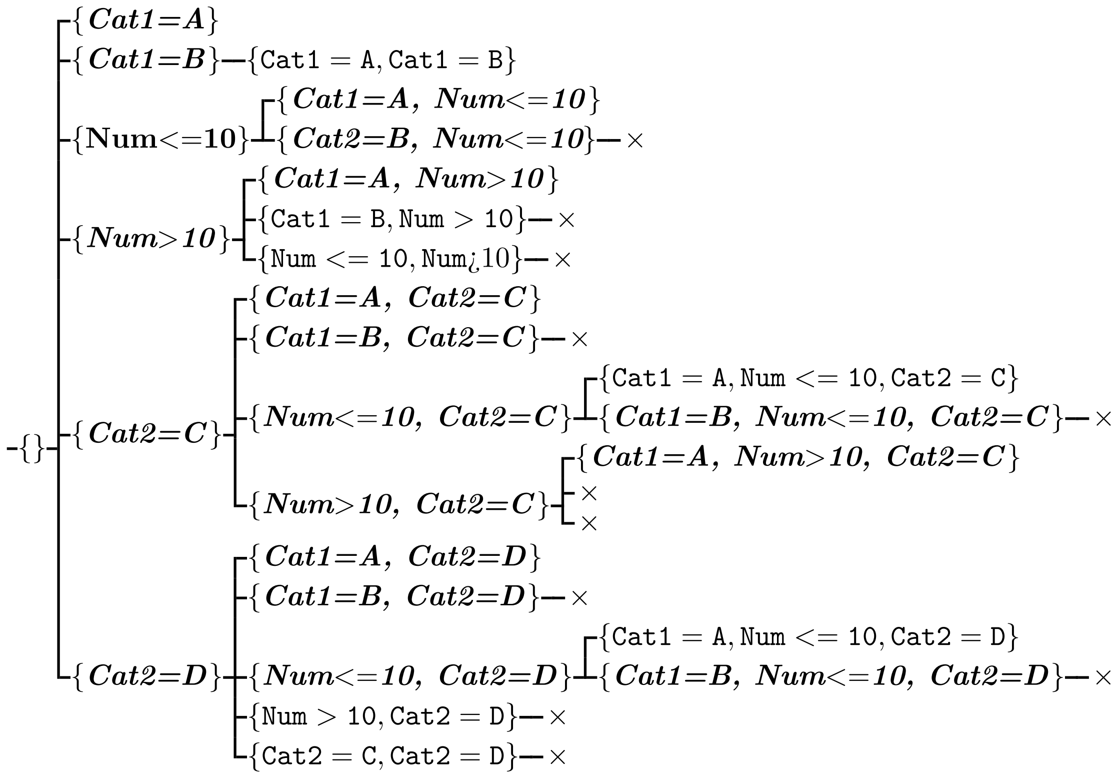


Figure 4.1: Search space for OPUS_IR_Filter, with minimum coverage 0.2

in comparison with their parents, are found. However, as far as we are concerned, these should be discarded in many contexts of application. For example, if the target mean of A & $B \rightarrow profit$ is worse than that of its parents': $A \rightarrow profit$, it follows that the condition B is negatively related to *profit* given condition A, and can only reduce the resulting profit. Thus, rule A & $B \rightarrow profit$ is of little interest to the users. Such rules are what we call: *derivative extended rules.*

Herewith, we present our definition of derivative extended impact rule as follows:

Definition 3 *An impact rule $A \rightarrow target$ is a* derivative extended rule *if the distribution of its target is not significantly improved in comparison with any of the target distribution of rule $A' \rightarrow target$, where $A' \subset A$ and $|A'| = |A| - 1$.*

$$DeriExtended(A \rightarrow target) =$$

$$\exists x \in A, dist(A \rightarrow target) \not\gg dist(A - x \rightarrow target)$$

A rule is significant *if it is not* derivative extended.

Unproductive rules and the statistically unproductive rules, which were introduced in chapter 3, are all derivative extended rules.

As far as we are concerned, rules whose performances cannot be understood through their parents, yet can be predicted using any of their grandparents' are still interesting. The following example exhibits our reasons.

Example 3 *Let us look at the following rules,*

$$District = A \rightarrow profit(mean = 1000)$$

$$District = A \ \& \ age > 50 \rightarrow profit(mean = 500)$$

$$District = A \ \& \ age > 50 \ \& \ professor = Engineer \rightarrow profit(mean = 1000)$$

The third rule is interesting, because with these resulting rules, it is obvious that high profits are produced by people in district A who are engineers above the age of 50, however, those are over 50 years of age but are not engineers do not yield desirable profits. The third rules can make decision makers' attention more focused without losing opportunities for profit improving.

The most important issue of implementing the derivative extended rule filter is how exactly the term *significantly improved* is defined. We assume a context where the users seek impact rules that maximize a certain measure of interestingness, such as *mean.* Equivalent techniques for minimization can be derived from our technique in a straightforward manner. In this thesis, we restrict ourselves to regard that if a distribution $dist_a$ has a mean significantly more desirable than that of $dist_b$ at a given significance level, $dist_a$ is said to be *significantly improved* in comparison to $dist_b$.

In our proposal, two kinds of impact rules are treated differently to assess the rule derivability. To decide whether a rule, the antecedent of which is composed of only one condition, is derivative extended or not, we compare the target mean of this rule with the global target mean of the database. Only if the former is significantly improved compared with the latter, should the rule be accepted. As regards rules with more than one condition as antecedent, if the target mean of such a rule is significantly higher than the target means *all* of its direct parents', it is a significant rule.

Statistical Test

Since by performing the exploratory rule discovery, we are aiming at discovering rules that characterize the features of a population with reference to available sample data, hypothesis tests must be applied to assess whether a rule is derivative or not.

The chi-square (Bay and Pazzani; 2001; Liu et al.; 1999b) and the Fisher exact tests (Webb; 2005), which are both adopted to judge whether propositional rules are derivative or not, are not applicable with distributional-consequent rules. The standard z test was adopted by Aumman and Lindell for identifying quantitative association rule derivability. However, it is notoriously inappropriate for small samples Webb (2005). The t-test is a better statistical test for comparing means of independent samples of any size. As the *degree of freedom* for the t test increases with the size of the data sample, the t-test approaches the standard z test. In this way, better scalability is achieved.

The t-test is a well-known parametric test for detecting difference between sample means of two distributions. For parametric methods, we assume that the populations from which the samples are drawn must be at least approximately normally distributed or we rely on the central limit theorem to give us a normal approximation (Johnson; 1996).

The t-test is only applicable for comparing means of two *independent* samples, however, $coverset(A)$ is a subset of $coverset(A-x)$ in definition 3. Practically, the target means of $coverset(A)$ and $coverset(A - x) - coverset(A)$ are compared. In practical implementation, the definition of derivative extended rules is:

Definition 4 *DeriExtended*$(A \rightarrow target) =$

$$\begin{cases} tarmean(coverset(A)) \not\gg tarmean(coverset(\neg A)) & if\ |A| = 1; \\ \exists x \in A, tarmean(coverset(A)) \not\gg tarmean(coverset((A - x)\ \&\ \neg A)) & if\ |A| > 1. \end{cases}$$

Note that using statistical tests to automatically identify derivative extended rules is inherently statistically unsound (Webb; 2005). There are high risks of type-1 errors of accepting spurious or uninteresting rules, as well as type-2 errors of rejecting rules that are actually interesting. However, this is not an issue of concern in our thesis, because it can be solved by introducing the technique proposed by Webb (2005).

After applying the derivative extended filter using a t-test, only two impact rules remained as significant. The number of resulting rules goes through a decrease of nearly 90%. Here are the rules identified as significant after applying the derivative extended rule filter:

$$Cat1 = A\ \&\ Num <= 10 \rightarrow Target(coverage : 3, mean : 11.8, variance : 0.52,$$

$$min : 11, max : 12.4, sum : 35.4, impact : 27.96)$$

$$Num <= 10 \rightarrow Target(coverage : 9, mean : 4.66667, variance : 35.4425,$$

$$min : -3, max : 12.4, sum : 42, impact : 19.68)$$

4.1.2 Derivative Partial Rules

There exists, however, another type of derivative rules that are spurious or potentially uninteresting, which remain in the resulting set even after applying all the existing rule pruning techniques. For any rule A & $B \rightarrow C$ which is not derivative from another rule and for which there is a correlation between the antecedent and the consequent, both A and B are seemingly correlated with C solely due to correlation between A & B and C. In this case, $A \rightarrow C$ and $B \rightarrow C$ are both derivative rules that are potentially uninteresting.

The following example illustrates an occasion where such a potentially uninteresting rule may be generated.

Example 4 *Suppose a retailer is trying to identify the groups of customers who are likely to buy some new products. After applying the impact rule discovery with the derivative extended rule, the following rules are identified as solutions:*

$$Age > 50 \rightarrow profit(coverage = 200, mean = 100)$$

$$District = A \rightarrow profit(coverage = 200, mean = 100)$$

$$District = A \ \& \ age > 50 \rightarrow profit(coverage = 100, mean = 200)$$

Although these three rules all survived the derivative extended impact rule filter, the first two, which are ancestors of the third one are misleading. Actually, no profit is produced by customers who belong to district A and are older than 50 or those who are older than 50 but living outside district A! The first two rules happen to be "interesting" by virtue of the profits induced by the records covered by the third rule, which are only half of those covered by the first two. The retailer's attention should be more sensibly concentrated on the group of customers who are under age 50 in district A, instead of on all those in district A or those over 50 years of age. Keeping the first two rules in the resulting solutions may confuse the decision makers.

We refer to the first two rules in this example, which are potentially uninteresting as derivative partial rules.

In this section, we investigate the identification of *derivative partial rules*, in the context of impact rule discovery, which are different from the previously mentioned derivative extended rules. We define derivative partial impact rules as follows:

Definition 5 *A significant impact rule, $A \rightarrow target$ is a derivative partial rule, if and only if there exists a condition $x \notin A$, where the target mean for $coverset(A) - coverset(A\ \&\ x)$ is not higher than the target mean for $coverset(\neg A)$ at a user specified level of significance.*

$$DeriPartial(A \rightarrow target) = \exists x \in (\mathcal{C} - \{A\}),$$

$$TarMean(coverset(A\ \&\ \neg x)) \not\gg TarMean(coverset(\neg A))$$

As is argued in the previous section, existing techniques cannot successfully remove derivative partial rules. Even after both rules: $A \rightarrow target$ and $A\ \&B \rightarrow target$, have been identified as non-derivative extended rules, there is still a risk that either or both of them are potentially uninteresting. For example, if the target mean of $coverset(A\ \&\ \neg B)$ is not significantly higher than the target mean of $coverset(\neg A)$, it can be asserted that the notably high target mean of $coverset(A)$ is produced solely by virtue of that of $coverset(A\ \&\ B)$, which is only a subset of $coverset(A)$. *Derivative partial rules* are derivative from fundamental rules which are their children as opposed to derivative extended rules which can be derived from their fundamental ancestors.

After further rule pruning using the derivative partial rule filter using the t-test, only one significant rules that survived the derivative extended rule filter remains.

$$Cat1 = A\ \&\ Num <= 10 \rightarrow Target(coverage : 3, mean : 11.8, variance : 0.52,$$

$$min : 11, max : 12.4, sum : 35.4, impact : 27.96)$$

4.1.3 Relationship among Rules

As has been defined, rules that can somehow be derived from their parent or child rules are referred to as *derivative rules.* Contrarily, rules that are not derivative

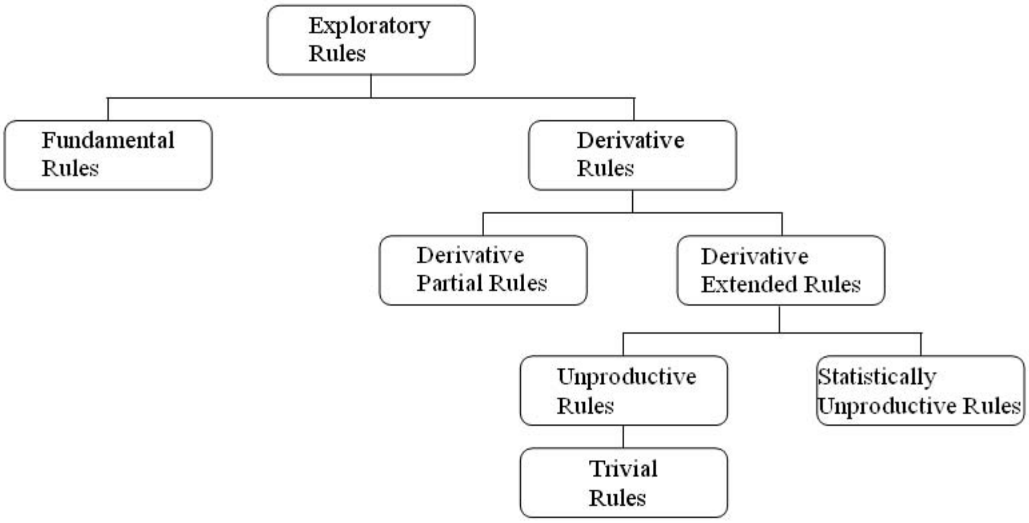


Figure 4.2: Relationship of different rules

with regard to either its generalizations or its specifications are all referred to as *fundamental rules*.

The relationships among different rules are explained in figure 4.2. In this figure, fundamental rules can also be referred to as *non-derivative rules*. Derivative extended rules are those defined as insignificant rules in previous research. Unproductive rules are those exhibit no improvement with respect to a specific measure of interestingness compared with their parent rules. Trivial rules are rules whose antecedents cover exactly the same set of records as that by one of their parent rules. As will be argued in the later part of this thesis, trivial rules are special derivative extended rules. Those that are productive, with respect to the sample, but fail the significance test are all classified as statistically unproductive.

