Association Pattern Analysis for Pattern Pruning, Clustering and Summarization

by

Chung Lam Li

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

Waterloo, Ontario, Canada, 2008

AUTHOR'S DECLARATION

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

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

Abstract

Automatic pattern mining from databases and the analysis of the discovered patterns for useful information are important and in great demand in science, engineering and business. Today, effective pattern mining methods, such as association rule mining and pattern discovery, have been developed and widely used in various challenging industrial and business applications. These methods attempt to uncover the valuable information trapped in large collections of raw data. The patterns revealed provide significant and useful information for decision makers. Paradoxically, pattern mining itself can produce such huge amounts of data that poses a new knowledge management problem: to tackle thousands or even more patterns discovered and held in a data set. Unlike raw data, patterns often overlap, entangle and interrelate to each other in the databases. The relationship among them is usually complex and the notion of distance between them is difficult to qualify and quantify. Such phenomena pose great challenges to the existing data mining discipline. In this thesis, the analysis of patterns after their discovery by existing pattern mining methods is referred to as *pattern post-analysis* since the patterns to be analyzed are first discovered.

Due to the overwhelmingly huge volume of discovered patterns in pattern mining, it is virtually impossible for a human user to manually analyze them. Thus, the valuable trapped information in the data is shifted to a large collection of patterns. Hence, to automatically analyze the patterns discovered and present the results in a user-friendly manner such as pattern post-analysis is badly needed. This thesis attempts to solve the problems listed below. It addresses 1) the important factors contributing to the interrelating relationship among patterns and hence more accurate measurements of distances between them; 2) the objective pruning of redundant patterns from the discovered patterns; 3) the objective clustering of the patterns into coherent pattern clusters for better organization; 4) the automatic summarization of each pattern cluster for human interpretation; and 5) the application of pattern post-analysis to large database analysis and data mining.

In this thesis, the conceptualization, theoretical formulation, algorithm design and system development of pattern post-analysis of categorical or discrete-valued data is presented. It starts with presenting a natural dual relationship between patterns and data. The relationship furnishes an explicit one-to-one correspondence between a pattern and its associated data and provides a base for an effective analysis of patterns by relating them back to the data. It then discusses the important factors that differentiate patterns and formulates the notion of distances among patterns using a formal graphical approach. To accurately measure the distances between patterns and their associated data, both the samples and the attributes matched by the patterns are considered. To achieve this, the distance measure between patterns has to account for the differences of their associated data clusters at the attribute value (i.e. item) level. Furthermore, to capture the degree of variation of the items matched by patterns, entropy-based distance measures are developed. It attempts to quantify the uncertainty of the matched items. Such distances render an accurate and robust distance measurement between patterns and their associated data. To understand the properties and behaviors of the new distance measures, the mathematical relation between the new distances and the existing sample-matching distances is analytically derived.

The new pattern distances based on the dual pattern-data relationship and their related concepts are used and adapted to pattern pruning, pattern clustering and pattern summarization to furnish an integrated, flexible and generic framework for pattern post-analysis which is able to meet the challenges of today's complex real-world problems. In pattern pruning, the system defines the amount of redundancy of a pattern with respect to another pattern at the item level. Such definition generalizes the classical closed itemset pruning and maximal itemset pruning which define redundancy at the sample level. A new generalized itemset pruning method is developed using the new definition. It includes the closed and maximal itemsets as two extreme special cases and provides a control parameter for the user to adjust the tradeoff between the number of patterns being pruned and the amount of information loss after pruning. The mathematical relation between the proposed generalized itemsets and the existing closed and maximal itemsets are also given. In pattern clustering, a dual clustering method, known as simultaneous pattern and data clustering, is developed using two common yet very different types of clustering algorithms: hierarchical clustering and *k*-means clustering. Hierarchical clustering generates the entire clustering hierarchy but it is slow and not scalable. *K*-means clustering produces only a partition so it is fast and scalable. They can be used to handle most real-world situations (i.e. speed and clustering quality). The new clustering method is able to simultaneously cluster patterns as well as their associated data while maintaining an explicit pattern-data relationship. Such relationship enables subsequent analysis of individual pattern clusters through their associated data clusters. One important analysis on a pattern cluster is pattern summarization. In pattern summarization, to summarize each pattern cluster, a subset of the representative patterns will be selected for the cluster. Again, the system measures how representative a pattern is at the item level and takes into account how the patterns overlap each other. The proposed method, called AreaCover, is extended from the well-known RuleCover algorithm. The relationship between the two methods is given. AreaCover is less prone to yield large, trivial patterns (large patterns may cause summary that is too general and not informative enough), and the resulting summary is more concise (with less duplicated attribute values among summary patterns) and more informative (describing more attribute values in the cluster and have longer summary patterns).

The thesis also covers the implementation of the major ideas outlined in the pattern post-analysis framework in an integrated software system. It ends with a discussion on the experimental results of pattern post-analysis on both synthetic and real-world benchmark data. Compared with the existing systems, the new methodology that this thesis presents stands out, possessing significant and superior characteristics in pattern post-analysis and decision support.

Acknowledgements

First of all, my deepest and highest praise to my Lord Christ Jesus for His unfailing love and amazing grace.

During my Ph. D. years at Waterloo, a great number of people have contributed much to my making this academic journey.

First of all, my deep gratitude goes to my supervisor, Professor Andrew K. C. Wong, for his continuous, life-long enthusiasm, encouragement, support and understanding during the whole course of the program. From Professor Wong, I not only learned how to do research but also learned how to pursue my dream and vision. He teaches me not by what he says but by what he does everyday – He is a man who does what he says. I learned from him how important a vision is in research and in life, which may very well be some of the most valuable lessons that I take with me from University of Waterloo. In the toughest times, his enthusiasm encouraged me and gave me courage to take the challenges. He is willing to share my burdens and pressures, and challenges me to go further. I am so grateful that I have a supervisor who wholeheartedly guides me and teaches me.

I would like to thank too Prof. Sherman Shen for his support on my Ph. D. studies.

My sincere thanks also go to Professor Xindong Wu, my external examiner from the University of Vermont and other members of my oral defense committee: Professor Daniel Stashuk of System Design Engineering, Professor En-hui Yang and Professor Fakhri Karray of Electrical and Computer Engineering, for their helpful comments on my thesis.

I greatly appreciate the technical and social interaction with the past and current PAMI members: the opportunity for technical collaboration and research in pattern discovery with Dr. Adams Kong, the valuable research advice from Dr. Yanmin Sun, the delicious dinners with Patrick Tsui, the enjoyable research chats in remote sensing with Tarek Khalifa and Masoud Makrehchi. Thank you all for the discussion, encouragement and the countless lunches, dinners and coffee breaks we spent together.

I would like to thank my KWCAC fellowship brothers and sisters. Some of you have left Waterloo, but your friendship will be with me forever. Without your friendship and encouragement, my Waterloo life would not be so rich and colorful. A special thank-you goes to Robert Chan, who proof-read my thesis. Thanks also go to Kenneth Ng, Sabrina Ngai, Vivian Tsui, Fanny Luk, Sue Su, Rosita Kwan, Sai Kit Lo, Eric Chan, Dennis Zhuang, Simon Chan, Long Shun Cheng, Miyuki Tsukimoto, Koko Lung, Synergy Shum, Winnie Lam and Tony Ho.

My grandmother in heaven deserves special mention for her love and patience in a grandson, who after she passed away 10 years, never has a chance to give back what she has sacrificed for him.

My parents of course deserve special gratitude for their remarkable tolerance in a son who, after 30 years, has yet to contribute meaningfully to society. They are always in my heart and thoughts.

I would like to include my brother and sisters who have been a constant source of joy and love over the years in this page of acknowledgement.

During my Waterloo years, a young, brave and beautiful woman passed many lonely days in a crowded city 13,000 km away. She sacrificed so much and provided continuous support allowing her man to chase his dream. Her boundless love and endless patience are truly remarkable. My very sincere thanks therefore go to my beloved fiancée, Vivi.