4.1.4 Algorithm with Derivative Rule Pruning

Aumann and Lindell (1999) used the frequent itemset framework for their implementation of quantitative association rule discovery. However, in cases where there are numerous large itemsets, the overheads of itemset maintenance and manipulation can be unwieldy (Webb; 2000). Our k-optimal impact rule discovery which is constructed on the OPUS algorithm can successfully overcome this problem by performing efficient search space pruning. Moreover, Aumann and Lindell separated the two processes of rule discovery and rule filtering, which sacrifices some opportunities for using filtering for rule discovery process efficiency gains. We manage to improve rule pruning efficiency by combining these two processes and pruning the search regions that only contain spurious rules.

The algorithm for derivative impact rule filtering, OPUS_IR_Filter, is set up on the basis of OPUS. It systematically searches through the combinations of conditions that may appear on the antecedent of an impact rule and prune the search space according to the requirements of a particular search. Based on the OPUS structure, there is no need to allocate huge memory space to store all the frequent itemsets during the rule generation process. Hence, this is a better approach for discovering rules in very large, dense databases. However, it is not restricted to applications with large, dense databases only. Table 4.1 contains the pseudo code of OPUS_IR_Filter, which is our impact rule discovery algorithm with filters. The filtering of derivative extended impact rules is done at step 2.3.1 during impact rule discovery, by comparing the target mean of *current_rule* with all its direct parents'. Derivative partial rules are removed from the *rule_list* at step 2.3.5 according to definition 5.

In this algorithm, *rule_list* is an ordered list for the top k optimal rules that have been encountered, where k is specified by the users before searching. *Current*, *available* and $\mathcal{M}$ have the same meaning as in table 3.2. All the parent rules of *current_rule* are stored in *parent_rule_list* while checking whether *current_rule* is derivative extended or not. After *current_rule* is identified as significant, the

```
Algorithm:  OPUS_IR_Filter(Current, Available, parent_rule, $\mathcal{M}$)

  1 SoFar := $\emptyset$;

  2 FOR EACH P in Available

    2.1 New := Current $\cup$ P
    2.2 $current\_rule$ = $New \rightarrow target$
    2.3 IF New satisfies all the prunable constraints in $\mathcal{M}$ THEN
      2.3.1 FOR EACH direct subset $New'$ of $New$
        2.3.1.1 get statistics of $coverset(New')$
        2.3.1.2 IF $tarmean(New \rightarrow target) \not\gg tarmean(coverset(New') - coverset(New))$ THEN
                  go to step 2.3.7;
        2.3.1.3 add $New' \rightarrow target$ to $parent\_rule\_list$;
        2.3.1.4 END IF
      2.3.2 END FOR
      2.3.3 IF $New \rightarrow target$ satisfies all non-prunable constraints in $\mathcal{M}$
              record $New \rightarrow target$ to $rule\_list$
      2.3.4 END IF
      2.3.5 FOR the antecedent $New'$ of EACH $rule$ in $parent\_rule\_list$
        2.3.5.1 IF $tarmean(coverset(New') - coverset(New)) \not\gg tarmean(coverset(\neg New'))$ THEN
                  delete $New' \rightarrow target$ from $rule\_list$
        2.3.5.2 END IF
      2.3.6 END FOR
      2.3.7 OPUS_IR_Filter(New, SoFar, $mathcalM$)
      2.3.8 SoFar := SoFar $\cup$ P

    2.4 END IF;

  3 END FOR
```

Table 4.1: OPUS_IR_Filter with efficiency improvement

derivative partial rule filter is then applied to assess whether the parents are derivative partial with regard to *current_rule*. Derivative partial rules are deleted from the *rule_list*. Since all the parent rules of *current_rule* has already been explored before *current_rule* (please refer to the search space of OPUS_IR), every derivative rule is guaranteed to be removed following this procedure.

4.1.5 Efficiency Improving Techniques

Distributional-consequent rule discovery requires several passes through the database in order to collect necessary statistics for describing resulting rules. To implement the derivative filters also has a stringent demand for further computation and data accesses. This make exploratory rule discovery efficiency problems more severe. Little research have been done regarding how to improve the efficiency of distributional-consequent rule discovery. Herewith, we propose three techniques for improving impact rule pruning efficiency. The *triviality filter* is proposed as and alternative and complement for the derivative filters. Efficient search space pruning can be performed by imposing a special property of triviality. To reduce the redundancy of data accesses, we propose the difference set statistics derivation approach. The circular intersection approach is designed in order to get rid of redundancies in coverset generation.

Trivial Impact Rules

Although applying statistical tests during rule discovery enables successful removal of derivative extended impact rules, this approach requires additional passes through the database so as to obtain necessary statistics for performing the tests. We exploited possible improving schemes for more efficient search space pruning, and present the definition of trivial impact rules, which is a special case of a derivative extended rule. The property of triviality can quicken up the identification and removal of derivative extended rules.

Definition 6 *An impact rule $A \rightarrow target$ is* trivial *iff there is a rule $A' \rightarrow target$ where $A' \subset A$, and A' and A cover the identical set of records.*

$$trivial(A \rightarrow target) = \exists A' \subset A, coverage(A) = coverage(A')$$

We have talked about the anti-monotone constraints that can facilitate effective search space pruning and considerably improve the efficiency of rule discovery. We identify the anti-monotonicty of triviality related constraints and give the proof in the context of impact rule discovery.

Theorem 1 *"An impact rule is not trivial" is an anti-monotone constraint: if a rule A & $B \rightarrow target$ is trivial with regard to its parent rule: $A \rightarrow target$, then all the rules, whose antecedent is a superset of A & B, are also trivial*[1].

Proof 1 *According to definition 6,*

$$coverset(A) = coverset(A \ \& \ B). \tag{4.1}$$

For any record $r' \in D$, if

$$r' \notin coverset(A \ \& \ B\& \ C)$$

$$\Rightarrow r' \notin coverset(A \ \& \ B) \vee r' \notin coverset(C) \tag{4.2}$$

Consider equation 4.1

$$\Rightarrow r' \notin coverset(A) \vee r' \notin coversetC$$

$$\Rightarrow r' \notin coverset(A \ \& \ C)$$

So

$$\forall r \notin coverset(A \ \& \ B \ \& \ C) \rightarrow r \notin coverset(A \ \& \ C)$$

[1] *This proof is essentially equivalent to a proof of Webb and Zhang (2002)*

$$coverset(A \mathbin{\&} C) \subseteq coverset(A \mathbin{\&} B \mathbin{\&} C) \tag{4.3}$$

Since A & C is a subset of A & B & C,

$$coverset(A \mathbin{\&} B \mathbin{\&} C) \subseteq coverset(A \mathbin{\&} C) \tag{4.4}$$

It can be concluded from 4.3 and 4.4 that

$$coverset(A \mathbin{\&} B \mathbin{\&} C) = coverset(A \mathbin{\&} C)$$

Hence, the rule A & B & $C \rightarrow target$ is a trivial rule with regard to its parent A & $C \rightarrow target$. The theorem is proved.

At step 2.3.1 of the algorithm in table 4.1, we assess whether a rule is derivative or not by comparing the target mean of *current_rule* with all its direct parents'. To ease the implementation of the triviality filter, we assert that if a rule is not trivial with respect to any of its direct parents, it is not trivial either with any of its ancestors.

Lemma 1 *If $A \rightarrow target$ is a trivial rule, there must exist a direct parent of $A \rightarrow target$, which covers exactly the same set of records as $A \rightarrow target$.*

$$\exists x \in A, trivial(A \rightarrow target) \wedge coverage(A) = coverage(A - x)$$

Proof 2 *If there is no $x \in A$ which satisfies $coverage(A) = coverage(A - x)$ then*

$$\forall x \in A, coverset(A) \subset coverset(A - x)$$

$$\forall S \subset (A - x), coverset(S) \subseteq coverset(A - x)$$

If $S \subset (A - x)$ then $S \subset A$, so it is obvious that

$$\forall S \subset A, coverset(S) \subset coverset(A)$$

which contradicts our assumption. The theorem is proved.

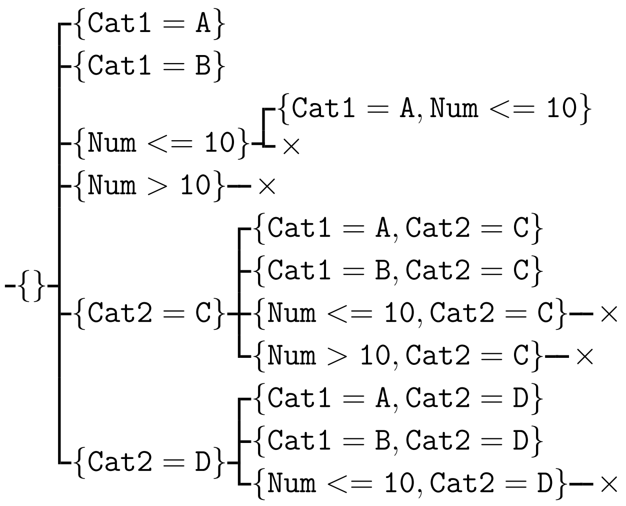


Figure 4.3: Pruned search space at step 2.2.1

According to lemma 1, if rule $A \rightarrow target$ is trivial, there must be a condition $x \in A$ where $coverset(A) = coverset(A - x)$. It follows that the target distribution of $A \rightarrow target$ and its direct parent $A - x \rightarrow target$ are the same. Considering also the definition of *derivative extended* impact rules, we can safely conclude that a trivial rule is a derivative extended rule. By applying a statistical test to compare the target mean of *current_rule* with all its direct parents only, we are able to identify all trivial impact rules. Nonetheless, the triviality filter is more powerful in its effect, for its anti-monotonicity accelerates the search space pruning process and no computational expenses are required for doing statistically tests.

Lemma 1 ensures that we only have to check all direct subset of the rule antecedent to assess the triviality of that rule. Theorem 1 justifies our pruning at step 2.3 in table 4.3, which dramatically improves the efficiency of search.

Figure 4.3 shows the effect of pruning according to triviality in OPUS_IR_Filter search space for the fictitious database. As an example, node {`Num>10, Cat2=D`} is trivial, so any superset of {`Num>10, Cat2=D`} should be pruned without accessing the records, according to theorem 1. After applying the triviality filter of impact

Trivial rules	Nontrivial counterparts
Cat1=B & Num<=10 & Cat2=D → Target (coverage:3 mean:2.26667 variance:15.4133 min:0 max:6.8 sum:6.8 impact:-0.64)	Cat1=B & Cat2=D → Target (coverage:3 mean:2.26667 variance:15.4133 min:0 max:6.8 sum:6.8 impact:-0.64)
Cat1=B & Num<=10 & Cat2=C → Target (coverage:3 mean:-0.0666667 variance:7.26333 min:-3 max:2.3 sum:-0.2 impact:-7.64)	Cat1=B & Cat2=C → Target (coverage:3 mean:-0.0666667 variance:7.26333 min:-3 max:2.3 sum:-0.2 impact:-7.64)
Cat1=B & Num<=10 → Target (coverage:6 mean:1.1 variance:10.704 min:-3 max:6.8 sum:6.6 impact:-8.28)	Cat1=B → Target (coverage:6 mean:1.1 variance:10.704 min:-3 max:6.8 sum:6.6 impact:-8.28)
Cat1=A & 10<Num & Cat2=C → Target (coverage:4 mean:-0.9 variance:3.28667 min:-2.7 max:1.6 sum:-3.6 impact:-13.52)	10<Num & Cat2=C → Target (coverage:4 mean:-0.9 variance:3.28667 min:-2.7 max:1.6 sum:-3.6 impact:-13.52)
Cat1=A & 10<Num → Target (coverage:6 mean:-0.8 variance:2.14 min:-2.7 max:1.6 sum:-4.8 impact:-19.68)	10<Num → Target (coverage:6 mean:-0.8 variance:2.14 min:-2.7 max:1.6 sum:-4.8 impact:-19.68)

Table 4.2: Trivial rules found in the fictitious database

rules, 5 out of the 19 rules found without using any filter are removed. The trivial rules and their corresponding nontrivial counterparts are listed in table 4.2.

Difference Set Statistics Derivative Approach without Data Access

According to the algorithm in table 4.1 and definition of derivative extended rules, we have to compare the target mean of current rule with those of all its direct parents' in order to assess whether a rule is *derivative extended* or not. The set difference operations necessary for performing the statistical tests require excessive data accesses and computation. However, by examining the implementation of the derivative extended rule filter, with the status of current rule and any of its parent rule known, we are able to derive the statistics of the difference sets for performing the statistical tests without additional accesses to the database. For example, since the OPUS_IR_Filter function calls itself recursively, when we are trying to identify the derivability of node {`Cat1=A, Num<=10`}, we can have the status of rule $Num <= 10 \rightarrow target$, which is a parent rule of *current_rule*: $Cat1 = A \,\&\, Num <= 10 \rightarrow target$, as the function input. The comparison between $coverset(Cat1 = A, Num <= 10)$ and $coverset(Num <= 10)$ can be done with no additional data access. The following lemma validates the above statement.