To my Lord

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

The ability to automatically discover useful knowledge from data is central to human intelligence. Since data mining and knowledge discovery are concerned with the science of discovering useful knowledge from data [1] – [3], the discovery of previously unknown patterns/rules that lead to understanding in support of human comprehension, decision making and knowledge discovery has always been one of their ultimate goals. Figure 1 depicts the process of knowledge discovery which consists of an iterative sequence of steps. First of all, raw data is preprocessed to produce features relevant to a problem. Typical pre-processing tasks include data cleaning, feature selection and transformation. The features are then inputted to pattern/rule mining systems which automatically discover previously unknown patterns/rules from the preprocessed data (i.e. feature vectors). Representative examples of pattern/rule mining methods are association rule mining [4] – [9] and pattern discovery [10] – [14]. Finally, the discovered patterns/rules are post-analyzed to support further discovery of useful information.

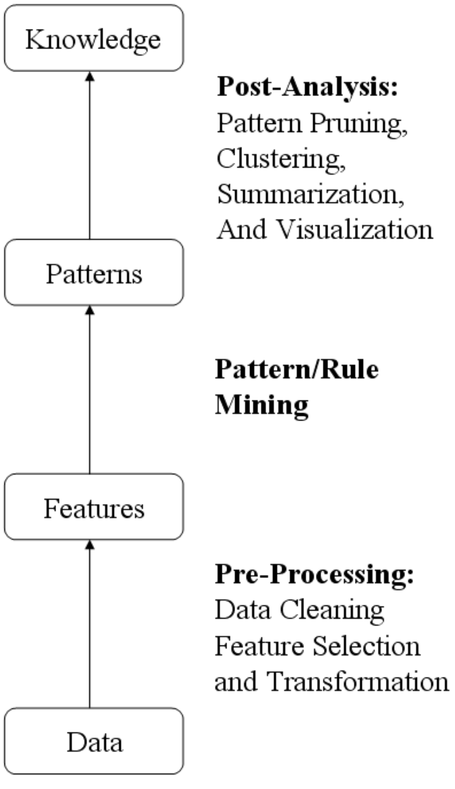


Figure 1. The process of knowledge discovery

The problem of pattern/rule mining has been extensively studied in data mining and knowledge discovery research area. There are well-developed methods and algorithms to discover different types of patterns such as association rules [4], [6] – [8], itemsets [4] – [9], correlation rules [15], [16] and association patterns [10] – [14]. In this thesis, to simplify the terminology, we collectively refer all these types of patterns or rules to as patterns unless otherwise stated. Similarly, all pattern or rule mining methods are collectively referred to as pattern mining.

Up-to-date, association rule mining [4] – [9] is the most popular rule mining technique. It is very fast and scalable to accommodate very large databases. This is particularly attractive for today's real-world problems because of the availability of huge amount of data due to fast advancement in data generation and collections technologies. It is user-friendly since the rules produced are easy to understand for human users. Furthermore, association rule mining also assumes little knowledge about the data from the users. Hence, it is relatively practical and easy to use. Since its advent, association rule mining has been widely studied in research and commonly used in industry. As the technologies of association rule mining grows, variations such as emerging patterns [17], hyperclique patterns [18], etc emerge.

One problem of association rule mining is that the use of confidence (i.e. the conditional probability of an association rule) does not capture the statistical correlation among items [15], [16]. Hence, association rule mining may discover strong rules that are not positively correlated at all (see more details in section 2.2.2). To capture correlation among items, Brin et al proposed correlation rule mining [15], [16] which uses chi-squared statistic as well as identify correlated items. Correlation rule mining is statistically sound; however, since chi-squared statistic is computed from the data of the entire contingency table, correlation rule mining is not accurate for high dimensional cases and is not scalable for large databases.

Another method to capture statistical correlation among items is pattern discovery [10] – [14]. Pattern discovery employed residual analysis to capture statistical correlation among items. It is statistically sound. Although pattern discovery is slower than association rule mining, it is faster than correlation rule mining. It is accurate for high-dimensional contingency table data and is scalable for large databases. In pattern discovery, statistical hypothesis test is employed to define and capture association patterns. As a result, each pattern discovered is associated with Type I (false positive) and Type II errors (false negative). In other words, the probability that a pattern is in fact NOT a pattern is known (Type I error). For example, at 95% significance level, such probability is 5%. Hence, the patterns discovered are highly trustable (with a very small probability of error). This property is particularly important for critical problems, such as medical diagnosis, oil and gas production and petrochemical refining, where wrong patterns will lead to huge financial loss or health hazards.

All of the above mentioned pattern mining methods suffer from the problem of rendering too many patterns or rules. Ironically, the number of patterns produced can be much more than the number of data samples. For instance, the Wine data set from UCI [19] has only 178 data samples but association rule mining can produce over 20,000 itemsets or association rules. Hence, it is very time-consuming and expensive to analyze the overwhelming number of discovered patterns. Such consequence diminishes its attractiveness as an effective knowledge discovery tool.

1.1 Problems Raised by Pattern Mining

While pattern mining has been studied extensively and new mining techniques continue to appear in literatures, the problem of the post-analysis of the discovered patterns has continually been attracting more and more attentions from both the academia and the industry. It is no doubt that pattern mining is a very important step towards knowledge discovery. However, the explosive amount of patterns discovered makes it virtually impossible for a human user to inspect them manually. Furthermore, unlike traditional data analysis where assumptions such as the independence and identical distribution (i.i.d.) assumption can be made to simplify the analysis, the discovered patterns often overlap, entangle and interrelate to each other within the data set. Hence, to my best knowledge, there is no assumption that can be made to simplify the relationship among patterns. Instead, new methods such as pattern post-analysis have to take into consideration the complex relationship among patterns. The complexity of the pattern post-analysis problem poses great challenges to existing data mining discipline – fundamental concepts, algorithms and techniques.

This thesis deals with the problems of computer-aided pattern post-analysis discussed above. Since the number of patterns discovered can be very large, post-analysis of the discovered patterns is very important for automatic knowledge acquisition from the huge volume of discovered patterns. In real-world applications, it is not uncommon to have hundreds and thousands of discovered patterns. This makes the domain experts virtually impossible to examine and comprehend all of them. In many situations, the domain experts will have to select a very small subset of patterns manually based on certain criteria such as support, confidence [4] – [9], h-confidence [18], chi-squared statistics [15], [16] and residual [10] – [14]. They have also to rely on their understanding about the patterns and their knowledge of the problem domain, etc. As the result, they will spend weeks and months to manually validate, inspect and interpret the patterns individually.

One would anticipate that such manual approach is not very effective, yet, it would be surprising to learn that it is still one of the most common approaches in industries. The situation is quite obvious. First, the criteria the domain experts used to select a small subset of patterns for further investigation are heavily based on their experiences and knowledge. A common problem in the industry (e.g. oil and gas production and petrochemical refining) is that when senior engineers retire the junior engineers may not have enough experiences and knowledge to analyze the discovered patterns. For complex systems, knowledge transfer is not easy. Second, this manual approach is very time-consuming and expensive. Usually, a team of senior experts is needed to spend a tremendous amount of times and efforts to interpret, validate and analyze the discovered patterns. Third, it is very easy for the domain experts to make mistakes during the investigation due to fatigue or other causes. Sometimes, they might be biased by their previous experiences. Since they need to look at a huge number of patterns manually for a very long period, it is almost inevitable for them to make occasional mistakes, which may lead to misleading, and, at worst, wrong conclusions. For many business applications, such consequences could cause hazards and huge financial loss. These pending problems create a huge hurdle for them to apply pattern mining systems to their problems.

In response to such challenges, this dissertation work attempts to develop a system to solve the above problems. It is intended to automatically analyze the relationship among the patterns discovered by pattern mining and organize them in a way that human being can easily comprehend, interpret and validate. Such a system, for instance, can help the junior engineers to interpret the discovered patterns easier, identify surprising patterns and obtain useful information from them. It could significantly save the amount of times and human resources as well as avoid unnecessary hazards and accidents caused by human mistakes.