Lemma 2 *Suppose we are searching for impact rules from a database $\mathcal{D}$. If $A \subset B$, and $coverset(A) - coverset(B) = \mathcal{R}$, where A and B are both conjunctions of conditions, and $\mathcal{R}$ is a set of records from $\mathcal{D}$. If the means and variances of the target attribute over $coverset(A)$ and $coverset(B)$ are known, as well as the coverages of both record sets, the mean and variance of the target attribute over set $\mathcal{R}$ can be derived without additional data access.*

Proof 3 *Since $coverset(A) - coverset(B) = \mathcal{R}$, it is obvious that*[2]

$$|\mathcal{R}| = coverage(A) - coverage(B) \tag{4.5}$$

$$mean(\mathcal{R}) = \frac{coverage(A) \times mean(A \rightarrow target) - coverage(B) \times mean(B \rightarrow target)}{|\mathcal{R}|} \tag{4.6}$$

$$variance(A \rightarrow target) = \frac{\sum_{x \in coverset(A)} (target(x) - mean(A \rightarrow target))^2}{coverage(A) - 1} \tag{4.7}$$

$$variance(B \rightarrow target) = \frac{\sum_{x \in coverset(B)} (target(x) - mean(B \rightarrow target))^2}{coverage(B) - 1} \tag{4.8}$$

$$\sum_{x \in coverset(A)} target(x) = mean(A \rightarrow target) \times coverage(A) \tag{4.9}$$

$$\sum_{x \in coverset(B)} target(x) = mean(B \rightarrow target) \times coverage(B) \tag{4.10}$$

[2] *Coverset(A) represents the set of records that satisfy all the conditions in A.*

From 4.7, 4.8, 4.9 and 4.10 it is feasible to derive the following equation:

$$\begin{aligned}\sum_{x\in\mathcal{R}} target(x)^2 &= \sum_{x\in coverset(A)} target(x)^2 - \sum_{x\in coverset(B)} target(x)^2\\ &= variance(A \rightarrow target) \times (coverage(A) - 1)\\ &\quad +mean(A \rightarrow target)^2 \times coverage(A)\\ &\quad -variance(B \rightarrow target) \times (coverage(B) - 1)\\ &\quad -mean(B \rightarrow target)^2 \times coverage(B)\end{aligned} \tag{4.11}$$

$$\sum_{x\in\mathcal{R}} target(x) = \sum_{x\in coverset(A)} target(x) - \sum_{x\in coverset(B)} target(x) \tag{4.12}$$

Thus,

$$\begin{aligned}variance(\mathcal{R}) &= \frac{\sum_{x\in\mathcal{R}} (target(x) - mean(\mathcal{R}))^2}{|\mathcal{R}| - 1}\\ &= \frac{\sum_{x\in\mathcal{R}} target(x)^2}{|\mathcal{R}| - 1} - \frac{2mean(\mathcal{R}) \sum_{x\in\mathcal{R}} target(x)}{|\mathcal{R}| - 1} + \frac{|\mathcal{R}|mean(\mathcal{R})^2}{|\mathcal{R}| - 1}\end{aligned}$$

Since all the parameters in the right hand side of the equation are known, we are able to derive all the necessary statistics for performing statistical tests without generating the difference set from the database, or accessing the records in $\mathcal{R}$. The lemma is proved.

Note: in this proof, $mean(A \rightarrow target)$ denotes the target mean of the records covered by rule $A \rightarrow target$, $variance(A \rightarrow target)$ denotes the target variance of the records covered by rule $A \rightarrow target$; while $mean(\mathcal{R})$ denotes the target mean of the records in record set $\mathcal{R}$, and $variance(\mathcal{R})$ represents the target variance of the records in $\mathcal{R}$.

According to definition 4, $Coverset(\neg A)$ or $coverset((A - x)\ \&\ \neg A)$ has to be generated before another necessary pass through the database for collecting necessary statistics for statistical test to determine whether *current_rule* is derivative or not. Considering the above lemma, we are able to save a great deal of data accesses and computation for collecting necessary statistics if we have already known the

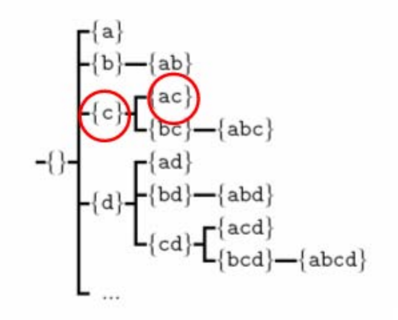


Figure 4.4: Difference set derivation approach

status of *parent_rule*. For example, in figure 4.4, when we are trying to identify whether rule $ac \rightarrow target$, which is connected with node `ac` is derivative or not, we can derive necessary statistics of $coverset(a\neg c)$ for testing rule derivability without any further data accesses.

Moreover, if *current_rule* is derivative from the input *parent_rule*, right after which the current rule is explored, following computation for checking the derivability of current rule with its other parents are not necessary. The efficiency of the search algorithm can thus be considerably improved.

Moreover, after *current_rule* is identified as significant, everyone of its parent rules stored in the *parent_rule_list* is compared with current rule to assess whether it is derivative partial or not. By applying the difference set statistics derivation approach, the comparisons can be done with no additional data accesses either.

Circular Intersection Approach

According to the definition of derivative extended impact rules, we compare *current_rule* with all its *direct parents* to identify whether it is derivative extended or not. In the original OPUS_IR_Filter algorithm in table 4.1, the procedure described in figure 4.5 is employed to generate the *coverset*s of every direct parent of the current rule. Each arrow in figure 4.5 represents an intersection operation. When deciding whether a rule with 5 conditions, namely A, B, C, D and E, on the antecedent is derivative extended or not, the algorithm requires 16 intersection operations! We call this approach the *parallel intersection approach*.

By examining figure 4.5, we notice that there are considerable overlaps in the *parallel intersection approach*. For example, by using the parallel intersection approach, we have to do the same intersection of $coverset(A)$ and

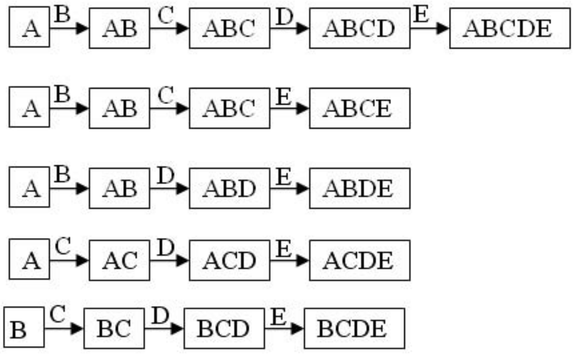


Figure 4.5: The parallel intersection Approach for $ABCDE$

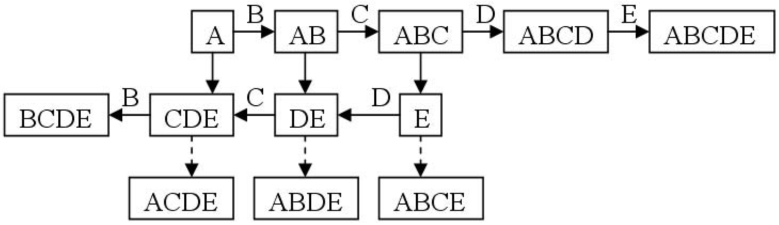


Figure 4.6: The circular intersection approach flow for $ABCDE$

$coverset(B)$ three times, when searching for $coverset(ABCD)$, $coverset(ABCE)$ and $coverset(ABDE)$.

We propose a more efficient approach for the same coverset generation task, named the *circular intersection approach*, which is shown in figure 4.6. Each dashed arrow in this figure points to the outcome of that specific intersection operation and does not represent an actual intersection operation. In this approach, intersections are done in two stages. Firstly, in the *forward stage*, intersections are done from condition A to condition E one at a time, and the meta-resulting coversets are kept in memory. Then, we do intersections from the last condition E back to the second one B, one by one, which is referred to as the *backward stage*. In the backward stage, the $coverset$ of each direct parent of $current\_rule$ is found by intersecting the meta-resulting coversets produced in the forward stage with those in the backward stage. The memory storage required for storing the outputs of the forward stage is freed during the backward stage. By introducing the circular intersection approach, the number of intersection operations required for identifying the derivability of the current rule is reduced to only 10.

Complexity

Using the parallel intersection approach, the maximum number of intersection operations for iterating through all the subsets is:

$$(n-2) \times n + 1,$$

where n is the maximum number of conditions on the rule antecedent. The complexity is $O(n^2)$.

After introducing the circular intersection approach, the maximum intersection operations for iterating through all the subsets are:

$$3n - 5.$$

The complexity is $O(n)$. However, practically, the difference in running time is so dramatic, since we have introduced several techniques for pruning the search space. Both the parallel intersection procedure and the circular intersection procedure are apt to stop at any point when it is identified that *current_rule* is potentially uninteresting.

The two approaches (the difference set statistics derivation approach and the circular intersection approach) mentioned above can be combined with each other so as to achieve more desirable efficiency. We can delete one more intersection operation by introducing the difference set statistics derivation technique in section 4.1.5. Suppose that we are deciding whether the rule $A \;\&\; B \;\&\; C \;\&\; D \;\&\; E \rightarrow target$ is significant or not. Now that the statistics of one of its parent $A \;\&\; B \;\&\; C \;\&\; D \rightarrow target$ is known, we don't have to derive necessary statistics for $coverset(ABCD)$. Hereby, one intersection operation can also be removed by following the procedure shown in figure 4.7 according to lemma 4.1.5. The maximum number of required intersection operations is reduced to

$$3n - 6.$$

The new OPUS_IR_Filter algorithm with impact rule discovery efficiency improving techniques that have been introduced is shown in table 4.3. In this table, *parent_rule* is the corresponding rule for the node whose children that are currently being explored. The antecedent of *parent_rule* is *current*.

```
Algorithm:  OPUS_IR_Filter(Current, Available, parent_rule, $\mathcal{M}$)

  1 SoFar := $\emptyset$;

  2 FOR EACH P in Available

    2.1 New := Current $\cup$ P

    2.2 $current\_rule$ = $New \rightarrow target$

    2.3 IF New satisfies all the prunable constraints in $\mathcal{M}$ except
        the nontrivial constraint THEN

      2.3.1 Derive the statistics of $coverset(Current) - coverset(New)$
            using the difference set statistics derivation
            approach.
      2.3.2 IF the $tarmean(New \rightarrow target) \not\gg$
            $tarmean(coverset(Current) - coverset(New))$ THEN
                go to step 2.3.4;
      2.3.3 ELSE use the circular intersection to compare
            $tarmean(New \rightarrow target)$ with the mean of its direct
            parents other than $parent\_rule$
        2.3.3.1 IF $tarmeanNew \rightarrow target$ is significantly improved
                comparing to all its direct parents' THEN
                    IF $New \rightarrow target$ satisfy all non-prunable
                constraints in $\mathcal{M}$ THEN
                        record $New \rightarrow target$ to $rule\_list$;
                    END IF
        2.2.3.2 END IF;
        2.3.3.3 OPUS_IR_Filter(New, SoFar, $New \rightarrow target$, $\mathcal{M}$);
        2.3.3.4 SoFar := SoFar $\cup$ P ;
      2.3.4 END IF;

    2.4 END IF;

  3 END FOR
```

Table 4.3: OPUS_IR_Filter with efficiency improvement

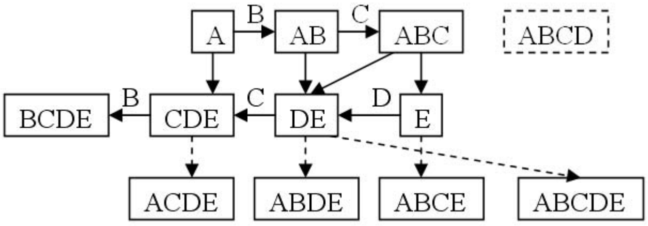


Figure 4.7: The circular intersection approach for $ABCDE$ when *current* is $ABCD$

4.2 Summary

Since we have observed the necessities for developing efficient distributional-consequent rule pruning techniques, we designed algorithms for automatically discarding spurious rules which are derivative with regard to other existing solutions in the context of impact rule discovery. We first proposed a derivative extended impact rule filter as an efficient variant of the insignificant quantitative association rule pruning proposed by Aumann and Lindell (1999). We also argued that there is another type of derivative rules, which is potentially uninteresting and has not been identified in previous research. We proposed a new filter for pruning such derivative partial impact rules. In this way, a more compact set of resulting impact rules is discovered.

Next, we presented three techniques for improving the efficiency of impact rule pruning. The first technique utilizes the anti-monotonicity of the non-triviality constraint and enables more powerful search space pruning during the course of rule discovery, which can theoretically improve the efficiency of our derivative extended rule filter.

We also identified substantial redundancies of data accesses and intersection operations in a straightforward implementation of impact rule discovery with filters. We proposed the *difference set statistic derivation approach* and the *circular intersection approach* to tackle these problems.

In the next chapter, we apply the resulting algorithms to several real world and synthetic databases. Experimental results are analyzed in detail and comparisons are done with previous approaches to provide experimental corroboration of the theoretical analyses.

Chapter 5

Experimental Evaluations

In chapter 4, we have proposed algorithms for automatically pruning derivative impact rules, together with three techniques which can theoretically improve the efficiency of derivative impact rule pruning.

In this chapter, we do our experiments by applying the techniques proposed in the last chapter to several large, dense databases selected from the UCI Machine Learning repository (Blake and Merz; 1998) and the UCI KDD Archives (Bay; 1999) to justify that a great amount of derivative extended and partial rules exist in the resulting set of rules. Our proposed algorithm with derivative rule filters can successfully remove such derivative rules during rule discovery and effectively reduce the number of resulting rules. Result analyses attest that the overall interestingness of the top k optimal resulting rules can be improved with un-useful rules removed and more rules that convey interesting information discovered.

We also evaluate the efficiency of OPUS based derivative extended rule filter against the significant *Quantitative Association Rule Discovery* proposed by Aumann and Lindell (1999) using the *Apriori* implementation provided by Borgelt and Kruse (2002). We draw the conclusion that *the trivial filter, the difference set statistics derivation approach* and *the circular intersection approach* can empirically improve the rule discovery efficiency to a great extent, especially for relatively denser databases.