To develop an effective pattern post-analysis system, the key challenges must first be identified. When compared with traditional data analysis, pattern post-analysis would encounter even greater challenges from the following standpoints:

1. the number of patterns is often much more than the number of raw data samples (the speed issue), and
2. the interrelating relationship between patterns is often more complex than the raw data (the quality issue),

While the first challenge concerns about the speed of the pattern post-analysis algorithms, the second concerns the quality of the results produced by the algorithms. An effective post-analysis method has to be fast and adequately scalable to handle a huge volume of patterns and to produce accurate results while taking the interrelating complex relationship among patterns into account. Like situations encountered in data analysis, there is always a tradeoff between speed and quality in pattern post-analysis. Hence, an effective and flexible system should allow the users to determine the tradeoff. To meet the above challenges, this thesis research focuses on the following pattern post-analysis problems:

1. The measurement of the complex interrelating relationship among patterns,
2. The automatic clustering of discovered patterns and their associated data into manageable groups,
3. The automatic pruning of redundant patterns,
4. The automatic summarization of patterns, and
5. The easefulness of the understandability of the organized patterns.

Further, to narrow down the pattern post-analysis problems, the following idealization and assumptions are imposed on the developed system.

1. All the attributes (variables) describing the data assume categorical or discrete values.
2. The number of samples in the data set is fixed and does not change during the pattern mining and post-analysis process.
3. The number of attributes describing the data set is finite. The domain of each attribute is also finite.
4. There is no order among the attributes.
5. There is no order among the samples.

Simply put, the format of the data is a relational table commonly used in machine learning and data mining settings. It should be noted that a transaction database used in association rule mining can always be mapped to an equivalent relational table (see section 2.4 for details). Such mapping is commonly used in association rule mining (e.g. see [20] – [22]). Assumptions 4 and 5 follow the formalism of the analysis of pattern distance measures in chapter 3.

It is not assumed that the data set is noise-free, complete or correct, or that the background knowledge of the domain and a discovering guide are available.

1.2 Motivation and Objectives

1.2.1 Thesis Motivation

Data is an extremely valuable asset,

but like a cash crop,

unless harvested, it is wasted.

– Sid Adelman

The importance of pattern mining has been repeatedly emphasized by a number of researchers and will not be repeated here. In order to turn the data into valuable assets, it is very important to enable the human user to comprehend, interpret and validate the huge amount of discovered patterns. To achieve this, pattern post-analysis methods are developed for automatically analyzing, organizing and managing the discovered patterns so that the human user can obtain useful information from them.

From the academic point of view, the discovery of useful information inherent from data is one of the ultimate goals in data mining and knowledge discovery. Either by being-told or by self-discovery, pattern post-analysis is a process the target of which is to obtain the behavioral phenomena or principles of the working domain from the huge amount of discovered patterns. The objective is to enable human to better comprehend the behavior in the same working domain.

From the application point of view, pattern post-analysis has huge potential applications in various real-world problems. In fact, it is the pressing demand from the business and industry that motivates this research. In particular, the thesis research motivation comes from the following practical problems. Consider a company which has a large database. An oil and gas production company, for instance, may have logged hundreds of thousands of records about the oil and gas production process. Or an education institution may have a database of the students' academic records. Suppose that these companies or institutions wish to understand the business better. The concerns they have in mind could be as follows. Could they use their existing databases to automatically derive the patterns for diagnosis or training purpose? Could they uncover the regularities that reflect the operations and performance of the companies? Could they find out the relationships among market demands and decision criteria? More specifically, for the oil and gas production company, what advices should the system give when a new engineer with little experience of the existing production process needs to monitor the process? If there are existing models describing components A and B in an oil and gas production process but no model describing their interaction, can the system discover the relationship between these two components?

With the advent of inexpensive electronic and magnetic storage media and the ever broadening use of computers in a vast spectrum of businesses, utilizing large databases is becoming a common practice. Nowadays, the average size of databases range from gigabytes to terabytes; however, analyzing these databases and providing the users with useful knowledge is very difficult in the meantime. The huge volume of data makes manual analysis virtually impossible. Since the number of patterns can be much larger than the number of data samples, the problem of pattern post-analysis can be even more challenging. The real-world characteristics of these databases such as noise, incompleteness, inconsistency and redundancy are open questions posed to today's machine learning research. These demands and concerns create both a need and an opportunity to automatically extract knowledge from databases. It is quite clear that if a company has an existing database of its business records, such a pattern post-analysis system would be very useful. On the one hand, it uncovers the useful information trapped in the huge set of patterns. Such information may provide insights for even experienced decision makers. On the other hand, the uncovered information can compensate for the lack of experiences and/or knowledge of junior staff. Hence, useful information will not be lost when senior staff leaves the company or retires.