In the beginning of this chapter, we introduce the databases that are selected for the experiments and predictions are made on possible effects of our algorithm. Next, the experiment design is explained, which is followed by the experimental result analysis and comparisons.

5.1 Experimental Data

Bayardo, Jr. et al. (1999) argued that a database is dense if it confirms with any of the following criteria:

1. Strong correlations between several items.
2. Many items in each record.

According to these criteria, we select ten databases from UCI Machine Learning (Blake and Merz; 1998) and UCI KDD Archives (Bay; 1999) with different density for our experiments. None of these databases has missing values. The numbers of records in these database vary from less than 300 to half a million. The numbers of attributes also vary from under 10 to over 80. With such great differences among these databases, we will be able to analyze the effectiveness of our techniques more thoroughly. The databases are described as below.

1. **Abalone:** This database is originally developed for predicting the age of abalone from physical measurements (Nash, Sellers, Talbot, Cawthorn and Ford; 1994). In our experiments, we choose to use this database to discover connections between the other attributes and *shucked weight*, which is *continuous* and is measured by grams. The data set samples are highly overlapped. The interactions implicit in this database are relatively sparse considering its size, with only 1 qualitative attribute, 8 quantitative attributes (including 1 discrete attribute and 7 continuous attributes) in this database.

2. **Heart:** This database was originally designed for classification tasks in which a diagnose is made about whether a patient has heart disease or not. In our experiments, we try to discover relationship between other attributes and the *maximum heart rate* a patient achieves. Totally 13 attributes are contained in this database, with 6 being qualitative, 1 being discrete quantitative and 6 being continuous quantitative. This is the smallest database both in terms of size and density.

3. **Housing:** Taken from the *StatLib* library which is maintained at Carnegie Mellon University, this database concerns about housing values in suburbs of Boston (Harrison and Rubinfeld; 1978). There are 13 continuous quantitative

attributes, except one which is nominal qualitative, in this database. In our experiments, we are interested in the values for the attribute *MEDV*. This database is relatively smaller and sparser compared with the others.

4. **German-credit:** This database was initially designed by Professor Dr. Hans Hofmann for classifying people described by a set of attributes as good or bad credit risks. In our impact rule discovery task, we try to discovery influence of other attributes on *Credit amount*. 7 quantitative attributes and 13 qualitative attributes are in this database.

5. **IPUMS Series:** The *Integrated Public Use Microdata Series* project (Ruggles and Sobek; 1997) standardize federal census data to allow researcher to compare demographic groups over different time periods. The databases we choose are the "large" versions, which contain unweighed, 1 in 100 samples of the Los Angeles and Long Beach area respectively for the years 1970 (for *Ipums.la.97*), 1980 (for *Ipums.la.98*), and 1990 (for *Ipums.la.99*). All these three databases are composed of 19 quantitative attributes among which we select the *total income* as the target variable, and 42 qualitative attributes. Several qualitative attributes have numerous values. The *occupation* attribute, as an example, may take over 160 different values.

6. **Ticdata2000:** This database is also known as the *COIL2000* (The Insurance Company Benchmark) database (van der Putten and van Someren; 2000). It contains great ranges of information about customers. The data was collected originally to identify customers who would be interested in buying a caravan insurance policy. Since the *income* attribute in this database has already been discretized for classification and regression, we choose the attribute *average income* as the target variable of concern. Although this database has less than 6000 records, the number of attributes turn out to be the greatest of all, which is 86. There are 60 qualitative attributes, and 26 quantitative attributes in the *Ticdata2000* database.

7. **Census income:** This dataset contains weighted census data extracted from the 1994 and 1995 *Current Population Surveys* conducted by the U.S. Census Bureau. The data contains 41 demographic and employment related

attributes. In this experiment *wage per hour* is chosen to be the *target variable*. Except the *target variable*, the database include 33 qualitative and 7 continuous quantitative attributes. Duplicate or conflicting instances find their places in this database. We predict that the number of derivative impact rules discovered from this database would be large.

8. **Covtype:** This database is composed of the forest cover types for 30 × 30 meter cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data. The *forest cover type* is the initial problem of classification. However, we choose to use the attribute ”elevation” which is measured in meters, as the target variable. Having more than 500,000 records, this is the largest database in size, among those we select. The computational and data access expenses are expected to be huge, especially when the derivative filters are applied. However, it does not have a huge number of attributes considering its size, with 10 quantitative attributes, 44 qualitative attributes.

To run the program on these databases, we applied 3-bin equal-frequency discretization to map of all continuous quantitative attributes in the above databases into ordinal qualitative ones, except the specified *target variables*. After the discretization, the *ipums* series, the *ticdata2000* and the *census income* databases turn out to have more than 500 conditions. We predict that our algorithms can perform much better on them than the frequent itemset based algorithm. The largest database of all, *covtype*, has only 131 conditions. It is not as dense as the previously mentioned five, but the time for discovering impact rules can be very long, considering its size. We predict that computation for collecting necessary statistics for rule description for this database is the most expensive. To produce a more clear idea of the databases, the basic information is also presented in table 5.1.

5.2 Experimental Design

Since the effectiveness and efficiency of our proposed techniques have been positively stated, we design our experiments in a protocol that these two improvements can be exhibited. We devote ourselves to analysis of reductions in the number of

database	**records**	**attributes**	**conditions**	**Target**
Abalone	4117	9	24	Shucked weight
Heart	270	13	40	Max heart rate
Housing	506	14	49	MEDV
German credit	1000	20	77	Credit amount
Ipums.la.97	70187	61	1693	Total income
Ipums.la.98	74954	61	1610	Total income
Ipums.la.99	88443	61	1889	Total income
Ticdata2000	5822	86	771	Ave. income
Census income	199523	42	522	Wage per hour
Covtype	581012	55	131	Elevation

Table 5.1: Basic information of the databases

resulting rules and the CPU time spent for loading data, discovering rules and outputing rules all together. For all the following experiments, we run our algorithms with the following parameter settings:

1. Minimum coverage: 0.01.

2. Significance level for the derivative filters: 0.05.

3. Maximum number of resulting rules: 1000.

4. Interestingness measure for ranking the resulting rules: imapct[1].

The computer on which the algorithms are run has two 933MHz processors (actually, only one is used for our algorithms), 1.5 G of actual memory and 4 G of virtual memory.

In the first suite of experiments for evaluating the effectiveness of the derivative filters, we run the impact rule discovery initially with no filters, after which with the derivative extended rule filter only. Both filters are applied in the last step. We run the program with the maximum number of conditions allowed on rule antecedent set to 3, 4 and 5. The changes in numbers of rules and percentages of decrease in resulting rules are shown in table 5.2.

To compare the efficiency of the derivative extended filter with that of the previous technique proposed by Aumann and Lindell (1999), we run the Apriori implementation using the same databases with target attributes removed. This is thus designed for the purpose of simulating the first step of frequent itemset

[1] Please refer to section 2.2.6 for definition of impact.

generation for Aumann and Lindell (1999)'s insignificant quantitative association rule pruning. The discovered frequent itemsets compose of the antecedents for the resulting quantitative association rules. We compile Borgelt and Kruse (2002)'s Apriori implementation, which, to our knowledge, is one of the most efficient implementations of Apriori, on the same computer with the same compiler settings as those for the OPUS_IR_Filter algorithm. The CPU time spent for their Apriori implementation to discover all the frequent itemsets whose sizes are under 5 is recorded, as well as the number of frequent itemsets generated by each database. Then we compare the efficiency of these two algorithms. It should be noted that, with the Apriori implementation, rule antecedents are discovered, without generating rules with distributional statistics. However, deriving statistics for rule description is prohibitively expensive.

In the last set of experiments, we introduce the three efficiency improving techniques proposed in the last chapter into the OPUS_IR_Filter implementation one by one. Since we have argued that the triviality filter can function as an alternative to as well as a complement for the derivative extended rule filter, we compare resulting rule sets as well as the running time for discovering rules with triviality filter with those for rule discovery with the derivative extended rule filter. We also combine both filters to show the efficiency improvement brought by introducing triviality filter.

After this, the difference set statistics derivation and the circular intersection approaches are introduced respectively into the algorithm before combining with each other. The algorithms are run with different maximum numbers of conditions allowed on rule antecedents. Comparisons are done with the OPUS_IR_Filter algorithm in table 4.1 with the triviality filter. Trends of changing in CPU time with the allowed maximum number of conditions are shown.

5.3 Results and Analyses

In this section, we present analyses of experimental results for all the proposed techniques in detail, according to the three set of experiments described in the section of experimental design.

5.3.1 Effectiveness of Derivative Filters

Table 5.2 systematically present the changes in resulting rules caused by applying the filters. The sub-columns titled “Rules” under the “extended” column contain the numbers of rules (before the slashes, if there are) that are found *significant* in the top 1000 impact rules after the extended rule filter is applied. The numbers in these sub-columns after the slashes are the numbers of resulting significant rules actually found after applying the derivative rule filter, if the number is under 1000. The sub-columns titled “percentage” contain the percentages of decrease in resulting rules after applying the derivative extended rule filter. Similarly, the sub-columns under “partial” show the numbers of rules remain as fundamental in the discovered top 1000 significant impact rules and the percentages of decrease in numbers of rules after introducing the derivative partial rule filter.

Here is an example of derivative extended rules in the *abalone* database:

$$Sex = M \ \& \ 1.0295 <= Whole\_weight \ \& \ 0.294 <= Shell\_weight \rightarrow Shucked\_weight$$

$$(coverage : 595, mean : 0.631413, variance : 0.0309362, min : 0.315,$$

$$max : 1.351, sum : 375.69, impact : 161.867)$$

With respect to its parent rule:

$$Sex = M \ \& \ 0.294 <= Shell\_weight- > Shucked\_weight$$

$$(coverage : 676, mean : 0.599229, variance : 0.0353118, min : 0.189,$$

$$max : 1.351, sum : 405.079, impact : 162.147)$$

The following impact rule is discarded as derivative partial for the *abalone* database:

$$Sex = M \rightarrow Shucked\_weight \ (coverage : 1528, mean : 0.432946,$$

$$variance : 0.049729, min : 0.0065, max : 1.351, sum : 661.542, impact : 112.428)$$

Database	MNC=3				MNC=4				MNC=5			
	extended		partial		extended		partial		extended		partial	
	Rules	pecentage	Rules	pecentage	Rules	pecentage	Rules	pecentage	Rules	pecentage	Rules	pecentage
Abalone	86/86	91.4%	82	4.65%	138/138	86.2%	127	7.97%	173/173	82.7%	149	13.87%
Heart	54/57	94.6%	43/57	24.56%	57/80	94.3%	63/80	21.25%	52/100	94.8%	81/1000	19.0%
Housing	135/171	86.5%	131	23.39%	102/255	89.8%	168	34.12%	/288	%	192	33.33%
German credit	74/197	92.6%	152	22.84%	46/273	95.4%	213	21.98%	/295	%	222	24.75%
Ipums.la.97	87	91.3%	949	5.1%	44	95.6%	867	13.3%	31	96.9%	809	19.1%
ipums.la.98	687	31.3%	944	5.6%	468	53.2%	890	11.0%	133	86.7%	761	23.9%
Ipums.la.99	572	42.8%	959	4.1%	471	52.9%	930	7.0%	297	70.3%	896	10.4%
Ticdata2000	6	99.4%	803	19.7%	1	99.9%	739	26.1%	1	99.9%	674	32.6%
Census income	337	66.3%	894	10.6%	62	93.8%	776	22.4%	30	97%	744	25.6%
Covtype	280	72%	918	8.2%	316	68.4%	829/1000	17.1%	255	74.5%	733	26.7%

Table 5.2: Changes in resulting set with derivative filters

It is derivative regarding its child rule:

$$Sex = M \ \& \ 1.0295 <= Whole\_weight \rightarrow Shucked\_weight \ (coverage : 687,$$

$$mean : 0.619229, variance : 0.0284951, min : 0.315, max : 1.351,$$

$$sum : 425.411, impact : 178.525)$$

In this example, if an abalone is male but have a whole weight less than 1.0295 can not have a very high shucked weight.

From the experimental results shown in table 5.2, we make the following observations:

1. As the number of maximum conditions allowed on rule antecedents increases, generally, more derivative rules are produced.

2. The derivative extended rule filter can successfully remove a great amount of uninteresting rules from resulting set. As we have predicted, the decreases of rules for the denser databases are dramatic. It is also notable that after the derivative extended rule filter is introduced, our k-optimal impact rule discovery algorithm is able to discover all the significant rules with *abalone, housing, heart* and *German credit*, even when the maximum number of conditions is set to 5 and k is set to 1000.

3. With the derivative partial rule filter applied, a great portion of the resulting significant rules are identified *derivative partial*. The greatest change encountered in the number of resulting rules after introducing the derivative partial rule filter is as much as 34% for the *housing* database with the maximum number of conditions allowed on antecedent set to 4. Even the database with the slightest change saw a decrease of over 4%. This justify our argument that there are considerable amount of derivative partial rules in the resulting rules even after the derivative extended rule filter is applied. Derivative partial rules exist profoundly in the discovered rules, and cannot be removed using previous rule pruning techniques. By applying the derivative partial rule filter, we make another step towards reducing resulting rule in the context of impact rule discovery.

5.3.2 Comparisons with Quantitative Association Rule Discovery

As is mentioned before, Aumann and Lindell's algorithm for removing insignificant quantitative association rules used the frequent itemset framework, which is limited in its capacity to analyze dense data by the requirement of vast amount of memory to store all the frequent itemsets and excessive computation for manipulating these frequent itemsets during the generation procedure. It is after this stage that statistical tests are performed over the set of resulting rules. This frequent itemset generation task is not optimal in term of efficiency when applied to large, dense databases.

The running time and the numbers of frequent itemsets discovered in each of the 10 selected databases are listed in table 5.3. The column titled "frequent itemsets" contains the numbers of generated frequent itemsets for each database, which equal to the numbers of resulting quantitative association rules. The columns titled "CPU time" and "CPU time for OPUS_IR" show the running time for Borgelt and Kruse (2002)'s Apriori to generate frequent itemsets and OPUS_IR_Filter (with no filters) to discovery the top 1000 impact rules respectively. The maximum size of frequent itemsets is set to 5, which is the same as the maximum number of conditions allowed on resulting impact rule antecedents for our OPUS_IR_Filter algorithm.

By comparing the experimental results, we discover that is Apriori cannot successfully discover rules in databases with a huge number of conditions. For *ipums.la.97, ipums.la.98, ipums.la.99* and *ticdata1000*, whose number of conditions exceed 700, the program stops because of insufficient memory before generating all the frequent itemsets can be generated. However, OPUS_IR Filter can be applied to the above databases successfully and efficiently. The time spent on looking for all the frequent itemsets in *german credit*, *census income* and *covtype* are much longer than that required for OPUS_IR_Filter. Although for *abalone* the running time appears more desirable than our approach, it should be noted that the recorded time is only for generating frequent itemsets, time spent for statistics computing and data accesses associated with the statistics calculation for the target attribute for each itemset are not taken into account. However, it is known to all that going

Database	Frequent Itemsets	CPU time(sec)	CPU time for OPUS_IR
Abalone	11131	0.07	0.29
Heart	91213	0.11	0.05
Housing	129843	0.20	0.06
German credit	2721279	4.16	0.47
Ipums.la.97	-	stop after 18462.20	7.25
Ipums.la.98	-	stop after 17668.01	1382.66
Ipums.la.99	-	stop after 10542.40	874.20
Ticdata2000	-	stop after 103.17	1996.57
Census income	314908607	7448.52	873.74
Covtype	58920053	17488.26	16971.99

Table 5.3: Results for Apriori

through the data is one of the disasters for efficiency. What is more, the computational and data access expenses are unimaginably cumbersome for large databases like *covtype* which has nearly 6 hundred thousand records.

Even if we do not take the time spent on itemset discovery into account, to apply statistical tests over all the resulting frequent itemset is time-consuming (note that the number of itemsets found in some of the databases exceeds 10^6).

5.3.3 Triviality Filter

The triviality filter was proposed as a complement for the derivative extended filter, and can successfully remove a subset of the derivative extended rules. To evaluate the effectiveness of the triviality filter, we compare both necessary running time and changes in resulting rules after introducing the triviality filter alone with those for the algorithm without filters and those for the algorithm with the derivative extended rule filter only. Changes in the resulting numbers of rules induced by the triviality filter is presented in table 5.4[2]. The second column in table 5.4 contains the number of rules accepted as significant (using the derivative extended rule filter) in the top 1000 impact rules, while the third column includes the numbers of non-trivial rules in the top 1000 impact rules. The last column shows the number of non-trivial rules in the top 1000 that survive the derivative extended rule filter.

CPU time comparisons are shown in table 5.5. The first two columns in this table present the CPU time for discovering the top 1000 impact rules with no filters

[2] The maximum number of conditions allowed on resulting impact rule antecedents is set to 5 in this experiment.

Database	Significant rules in top 1000	Nontrivial rules in top 1000	Significant rules in top 1000 nontrivial rules
Abalone	173(173)	998	173
Heart	52(100)	923	54
Housing	83(288)	935	84
German credit	31(295)	738	43
Ipums.la.97	31(1000)	31	1000
Ipums.la.98	133(1000)	138	803
Ipums.la.99	297(1000)	578	507
Ticdata2000	1(1000)	564	1
Census income	30(1000)	466	42
Covtype	255(1000)	410	533

Table 5.4: Comparison in number of rules

Database	top 1000	triviality	derivative extended rule filter	
	impact rules	Filter	Derivative extended only	Both
abalone	0.29	0.57	0.75	0.74
heart	0.05	0.08	1.16	1.2
housing	0.06	0.16	1.62	1.47
german-credit	0.47	0.85	30.35	29.14
ipums.la.97	7.25	471.56	7365.23	623.52
ipums.la.98	1382.66	1551.8	1871.35	1860.31
ipums.la.99	874.2	1006.9	1886.07	1414.88
ticdata2000	1996.57	2082.1	10933.98	10808.03
census-income	873.74	1396.2	3960.84	3781.6
Covtype	16971.99	18682.52	20686.95	19496.71

Table 5.5: Running time for discovering rules (in seconds)

and for discovering top 1000 non-trivial rules using the triviality filter. Then the time spent for discovering significant rules with and without the introduction of triviality filter is listed in the last two columns.

We can see from column 2 and column 3 of table 5.4 that, although the triviality filter can not automatically discard as many spurious impact rules as those by the derivative extended rule filter, the decrease in number is also considerable. For i*pums.la.97* only 31 rules among the top 1000 impact rules found without using any filter is nontrivial, while all the nontrivial impact rules are accepted as significant! Moreover, for databases *ipums.la.98, ipums.la.99, covtype, ticdata2000* and *census income*, more than 40% of the resulting impact rules are discarded as trivial.

By examining the data in table 5.5, we conclude that applying only the triviality filter requires less CPU time, and the efficiency of for discovering significant impact rules is improved considerable when the derivative extended filter is combined with the triviality filter. The most dramatic reduction in running time can be found for *ipums.la.97*, which is as much as 90%. Theoretically, the larger the allowed number of maximum conditions on rule antecedent the more obvious the efficiency is improved. Hence, the triviality filter can be regarded as an efficient complement for the derivative extended filter.

5.3.4 Difference Set Statistics Derivation and Circular Intersection Approach

In the first step of this experiment, we ran our original algorithm with the two derivative filters in table 4.1 with the triviality filter. For databases *abalone, heart, housing, German credit* and *ipmus.la.97*, which are relatively smaller, we set the maximum number of conditions on the rule antecedents from 3 to 8, and then run the program with no limit on the maximum number of conditions allowed on rule antecedents. After this, the difference set statistics derivation approach and the circular intersection approach are introduced respectively, before the efficient algorithm in table 4.3 is ran following the same procedure. For *ipmus.la.98, ipmus.la.99, ticdata2000, census income* and *covtype*, which are relatively larger databases, we only ran the programs with maximum number of conditions allowed on rule antecedents set to 3, 4, and 5. We plot the allowed number of maximum conditions on antecedents against required running time for these programs to discover the top 1000 significant impact rules in figure 5.1 and 5.2. The pink lines with square dots show the changes in CPU time for algorithms with neither of these efficiency improving techniques. The purple lines with round dots show the results for algorithm with difference set statistics derivation only, while the yellow lines with triangular dots denote the trends brought by the algorithms with the circular intersection approach only. The results for algorithm with both techniques introduced are plotted using the dark blue lines with diamond dots.

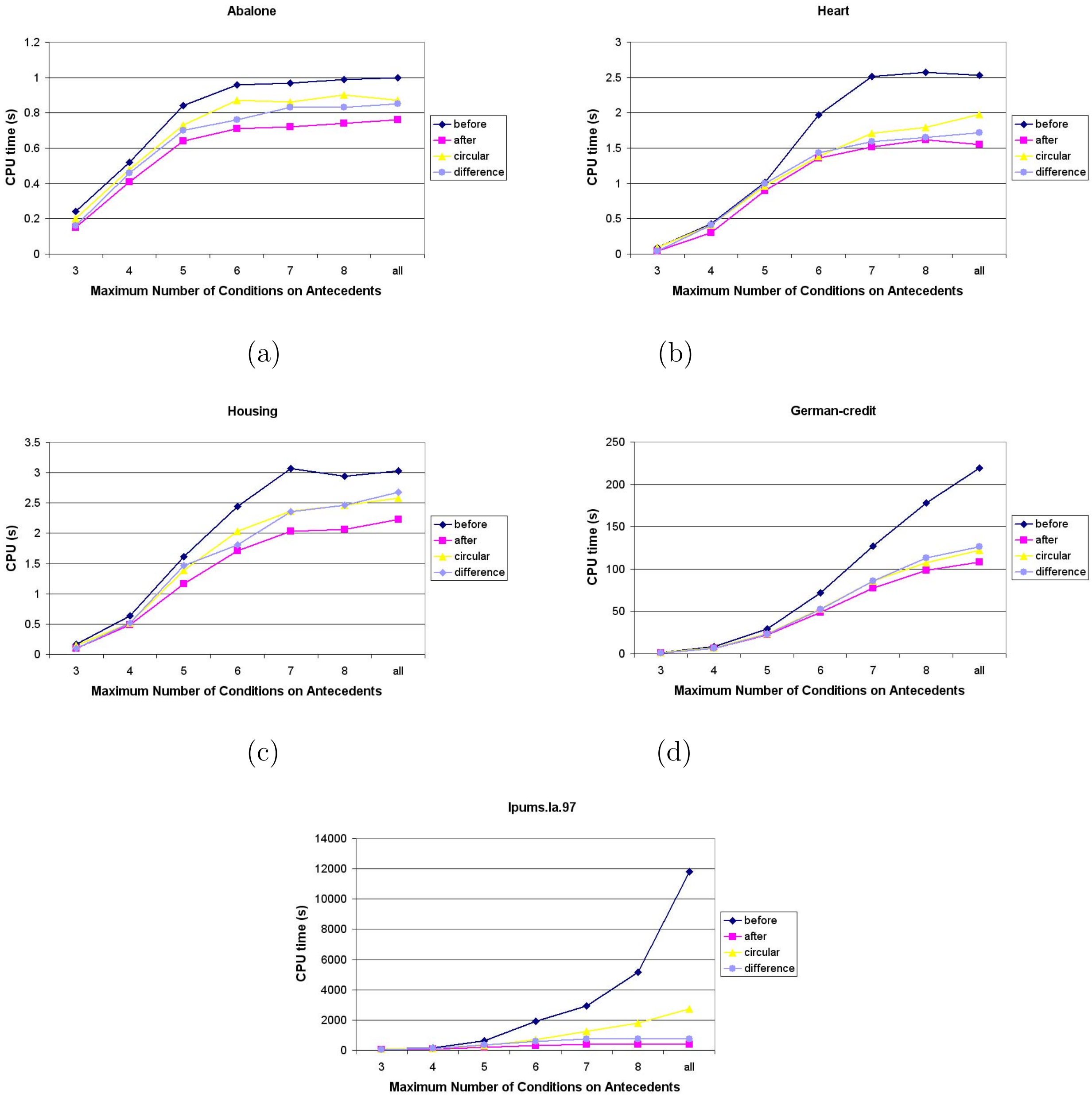
Almost every database undergoes considerable reduction in running time after the introduction of these two efficiency improving approaches. The differences in efficiency increases with the maximum number of conditions allowed on rule antecedent. When there is no limit on the maximum number of conditions on rule antecedent, CPU time spent for the OPUS_IR_Filter algorithm with the two efficiency improving techniques applied to search for top 1000 significant impact rules in *ipums.la.97* is less than one sixth of that necessary for OPUS_IR_Filter without introducing the techniques. However, necessary running time is also influenced by other factors including the size of the databases, the number of trivial rules in the top 1000 impact rule, and the number of significant rules.

After examining the effects of these two efficiency improving techniques independently, we come to the conclusion that the difference statistics derivation technique works better in some databases like *census income*; while the circular intersection approach has a greater effect on databases including *ipums.la.98*. However, the differences in effect are associated with several subtle factors including the order in the available conditions are ranked as the input of algorithm, and the order in which different parent rules are compared with the current rule to be assessed.

5.4 Conclusion of Experiments

This chapter has dealt with evaluating the practical effects of our proposed rule pruning and efficiency improving techniques, including the derivative extended and partial rule filters, the triviality filter, the difference set statistics derivation approach and the circular intersection approach. 10 databases which vary a lot in size and density are selected for the experiments. The experimental results show that derivative rules exist profoundly in resulting distributional-consequent rules. As the allowed maximum number of conditions on rule antecedents increases, so are the percentages of derivative rules among the resulting sets. Some of the databases experienced stunning changes in resulting rules after the derivative extended rule filter is applied. The derivative filters we proposed enable effective pruning of such potentially uninteresting impact rules and the derivative partial rule filter enables further reduction of resulting rules after the derivative extended filter is applied.