Academically, this research targets some open problems in data mining and automatic knowledge acquisition from large databases. These challenging problems motivate the research and form the objectives of the research. Like data analysis, there are several common tasks for pattern post-analysis. For example, pattern pruning removes uninteresting and/or redundant patterns [8], [9], [20], [23] – [39]; pattern clustering groups similar patterns into clusters [23], [40] – [44]; pattern summarization builds a representative summary of the patterns [20], [21]; and visualization presents the patterns to the users in an easy-to-understand manner [40].

Fundamental to all of the above post-analysis tasks is how to measure the distance between discovered patterns effectively. Like the role that distance between data samples plays in data analysis, clustering and classification, the distance between patterns are crucial. It can be used for a wide variety of tasks such as: pruning redundancies in pattern pruning; grouping similar patterns for pattern clustering; and identifying representative patterns for summarization and visualization.

This thesis first focuses on the crucial problem of measuring distances between patterns (including itemsets and association rules). Based on the developed pattern distances, algorithms for pattern pruning, pattern clustering and pattern summarization developed in the dissertation work are presented. In the literatures, it was found that the matched samples of the discovered patterns are a very good source to provide additional information about the patterns [8], [9], [20], [21], [23], [40]. To measure distance between association rules, Toivonen et al [23] proposed the well-known distance that counts the number of samples where patterns differ. Similarly, Gupta et al [40] developed a normalized distance that counts the number of samples where patterns share. Their distance was applied to the agglomerative chain clustering for pattern clustering and to the self-organizing map for pattern visualization.

The concept of samples matching has also been used in pattern pruning. In [24], the distance proposed by Gupta et al [40] was used for pattern pruning. Another representative method is RuleCover pruning [23]. Given a set Γ of association rules having the same consequent, its subset Δ is called a rule cover if the rules in Δ match the same set of samples matched by the original rules in Γ. In other words, a rule cover is a subset of the original set of association rules such that the cover matches all the samples that the original set matches. Furthermore, closed itemset pruning [8], [9], [29] – [31], which is widely used for pruning itemsets and for improving the speed of mining association rules, is also based on sample matching. An itemset is called a closed itemset if it is the superset of all pruned itemsets having the same number of matched samples (i.e. support).

While sample-matching distances have been widely used in various pattern post-analysis tasks, they do not capture certain important characteristics of the discovered patterns. When applied to various pattern post-analysis tasks, misleading results may follow. The introduction of the new distance measures is an important step to advance effective pattern post-analysis of categorical data. As in the analysis of categorical data, the impediment of effective analysis is due to the lack of effective distance measures which should take into consideration of the probabilistic variation of the data. Similar impediment is also encountered in pattern post-analysis in order to further explore the problem and examine the difficulties. In this thesis, to overcome the weakness of sample-matching distance, I formally analyze the properties of sample-matching distances and propose new distance measures based on a dual pattern-data relationship. The dual relationship provides an explicit one-to-one correspondence between patterns and their associated. Such explicit correspondence enables effective analysis of distances among patterns, resulting in two new distance measures, namely sample-attribute-matching distances and entropy-based distance.

The practical problems of how the new distance measures and their related concepts can be used in pattern pruning, pattern clustering and pattern summarization are also addressed. The experimental works are extensively carried out. A comprehensive comparative study of the experimental results on carefully planned synthetic data and real-world data then follow. The objective of the study is to have a full grasp of the problem, identify the hurdles to overcome and layout the path for further development. The ultimate goal is to develop an integrated system that can automatically analyze the discovered patterns and present the results understandable by human.

The following list gives a more detailed description to each of the research problems as stated in section 1.1.

1. The quantitative description of the complex interrelating relationship among patterns.

   Since the relationship between patterns is realized in their associated data set, the crucial relationship between pattern and data has to be established explicitly before the distance between patterns can be meaningfully and accurately measured. Once such explicit pattern-data relationship is established, formal analysis of distance measure among discovered patterns can be derived. Up-to-date, most of the existing pattern post-analysis systems developed use the sample-matching distances. Hence, they could not account for the number of attributes that two patterns differ nor the difference between two patterns attributed by noise or the variation of their associated data. In this thesis, an analysis of sample-matching distances in categorical data is first presented. From the analysis, new concepts of sample-attribute matching and data variation within clusters are proposed for defining more accurate measurement of distances between patterns. These two concepts are used to develop effective and generalized algorithms for pattern pruning, pattern clustering and pattern summarization.

2. The automatic clustering of discovered patterns and their associated data.

   A common task in pattern post-analysis is pattern clustering. It is an important task for organizing and analyzing the discovered patterns. In data clustering, the key function of clustering is to bring similar data together. In pattern clustering, the process that brings patterns together has to rely on the detected closeness of their associated data clusters. Thus, a good pattern clustering method needs to simultaneously cluster patterns as well as their associated data into coherent groups. The pattern-data relationship should be made explicit for further analysis. Hence, in this thesis, a dual clustering process is proposed which is able to cluster both patterns and their associated data while maintaining an explicit one-to-one pattern-data relationship. Such explicit pattern-data relationship enables further analysis of individual clusters including pattern summarization.

   Using the concepts of sample-attribute matching and data variation within clusters, the dual clustering process employs an entropy-based distance measures between patterns. The new distance measures are able to take into account the noise and data variation inherent in the data clusters.

3. The automatic pruning of redundant patterns,

   Automatic pruning of redundant patterns is highly desirable because it can significantly reduce the complexity of the subsequent analyses. Due to the interrelating nature of patterns, the notion of redundancy has to be defined by considering how patterns overlap and entangle with each other in the data through the explicit pattern-data relationship. Once the notion of redundancy is defined, redundant patterns can be pruned. In this thesis, the concept of sample-attribute matching is used to define redundancy between patterns.

4. The automatic summarization of patterns

   It is very useful in real world applications for a pattern post-analysis system to automatically generate a summary of the patterns. A concise and accessible informative summary could help a human user significantly in gaining a better grasp of the patterns discovered from the data. When searching for representative patterns for summarization, the interrelating relationship among patterns is taken into consideration. Here, the concept of sample-attribute matching is used to define how representative a pattern is by considering the overlapping nature of patterns. Then a set of representative patterns is selected to summarize the rest of the patterns.

5. The easefulness of the understandability of the organized patterns.

   The ultimate goal of pattern post-analysis is to support easy and quick knowledge acquisition from the huge volume of patterns for the users. First, pattern post-analysis system should produce results interpretable by human. Second, the post-analysis process should be made explicit and the results should be made transparent. Such demands are related to one of the natural requirement of learning *transparency* versus *blackbox* [45], [46]. Unlike regression or prediction, a black-box approach is not suitable for knowledge discovery and decision making support. With a transparent system, it is much easier to construct a meaningful explanation of the patterns and their relationships. Throughout the pattern post-analysis system, the analyses are made explicit and the results are transparent and understandable. The easefulness of the understandability is one of the most important aspects for a knowledge discovery and data mining system.

1.2.2 Thesis Objectives

The followings are the objectives of this study. This study will:

- develop an integrated system which is able to automatically analyze and organize the discovered patterns from a given data set in a way that human can easily comprehend, interpret and validate;
- develop an effective and robust dual clustering algorithm supported by effective distance measures between patterns. The algorithm is able to simultaneously cluster both patterns and their associated data and maintains an explicit one-to-one relationship between patterns and data for subsequent analysis;
- develop the ability to prune redundant patterns with controllable tradeoff of information loss and the number of patterns retained;
- develop the ability to generate a concise and informative summary describing a huge set of interrelating and entangling patterns; and
- accomplish experimental demonstration and evaluation for analyzing the performance of the proposed method.