We also proved that using the OPUS framework for rule discovery can successfully discover distributional-consequent rules in very large, dense databases for which the Apriori based algorithms fail. A typical example is the *ipums.la.99* database, which turns out to have nearly 2000 items after discretization. For other

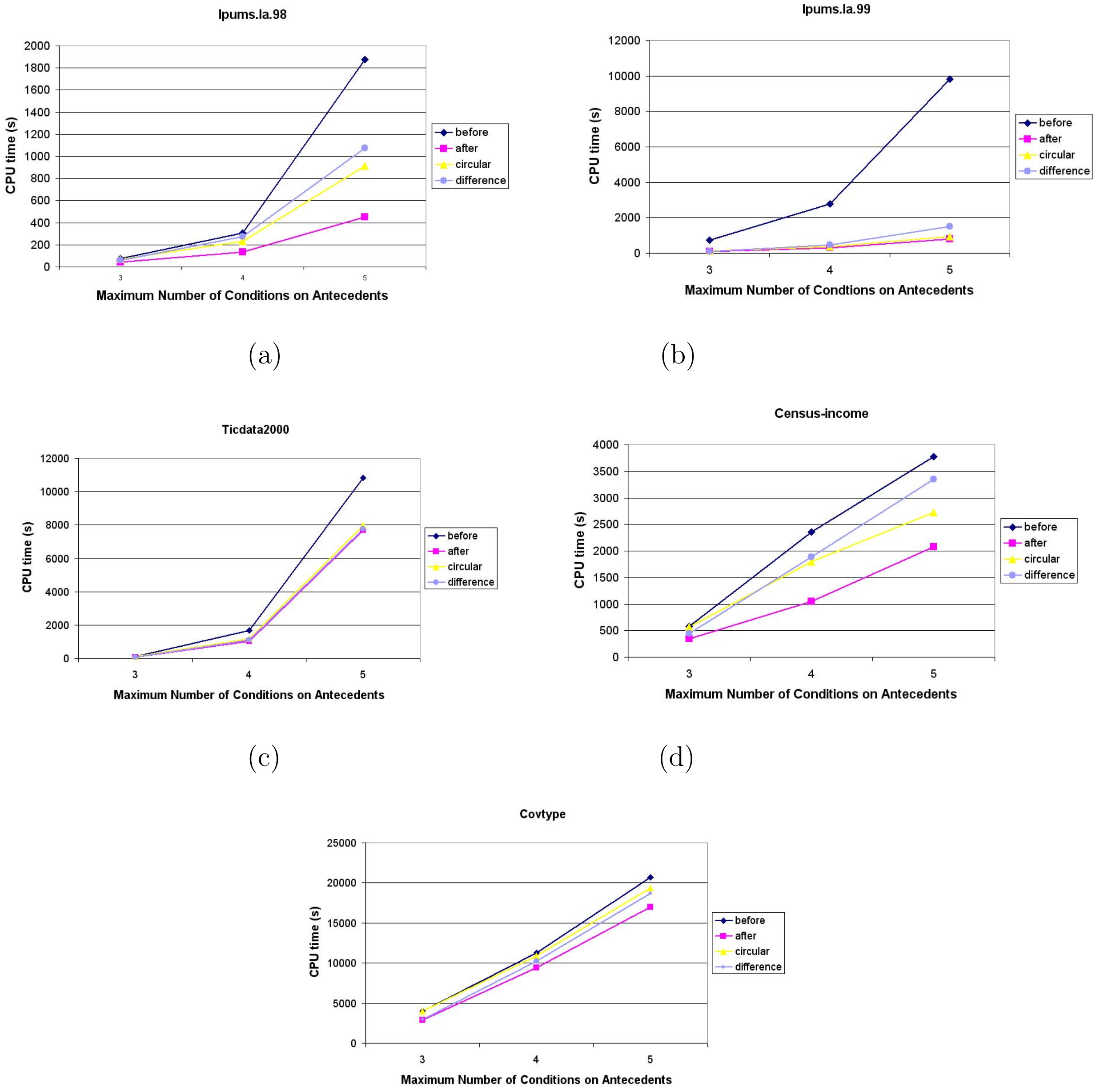


(a) (b)

(c) (d)

(e)

Figure 5.1: Comparison of Running Time before and after applying data access saving techniques for (a) *abalone*, (b) *heart*, (c) *housing*, (d) *German credit*, and (e) *ipums.la.97* with maximum number of conditions allowed on rule antecedent set to 3-8, and with no restriction on maximum number of conditions allowed on rule antecedent



(a)

(b)

(c)

(d)

(e)

Figure 5.2: Comparison of Running Time before and after applying data access saving techniques for (a) *Ipums.la.98*, (b) *ipums.la.99*, (c) *Ticdata2000*, (d) *Census income*, and (e) *covtype* with maximum number of conditions allowed on rule antecedent set to 3, 4 and 5

databases where Borgelt and Kruse (2002)'s Apriori implementation can successfully discover rules, our proposal generally discover rules in a faster manner.

The triviality filter has been justified as an alternative to derivative extended rule filter and can introduce desirable efficiency gains when functioning with the derivative extended rule filter.

The *difference set statistics derivation approach* for getting rid of data access redundancies and the *circular intersection approach* for removing intersection operation redundancies were also proved to independently and effectively reduce necessary running time for discovering significant impact rules practically. Effects are more notable when the two techniques are combined with each other.

Chapter 6

Conclusions and Future Research

By now, we have studied the techniques for performing efficient exploratory rule pruning, and have also proposed several techniques for efficiently mining interesting rules in the context of k-optimal impact rule discovery. In this chapter, we present a summarization of this thesis, highlighting our contributions. Finally, we finish up this thesis with a discussion of future research and some concluding comments.

6.1 Summary of this Thesis

This thesis committed itself to devising and implementing algorithms for efficiently and effectively rule pruning in *distributional-consequent exploratory rule discovery*, based on the analysis of the importance of it in the community of data mining.

We first explained the essentiality of *exploratory rule discovery*, of which the distributional-consequent rule discovery is a subclass. Exploratory rule discovery plays an important role in nowadays data mining, because it seeks multiple models, instead of one, that satisfy a user specified set of criteria, called *constraints.* Contrarily, traditional classification techniques, that only one model is learnt from the training data maximizing some object function of performance for prediction or classification. However, whether a model is best or not varies with the context of application. It is well recognized that in some occasions models that perform equally well coexist. This make it difficult and unsensible to select only one "best" model over the others. The characteristics of exploratory rule discovery can successfully solve this problem. A rule discovered using exploratory rule discovery consists of a *rule body* with Boolean conditions and a description with a set of measures and statistics that is comprehensible for human analysts and can be easily translated for decision making. In some applications, this feature is precious for result analysis. A famous example is the mining of *market basket data*, in which exploratory rule are able to explicitly clarify the relationship among various products and services.

Drawbacks of exploratory rule discovery come with its virtues. Mining multiple models can lead to the problem of unmanageable numbers of resulting rules, a great portion of which are potentially uninteresting. How to control the resulting rules becomes the most important issue of concern in exploratory rule discovery. Unnecessary computation is wasted on searching for uninteresting models, jeopardizing the efficiency of rule discovery. Accordingly, in the latter part of this thesis, we occupied ourselves with discussions on how to address these problems.

We classified existing exploratory rule discovery techniques into *propositional rule discovery* and *distributional-consequent rule discovery* considering their traits. Propositional rule discovery searches for rules with qualitative or discretized quantitative attributes only. Propositional rule pruning and optimization techniques are extensively studied. Plenty research has also been contributed to discretize quantitative attributes for propositional rule discovery, so as to minimize resulting information loss. Most of such techniques are constructed following a discretize-and-merge paradigm. However, since discretized quantitative attributes have lower levels of measurement scales than its undiscretized counterparts, discretization is not the best solution for describing quantitative attributes in rule discovery.

Distributional-consequent rule discovery was designed to overcome the limitations of propositional rule discovery. *Quantitative association rule discovery* proposed by Aumann and Lindell (1999) is a paradigmatic example. Distributional-consequent rule discovery is so termed, for the consequent of the resulting rules is a chosen target quantitative variable (or set of variables) of user interests, described using distributional statistics.

Aumann and Lindell (1999)'s implementation of distributional-consequent rule discovery was contrived based on the frequent itemset framework, which is inherently unsuitable for performing rule discovery in very dense databases, due to the prohibitive memory requirements. We referred ourselves to the OPUS framework instead, as the foundation of our implementation of impact rule discovery (a new name for quantitative association rule discovery given by Webb (2001) to avoid confusions with the quantitative association rule discovery proposed by Srikant and Agrawal (1996). Since our implementation commits depth-first transversal of a tree-style search space, and applies the *branch and bound* techniques for more effective search space pruning, it manages to discover rules in large and dense databases for which Aumann and Lindell (1999)'s algorithm fails.

To answer the question of why our research was oriented to developing efficient distributional-consequent rule pruning techniques, we explained the similarities and differences between propositional rule discovery and distributional-consequent rule discovery in chapter 3. The differences in descriptive natures of these two types of techniques determine that rule pruning techniques designed specially for propositional rule discovery cannot be transplanted directly to function in distributional-consequent rule discovery. However, comparing with the popularity in propositional rule pruning development, research on rule pruning with distributional-consequent rule discovery is limited. We also argued that it is crucial to improve efficiency of distributional-consequent rule discovery inasmuch as extra computation and data accesses are rooted in collecting rule descriptions.

We reviewed existing propositional rule pruning techniques in chapter 3. Constraints with properties, including anti-monotonicity, monotonicity, succinctness, can be utilized for performing powerful search space pruning, and enhancing the efficiency of rule discovery. The constraint-based techniques can be applied to distributional-consequent rule discovery with adaptions. Some of the measures for proportional rule interestingness are applicable in distributional-consequent rules. Investigations into techniques for deriving compact representations of resulting rules are among the key issues of our research. We asserted that mining maximal frequent itemsets is not desirable in distributional-consequent rule discovery, in respect that the most specific distributional-consequent rules do not imply useful information about relationship with respect to the target quantitative variable. The closed set techniques agree only with propositional rule discovery, too. Techniques regarding statistical rule significance are interesting topics for distributional-consequent rule pruning.

How can we efficiently and effectively prune potentially uninteresting distributional-consequent rules according to the above observations? We implemented two derivative rule filters in OPUS based k-optimal impact rule discovery. The derivative extended rule filter is proposed for pruning rules that can somehow be derived from its parents. It is an efficient variant of the insignificant quantitative association rule pruning proposed by Aumann and Lindell (1999). The second filter, the derivative partial rule filter, is a brand new technique for pruning rules that are misleading with the presence of its children. Derivative partial rules have shown to be extremely common in resulting significant rules (resulting rules that have been pruned using the derivative extended rule filter). Such derivative partial rules cannot be discarded using previous rule pruning techniques.

We have demonstrated in chapter 3 that the efficiency impediment in distributional rule discovery is critical. To employ the derivative rule filters, massive additional computational and I/O expenses are essential for identifying the derivability of rules. We proposed the *triviality filter* as a venue of attack. Trivial rules are special derivative extended rules, which cover the same set of records as that covered by at least one of their corresponding parents. By utilizing the anti-monotonicity of the non-trivial constraint, the rule discovery process can be expedited.

We also detected excessive redundancies in our implementation of derivative rule pruning algorithm. We proposed an efficiency improving technique by deriving difference set statistics for identifying whether a rule is derivative or not with available statistics to eliminate unnecessary data accesses. A new approach for deriving the coversets of parent rules for identifying whether a rule is derivative or not, named the *circular intersection approach*, was employed in place of the original *parallel intersection approach* for the purpose of reducing the redundancies in intersection operations.

To attest our theoretical analysis and arguements for desirable expectations of the efficiency and effectiveness of our proposed techniques, we conducted empirical evaluations with ten representative databases. These databases cover a wide range of size and density. Experiments demonstrated dramatic impact on the rules discovered when the derivative extended rule filter was applied. As much as 99% of the rules otherwise discovered were shown to be derivative extended. The implementation of the derivative partial rule filter also witnessed further removal of many resulting significant rules which are further proved to be "derivative" with respect to any of their children. Comparisons with the efficient Apriori implementation supported our declaration about the efficiency of the OPUS_IR_Filter algorithms by showing that impact rules in the selected databases are successfully discovered in reasonable period of time. While Borgelt and Kruse (2002)'s Apriori, which is one of the most efficient implementation, stopped midways during the course of rule discovery in some of the dense databases.

The efficiency gains brought by incorporating the triviality filter provide empirical support for our statement that triviality filter is a complement for the derivative extended rule filter. Without affecting the results produced, rule discovery time was reduced by as much as 90% (for *ipums.la.97*). Experiments were also done to evaluate the practical effects of the difference set statistics derivation and the circular intersection approaches, for which conclusions can be drawn that dramatic reductions in running time are experienced by our impact rule discovery algorithm with these efficiency improving schemes, with running time being reduced by up to 90% for some of the databases.

6.2 Future Research

Although light has been shed on efficiently and effectively mining interesting rules with undiscretized quantitative attributes on rule consequents: the distributional-consequent rule discovery in this thesis, our studies also bring before us some research topics that deserve further attention.

1. Our discussions have been restricted primarily on mining rules with a single undiscretized quantitative attribute in the consequents. However, it would be helpful to extend single target variable in the rules to a set of arbitrary number of quantitative target attributes. The principle matter facing us for implementing this idea is to find interestingness measures or statistics for describing the target variables. Computational costs are expected to be overwhelming. Therefore, techniques for efficiency improvement may attract most attention.

2. No effort has been bestowed on optimally introducing quantitative attributes into distributional-consequent rule antecedents, either discretized or undiscretized, in our research. Discretized-and-merge structures can be employed, yet the measures of information loss for distributional-consequent rules are to be devised. Another possible solution is to have undiscretized quantitative

attributes in rule antecedent and present the resulting rules using novel methods of presentations. Visualizations are applicable. In this way, information loss can be minimized.

3. After examining the resulting rules generated using our algorithm, we discovered the existence of multivariate qualitative attributes which are allowed to have hundreds of values, is often associated with the production of very large numbers of rules. For example, the *occupation* in the *ipums* series databases can take more than 160 different values. Similarities among these qualitative attributes cannot be captured and the number of rules increases consequently. By clustering different qualitative values according to their relationships with other attributes, especially the target variable of interest, resulting rule sets can be compressed in a straight forward manner.

4. The *t-test*, which we applied in our research for comparing the means of two independent samples, is a parametric test. For parametric tests, we assume that the populations from which the samples are drawn are approximately normally distributed, or we rely on the central limit theorem to give us a normal approximation. Since we have no guarantee that the populations uniformly take normal distributions, non-parametric tests, for which no such assumptions are imposed, are better alternatives for this purpose. Zhang, Padmanabhan and Tuzhilin (2004) have proposed significant market share rule discovery with non-parametric tests, yet their research was tailored for propositional rule discovery only. It would be interesting to explore how their proposals might be transferred to impact rule discovery.

6.3 Concluding Comments

We have investigated efficient techniques for pruning set of rules with undiscretized quantitative attributes in the consequents, which we call *distributional-consequent rule* pruning. This topic attracted our attention because mining multiple models that satisfy a given set of constraints in exploratory rule discovery may generate too many rules for users to analyze and may lead to serious efficiency problems. Moreover, existing research aiming at efficiently discovering and controlling sets of resulting rules for user analysis are mostly devoted to propositional exploratory rule discovery, which mines interrelationships among qualitative attributes or discretized quantitative attributes. However, it is well recognized that discretization may lead to unavoidable information loss. Considering quantitative attributes exist in many databases and require effective analysis, distributional-consequent rules are proposed, which minimize potential information loss by describing a user-specified quantitative variable using its distribution.

Rule pruning techniques for distributional-consequent rule discovery and propositional rule discovery are different due to the dissimilarities between their descriptive natures. Thus, we devoted ourselves to developing effective and efficient rule pruning techniques for OPUS based k-optimal impact rule discovery, which is a typical type of distributional-consequent rule discovery. We first proposed an algorithm for removing derivative extended impact rules which delivers better performance over previous frequent itemset based techniques. Experimental results showed that our algorithm can successfully remove up to 99% of potentially uninteresting impact rules in some databases and can work with very dense databases which previous techniques fail. We also proposed the definition of an other type of derivative rules which we called derivative partial impact rules. Derivative partial impact rules, which no existing techniques has been developed to discard, are potentially uninteresting rules that can somehow be derived from their children. Our derivative partial rule filter identified as much as 34% potentially uninteresting rules in the resulting set.

It was also recognized that the efficiency problem with distributional-consequent rule discovery is much more serious than that for propositional rule discovery. The situation is much worse when the derivative rule filters are applied, because considerable computation and data accesses are required for collecting necessary statistics for rule descriptions and implementing the filters. We proposed the *triviality filter* which can remove a subset of derivative extended rules during rule discovery and can speed up the efficiency for discovering significant impact rules without affecting the results, when combined with the derivative extended rule filter. Two other efficiency improving techniques, the *difference set statistics derivation* and the *circular intersection* approaches were also proposed for eliminating redundant computation and data accesses. Our algorithms when integrated with these efficiency improving techniques demonstrated substantial reduction in running time for discovering significant impact rules, in some cases reducing computation by as much as 90

Both theoretical analysis about the importance of effective and efficient distributional-consequent rule pruning together with the outstanding practical performances of our proposed algorithms give us grounds for being positive that our research is of great utility for mining interrelationships with undiscretized attributes in very large, dense databases. We hope that the work described in this thesis may lay an excellent foundation for future research in this context.

References

Abbattista, F., Degemmis, M., Licchelli, O., Lops, P., Semeraro, G. and Zambetta, F. (2002). Improving the usability of an e-commerce web site through personalization, *in* F. Ricci and B. Smyth (eds), *Proceedings of RPeC'02, Malaga, Spain*, pp. 20–29.

Agarwal, R. C., Aggarwal, C. C. and Prasad, V. V. V. (2000). Depth first generation of long patterns, *Knowledge Discovery and Data Mining*, pp. 108–118.

Agarwal, R. C., Aggarwal, C. C. and Prasad, V. V. V. (2001). A tree projection algorithm for generation of frequent item sets, *Journal of Parallel and Distributed Computing* **61**(3): 350–371.

Aggarwal, C. C. and Yu, P. S. (1998). A new framework for itemset generation, *PODS '98: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pp. 18–24.

Agrawal, R., Imielinski, T. and Swami, A. (1993). Mining association rules between sets of items in large databases, *in* P. Buneman and S. Jajodia (eds), *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, D.C., pp. 207–216.

Agrawal, R. and Shafer, J. C. (1996). Parallel mining of association rules, *IEEE Transaction On Knowledge and Data Engineering* **8**: 962–969.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules, *in* J. B. Bocca, M. Jarke and C. Zaniolo (eds), *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, Morgan Kaufmann, pp. 487–499.

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns, *in* P. S. Yu and A. S. P. Chen (eds), *Eleventh International Conference on Data Engineering*, IEEE Computer Society Press, Taipei, Taiwan, pp. 3–14.

Ahonen, H., Heinonen, O., Klemettinen, M. and Verkamo, A. I. (1998). Applying data mining techniques for descriptive phrase extraction in digital document collections, *Advances in Digital Libraries*, pp. 2–11.

Ale, J. M. and Rossi, G. H. (2000). An approach to discovering temporal association rules, *Proceedings of the 2000 ACM symposium on Applied computing*, ACM Press, pp. 294–300.

Alhammady, H. and Ramamohanarao, K. (2004). The application of emerging patterns for improving the quality of rare-class classification., *Proceedings of The Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD2004)*, pp. 207–211.

Aumann, Y. and Lindell, Y. (1999). A statistical theory for quantitative association rules, *Knowledge Discovery and Data Mining*, pp. 261–270.

Baralis, E. and Psaila, G. (1997). Designing templates for mining association rules, *Journal of Intelligent Information Systems* **9**(1): 7–32.

Bastide, Y., Pasquier, N., Taouil, R., Stumme, G. and Lakhal, L. (2000). Mining minimal non-redundant association rules using frequent closed itemsets, *Lecture Notes in Computer Science* **1861**: 972–986.

Bay, S. D. (1999). The UCI KDD archives [http://kdd.ics.uci.edu].
URL: *http://www.ics.uci.edu/∼mlearn/MLRepository.html*

Bay, S. D. and Pazzani, M. J. (2001). Detecting group differences: Mining contrast sets, *Data Mining and Knowledge Discovery*, pp. 213–246.

Bayardo, Jr., R. J., Agrawal, R. and Gunopulos, D. (1999). Constraint-based rule mining in large, dense databases, *ICDE '99: Proceedings of the 15th International Conference on Data Engineering*, IEEE Computer Society, pp. 188–197.

Bayardo, R. J. (1998). Efficiently mining long patterns from databases, *in* L. M. Haas and A. Tiwary (eds), *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA*, ACM Press, pp. 85–93.

Bench-Capon, T., Coenen, F. and Leng, P. (2000). An experiment in discovering association rules in the legal domain, *DEXA '00: Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, IEEE Computer Society, pp. 1056–1060.

Bettini, C., Wang, X. S. and Jajodia, S. (1998). Mining temporal relationships with multiple granularities in time sequences, *Data Engineering Bulletin* **21**(1): 32–38.

Bhattacharyya, R. A. J. . G. K. (2000). *Statistics: Principles and Methods*, fourth edition edn, John Wiley and sons, Inc, New York, US.

Blake, C. and Merz, C. (1998). UCI repository of machine learning databases.
URL: *http://www.ics.uci.edu/∼mlearn/MLRepository.html*

Bonchi, F. and Geothals, B. (2004). Fp-bonsai: the art of growing and pruning small fp-trees, *The Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04)*, pp. 155–160.

Bonchi, F., Giannotti, F., Mazzanti, A. and Pedreschi, D. (2003). Exante: Anticipated data reduction in constrained pattern mining., *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pp. 59–70.

Borgelt, C. and Kruse, R. (2002). Induction of association rules: Apriori implementation, *Proceedings of the 15th Conference on Computational Statistics (Compstat 2002, Berlin, Germany)*, Physika Verlag, Heidelberg, Germany.

Brijs, T., Swinnen, G., Vanhoof, K. and Wets, G. (1999). Using association rules for product assortment decisions: a case study, *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 254–260.

Brin, S., Motwani, R. and Silverstein, C. (1997). Beyond market baskets: Generalizing association rules to correlations, *in* J. Peckham (ed.), *SIGMOD 1997,* *Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*, ACM Press, pp. 265–276.

Brin, S., Motwani, R., Ullman, J. D. and Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data, *in* J. Peckham (ed.), *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*, ACM Press, pp. 255–264.

Brin, S., Rastogi, R. and Shim, K. (1999). Mining optimized gain rules for numeric attributes, *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, pp. 135–144.

Burdick, D., Calimlim, M. and Gehrke, J. (2001). MAFIA: A maximal frequent itemset algorithm for transactional databases, *Proceedings of the 17th International Conference on Data Engineering*, IEEE Computer Society, Washington, DC, pp. 443 – 452.

Carter, C. L., Hamilton, H. J. and Cercone, N. (1997). Share based measures for itemsets, *PKDD '97: Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery*, Springer-Verlag, pp. 14–24.

Chou, P. B., Grossman, E., Gunopulos, D. and Kamesam, P. (2000). Identifying prospective customers, *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, pp. 447–456.

Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J. D. and Yang, C. (2001). Finding interesting associations without support pruning, *Knowledge and Data Engineering* **13**(1): 64–78.

Cooley, R. W. (2000). Web usage mining: Discovery and application of interesting patterns from web data. Ph.D. Thesis. University of Minnesota. May 2000.

Cuppens, F. and Mige, A. (2002). Alert correlation in a cooperative intrusion detection framework, *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, IEEE Computer Society, pp. 202–215.

Doddi S, Marathe A, R. S. T. D. (2001). Discovery of association rules in medical data, *Med Inform Internet Med*, Vol. 26(1), pp. 25–33.

Dong, G. and Li, J. (1998). Interestingness of discovered association rules in terms of neighborhood-based unexpectedness, *in* X. Wu, Kotagiri Ramamohanarao and K. B. Korb (eds), *Research and Development in Knowledge Discovery and Data Mining, Proc. 2nd Pacific-Asia Conf. Knowledge Discovery and Data Mining, PAKDD*, Vol. 1394, Springer, pp. 72–86.

Dong, G. and Li, J. (1999). Efficient mining of emerging patterns: discovering trends and differences, *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, pp. 43–52.

Dong, J., Perrizo, W., Ding, Q. and Zhou, J. (2000). The application of association rule mining to remotely sensed data, *Proceedings of the 2000 ACM symposium on Applied computing*, ACM Press, pp. 340–345.

Dua, S., Cho, E. and Iyengar, S. (2000). Discovery of web frequent patterns and user characteristics from web access logs: a framework for dynamic web personalization, pp. 3–8.

Fonseca, B. M., Golgher, P. B., de Moura, E. S. and Ziviani, N. (2003). Using association rules to discover search engines related queries, *Proceedings of the First Latin American Web Congress (LA-WEB'03)*, IEEE Computer Society, pp. 66–71.

Fukuda, T., Morimoto, Y., Morishita, S. and Tokuyama, T. (1996a). Data mining using two-dimensional optimized association rules: scheme, algorithms, and visualization, *SIGMOD Rec.* **25**(2): 13–23.

Fukuda, T., Morimoto, Y., Morishita, S. and Tokuyama, T. (1996b). Mining optimized association rules for numeric attributes, *PODS '96: Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pp. 182–191.

Governatori, G. and Stranieri, A. (2001). Towards the application of association rules for defeasible rule discovery, *in* R. L. . A. M. B. Verheij, A. Lodder (ed.), *Frontiers in Artificial Intelligence and Applications*, Vol. 70, IOS Press, pp. 66–75. Proceedings of JURIX 2001.

Graaf, J. M. D., Kosters, W. A. and Witteman, J. J. (2000). Interesting association rules in multiple taxonomies, *in* A. van den Bosch and H. Weigand (eds), *Proceedings of the Twelfth Belgium-Netherlands Articial Intelligence Conference (BNAIC'00)*, pp. 93–100.

Grahne, G., Lakshmanan, L. V. S. and Wang, X. (2000). Efficient mining of constrained correlated sets, *Proceedings of the Sixteenth International Conference on Data Engineering*, pp. 512–521.

Gray, B. and Orlowska, M. E. (1998). Ccaiia: Clustering categorial attributed into interseting accociation rules, *PAKDD '98: Proceedings of the Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining*, Springer-Verlag, pp. 132–143.

Gunopulos, D., Mannila, H. and Saluja, S. (1997). Discovering all most specific sentences by randomized algorithms, *Proceedings of the 6th International Conference on Database Theory*, pp. 215–229.

Han, J., Dong, G. and Yin, Y. (1999). Efficient mining of partial periodic patterns in time series database, *Fifteenth International Conference on Data Engineering*, IEEE Computer Society, Sydney, Australia, pp. 106–115.

Han, J. and Fu, Y. (1995). Discovery of multiple-level association rules from large databases, *Proc. of 1995 International Conference on Very Large Data Bases (VLDB'95), Zürich, Switzerland, September 1995*, pp. 420–431.

Han, J. and Kamber, M. (2001). *Data mining : concepts and techniques*, Morgan Kaufmann, San Francisco, CA.

Han, J., Koperski, K. and Stefanovic, N. (1997). Geominer: a system prototype for spatial data mining, *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pp. 553–556.

Han, J., Pei, J. and Yin, Y. (2000). Mining frequent patterns without candidate generation, *in* W. Chen, J. Naughton and P. A. Bernstein (eds), *2000 ACM SIGMOD International Conference on Management of Data*, ACM Press, pp. 1–12.

Harrison, D. and Rubinfeld, D. (1978). Hedonic prices and the demand for clean air, *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, Vol. 5, pp. 81–102.

Hipp, J., Güntzer, U. and Grimmer, U. (2001). Integrating association rule mining algorithms with relational database systems., *Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS)*, pp. 130–137.

IBM (1996). *IBM Intelligent Miner User's Guide*, International Business Machines. Version 1, Release 1.

Johnson, R. (1996). *Elementary Statistics*, seventh edition edn, An International Thomson Publishing company, New York, US.

Johnston, B. and Governatori, G. (2003). An algorithm for the induction of defeasible logic theories from databases, *Proceedings of the Fourteenth Australasian database conference on Database technologies 2003*, Australian Computer Society, Inc., pp. 75–83.

Kawano, H. and Hasegawa, T. (1998). Mondou: Interface with text data mining for web search engine., *Thirty-First Annual Hawaii International Conference on System Sciences (HICSS '98)*, pp. 275–283.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H. and Verkamo, A. I. (1994). Finding interesting rules from large sets of discovered association rules, *in* N. R. Adam, B. K. Bhargava and Y. Yesha (eds), *Third International Conference on Information and Knowledge Management (CIKM'94)*, ACM Press, pp. 401–407.

Koperski, K. and Han, J. (1995). Discovery of spatial association rules in geographic information databases, *Proceedings of the 4th International Symposium on Advances in Spatial Databases*, Springer-Verlag, pp. 47–66.

Koperski, K., Han, J. and Adhikary, J. (1998). Mining knowledge in geographical data, *Communications of the ACM* **26(1)**: 65–74.

Lakshmanan, L. V. S., Ng, R., Han, J. and Pang, A. (1999). Optimization of constrained frequent set queries with 2-variable constraints, *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, ACM Press, pp. 157–168.

Lawler, E. L. and Wood, D. E. (1966). Branch and bound methods: A survey, *Operations Research*, Vol. 14, pp. 699–719.

Lee, W. and Stolfo, S. (1998). Data mining approaches for intrusion detection, *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX.

Lee, W., Stolfo, S. J. and Mok, K. W. (1999). A data mining framework for building intrusion detection models, *IEEE Symposium on Security and Privacy*, pp. 120–132.

Li, J., Zhang, X., Dong, G., Ramamohanarao, K. and Sun, Q. (1999). Efficient mining of high confidence association rules without support thresholds, *in* J. Zytkow and J. Rauch (eds), *Principles of Data Mining and Knowledge Discovery PKDD'99, LNAI 1704, Prague, Czech Republic*, Springer-Verlag, pp. 406–411.

Lin, D.-I. and Kedem, Z. M. (1998). Pincer search: A new algorithm for discovering the maximum frequent set, *Lecture Notes in Computer Science* **1377**: 105–119.

Lin, J.-L. and Dunham, M. H. (1998). Mining association rules: Anti-skew algorithms, *Proceedings of the Fourteenth International Conference on Data Engineering, February 23-27, 1998, Orlando, Florida, USA*, IEEE Computer Society, pp. 486–493.

Liu, B., Hsu, W. and Ma, Y. (1999a). Mining association rules with multiple minimum supports, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-99)*, pp. 337–341.

Liu, B., Hsu, W. and Ma, Y. (1999b). Pruning and summarizing the discovered associations, *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 125–134.

Liu, B., Ma, Y., Wong, C. K. and Yu, P. S. (2003). Scoring the data using association rules, *Applied Intelligence* **18**(2): 119–135.

Lo, C. and Ng, V. (1999). Discovering web access orders with association rules, *Systems, Man, and Cybernetics 1999. IEEE the International Conference on Systems Management and Cybernetics*, Vol. 4, pp. 99–104.

Loh, S., Wives, L. K. and de Oliveira, J. P. M. (2000). Concept-based knowledge discovery in texts extracted from the web, *SIGKDD Explor. Newsl.* **2**(1): 29–39.

Lu, H., Han, J. and Feng, L. (1998). Stock movement prediction and n-dimensional inter-transaction association rules, *In Proceedings of ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Seattle, Washington, June 1998.*, pp. 12:1–12:7.

Ma, Y., Liu, B. and Wong, C. K. (2000). Web for data mining: organizing and interpreting the discovered rules using the web, *SIGKDD Explorations* **2**(1): 16–23.

Ma, Y., Liu, B., Wong, C. K., Yu, P. S. and Lee, S. M. (2000). Targeting the right students using data mining, *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, pp. 457–464.

Mannila, H., Toivonen, H. and Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences, *Data Mining and Knowledge Discovery* **1**(3): 259–289.

Meo, R., Psaila, G. and Ceri, S. (1996). A new sql-like operator for mining association rules, *VLDB '96: Proceedings of the 22th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., pp. 122–133.

Michail, A. (1999). Data mining library reuse patterns in user-selected applications, *Proceedings of the 14th IEEE International Conference on Automated Software Engineering*, IEEE Computer Society, pp. 24–33.

Michail, A. (2000). Data mining library reuse patterns using generalized association rules, *Proceedings of the 22nd international conference on Software engineering*, ACM Press, pp. 167–176.

Mueller, A. (1995). Fast sequential and parallel algorithms for association rule mining: A comparison, *Technical Report CS-TR-3515*, College Park, MD.

Nash, W. J., Sellers, T. L., Talbot, S. R., Cawthorn, A. J. and Ford, W. B. (1994). The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. Technique Report, No. 48 (ISSN 1034-3288).

Ng, R. T., Lakshmanan, L. V. S., Han, J. and Pang, A. (1998). Exploratory mining and pruning optimizations of constrained association rules, *in* L. M. Haas and A. Tiwary (eds), *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA*, ACM Press, pp. 13–24.

Oyama, T., Kitano, K., Satou, K. and Ito, T. (2000). Mining association rules related to protein-protein interactions, *Genome Informatics*, pp. 358–359.

Ozden, B., Ramaswamy, S. and Silberschatz, A. (1998). Cyclic association rules, *ICDE '98: Proceedings of the Fourteenth International Conference on Data Engineering*, IEEE Computer Society, pp. 412–421.

Park, J. S., Chen, M.-S. and Yu, P. S. (1995a). An effective hash based algorithm for mining association rules, *in* M. J. Carey and D. A. Schneider (eds), *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, San Jose, California, pp. 175–186.

Park, J. S., Chen, M.-S. and Yu, P. S. (1995b). Efficient parallel data mining for association rules, *Proceedings of the 4th Int'l Conf. on Information and Knowledge Management*, pp. 31–36.

Pasquier, N., Bastide, Y., Taouil, R. and Lakhal, L. (1999a). Closed set based discovery of small covers for association rules, *Proc. 15emes Journees Bases de Donnees Avancees, BDA*, pp. 361–381.

Pasquier, N., Bastide, Y., Taouil, R. and Lakhal, L. (1999b). Discovering frequent closed itemsets for association rules, *Lecture Notes in Computer Science* **1540**: 398–416.

Pei, J., Han, J. and Lakshmanan, L. V. (2001). Mining frequent itemsets with convertible constraints, *Proceedings of the 17th International Conference on Data Engineering*, IEEE Computer Society, pp. 433–442.

Pei, J., Han, J. and Mao, R. (2000). CLOSET: An efficient algorithm for mining frequent closed itemsets, *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 21–30.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules, *Knowledge Discovery in Databases*, AAAI/MIT Press, pp. 229–248.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., San Mateo, Calif.

Rastogi, R. and Shim, K. (2001). Mining optimized support rules for numeric attributes, *Information Systems* **26**(6): 425–444.

Roberto J. Bayardo, J. and Agrawal, R. (1999). Mining the most interesting rules, *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, pp. 145–154.

Ruggles, S. and Sobek, M. (1997). Integrated public use microdata series: version 2.0.
URL: *http://www.ipums.umn.edu/*

Satou, K. (1997). Finding association rules on heterogeneous genome data, *Proc. Pacific Symposium on Biocomputing '97*, pp. 397–408.

Savasere, A., Omiecinski, E. and Navathe, S. B. (1995). An efficient algorithm for mining association rules in large databases, *The VLDB Journal*, Morgan Kaufmann Publishers Inc., pp. 432–444.

Seno, M. and Karypis, G. (2001). Lpminer: An algorithm for finding frequent itemsets using length decreasing support constraint, *in* N. Cercone, T. Y. Lin and X. Wu (eds), *Proceedings of the 2001 IEEE International Conference on Data Mining, 29 November - 2 December 2001, San Jose, California, USA*, IEEE Computer Society, pp. 505–512.

Shenoy, P., Haritsa, J. R., Sundarshan, S., Bhalotia, G., Bawa, M. and Shah, D. (2000). Turbo-charging vertical mining of large databases, *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, ACM Press, pp. 22–33.

Silverstein, C., Brin, S., Motwani, R. and Ullman, J. D. (2000). Scalable techniques for mining causal structures, *Data Mining and Knowledge Discovery* **4**(2/3): 163–192.

Srikant, R. and Agrawal, R. (1996). Mining quantitative association rules in large relational tables, *in* H. V. Jagadish and I. S. Mumick (eds), *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Quebec, Canada, pp. 1–12.

Srikant, R. and Agrawal, R. (1997). Mining generalized association rules, *Future Generation Computer Systems* **13**(2–3): 161–180.

Srikant, R., Vu, Q. and Agrawal, R. (1997). Mining association rules with item constraints, *in* D. Heckerman, H. Mannila, D. Pregibon and R. Uthurusamy (eds), *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining, KDD*, AAAI Press, pp. 67–73.

Tao, F., Murtagh, F. and Farid, M. (2003). Weighted association rule mining using weighted support and significance framework, *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 661–666.

Toivonen, H. (1996). Sampling large databases for association rules, *in* T. M. Vijayaraman, A. P. Buchmann, C. Mohan and N. L. Sarda (eds), *Proceedings of 1996 International Conference on Very Large DataBases*, Morgan Kaufman, pp. 134–145.

Toivonen, H., Klemettinen, M., Ronkainen, P., Hgtijnen, K. and Mannila, H. (1995). Pruning and grouping of discovered association rules, *In Workshop Notes of the ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases*, pp. 47–52.

Toivonen, H., Onkamo, P., Hintsanen, P., Terzi, E. and Sevon, P. (2004). Data mining for gene mapping. to appear.

van der Putten, P. and van Someren, M. (2000). *CoIL Challenge 2000: The Insurance Company Case*, Sentient Machine Research, Amsterdam, Leiden Institute of Advanced Computer Science.

Wang, K., Tay, S. H. W. and Liu, B. (1998). Interestingness-based interval merger for numeric association rules, *in* R. Agrawal, P. E. Stolorz and G. Piatetsky-Shapiro (eds), *Proc. 4th Int. Conf. Knowledge Discovery and Data Mining, KDD*, AAAI Press, pp. 121–128.

Webb, G. I. (1995). OPUS: An efficient admissible algorithm for unordered search, *Journal of Artificial Intelligence Research* **3**: 431–465.

Webb, G. I. (2000). Efficient search for association rules, *The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, pp. 99–107.

Webb, G. I. (2001). Discovering associations with numeric variables, *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, pp. 383–388.

Webb, G. I. (2003). Preliminary investigations into statistically valid exploratory rule discovery, *Proceedings of the Australian Data Mining Workshop (AusDM03)*, pp. 1–9.

Webb, G. I. (2005). Statistically sound exploratory rule discovery. To appear.

Webb, G. I., Butler, S. and Newlands, D. (2003). On detecting differences between groups, *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 256–265.

Webb, G. I. and Zhang, S. (2002). Efficient techniques for removing trivial associations in association rule discovery, *in the Proceedings of the First International NAISO Congress on Autonomous Intelligent Systems, (ICAIS2002).*

Webb, G. I. and Zhang, S. (2005). k-optimal-rule-discovery, *Data Mining and Knowledge Discovery* **10**(1): 39–79.

Wetjen, T. (2002). Discovery of frequent gene patterns in microbial genomes, *TZI-Report, Technologie Zentrum Informatik (TZI) 27*, University of Bremen, Germany.

Wong, P. C., Whitney, P. and Thomas, J. (1999). Visualizing association rules for text mining, *Proceedings of the 1999 IEEE Symposium on Information Visualization*, IEEE Computer Society, pp. 120–127.

Wu, N. and Jajodia, S. (2004). Mining unexpected rules in network audit trails, *Distributed and Parallel Databases.* in review.

Yip, C. L., Loo, K. K., Kao, B., Cheung, D. W.-L. and Cheng, C. K. (1999). LGen - a lattice-based candidate set generation algorithm for I/O efficient association rule mining, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 54–63.

Zaki, M. J. (2000). Generating non-redundant association rules, *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 34–43.

Zaki, M. J. and Hsiao, C.-J. (1999). Charm: an efficient algorithm for closed association rule mining, *Technical Report 99-10*, Rensselaer Polytechnic Institute.

Zaki, M. J., Parthasarathy, S. and Li, W. (1997). A localized algorithm for parallel association mining, *ACM Symposium on Parallel Algorithms and Architectures*, pp. 321–330.

Zaki, M. J., Parthasarathy, S., Ogihara, M. and Li, W. (1997a). New algorithms for fast discovery of association rules, *Technical Report TR651*, University of Rochester.

Zaki, M. J., Parthasarathy, S., Ogihara, M. and Li, W. (1997b). Parallel algorithms for discovery of association rules, *Data Mining and Knowledge Discovery* **1**(4): 343–373.

Zhang, H., Padmanabhan, B. and Tuzhilin, A. (2004). On the discovery of significant statistical quantitative rules, *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 374–383.

Zhong, N., Yao, Y. Y. and Ohsuga, S. (1999). Peculiarity oriented multi-database mining, *PKDD '99: Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, Springer-Verlag, pp. 136–146.

Vita

Publications arising from this thesis include:

Shiying Huang and Geoffrey I. Webb (2004), Efficiently Identifying rules' Significance. In *The 3rd Australiasian Data Mining Conference (AusDM 2004)*, Cairns, Queensland, Australia, pages 169-182.

Shiying Huang and Geoffrey I. Webb (2005), Discarding Insignificant Rules during Impact Rule Discovery in Large, Dense Databases. In proceedings of *SIAM Data Mining Conference 2005*, Newport Beach, CA, USA, pages 541-545.

Shiying Huang and Geoffrey I. Webb (2005) Pruning Derivative Partial Rules During Impact Rule Discovery. In proceedings of *The Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-05)*, Hanoi, Vietnam, pages 71-80.

Permanent Address: School of Computer Science and Software Engineering
Monash University
Australia

This thesis was typeset with LaTeX 2ε[1] by the author.

[1] LaTeX 2ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Glenn Maughan and modified by Dean Thompson and David Squire of Monash University.