1.3 Research Outline

The research presented in this thesis can be subdivided into four sections. The first section focuses on the explicit dual relationship between patterns and their associated data, and the pattern distance measures derived from such pattern-data relationship. The rest of the three sections describe the use of pattern-data relationship and its related concepts including sample-attribute matching and data variation consideration in the problems of pattern clustering, pattern pruning and pattern summarization.

In the first section, the dual relationship between patterns and their associated data is introduced through the discussion of sample-attribute matching and the impact of noise and variation inherent in the data. Using the pattern-data relationship, these concepts are naturally incorporated in the measurement of distances between patterns. Thus, two types of pattern distances are developed. The first is sample-attribute-matching distances which extend from the existing sample-matching distances [20], [23], [24], [26] – [28]. The second type is entropy-based distances which take into the consideration of noise and variation inherent in the data. In the second section, using the proposed distance measures, a dual clustering algorithm is developed for clustering patterns as well as their associated data. In the third section, the concept of sample-attribute matching is used to develop a new type of itemsets in association rule mining which generalizes the classical closed [8], [9], [29] – [31] and maximal itemsets [32] – [35] for pattern pruning. In the fourth section, sample-attribute matching is again employed to develop a new pattern summarization method. It generalizes the RuleCover algorithm [23] into AreaCover algorithm to select patterns that would have better and more effective coverage in terms of both samples and attributes of the data sets.

1.3.1 Dual Relationship between Patterns and Data and Its Related Concepts

The first portion of the research introduces the dual relationship between patterns and their associated data. The relationship provides an explicit one-to-one correspondence between patterns and data. This provides an important base to measure the distance between patterns through the differences obtained from their associated data.

With the pattern-data relationship, a formal analysis of distance measures between patterns is provided. Under the assumption that there is no order among samples and attributes in a relational table (see section 1.1), the data containing a pattern can always be considered as a continuous rectangular block by swapping the samples (rows) and attributes (columns) of the table. This assumption gives a simple and unified view of various distance measures in the literatures. Under this unified view, exiting distances measures, including item-matching distances and sample-matching distances [20], [23], [24], [26] – [28], are analyzed.

From the analysis, we propose new pattern distance measures that take into consideration of the effects of attribute matching and data variation in clusters. The new distance measures are analyzed under the same unified view so that the relationship between the new measures and the existing ones are made clear. In particular, we show that the new measures are extended from the existing sample-matching distances [20], [23], [24], [26] – [28] by considering additional important factors that have been overlooked.

1.3.2 Simultaneous Pattern and Data Clustering

The proposed dual pattern-data relationship is used to develop a dual clustering algorithm, known as simultaneous pattern and data clustering. The algorithm simultaneously clusters patterns as well as data while keeping an explicit dual pattern-data relationship. Two common clustering algorithms, namely hierarchical clustering and *k*-means clustering, are implemented. Like data clustering, hierarchical pattern clustering produces the entire clustering hierarchy and always produces the same result given the same distance measure. Hence, it is ideal for studying and comparing different distances. However, hierarchical pattern clustering is not scalable. In contrast, *k*-means pattern clustering is fast and scalable since it only produces a partition rather than the entire hierarchy. However, it requires users to set the number of clusters. It also involves a random cluster initialization. Hence, different executions of *k*-means will produce different clustering results, making evaluation and comparison difficult. Despite the limitations of the two methods, they are commonly used in clustering. The dual clustering algorithm is implemented using both methods.

1.3.3 Pattern Pruning

It is common to prune redundant patterns before conducting other pattern post-analyses such as pattern clustering and pattern summarization. Today, most of the pruning algorithms are based on sample matching. Using the concept of sample-attribute matching, a new type of itemsets is proposed. It generalizes closed itemsets [8], [9], [29] – [31] and maximal itemsets [32] – [35] into generalized itemsets while considering them as two special extreme cases. It provides a way for the users to control the amount of information loss in pattern pruning, thus enabling them to balance the tradeoff between information loss in pruning and number of patterns pruned. The generalized itemsets provides a more general alterative to closed itemsets and maximal itemsets.

1.3.4 Pattern Summarization

One of the ultimate goals of pattern post-analysis is to support the discovery of useful knowledge from data. As stated in the introductory sections, a more desirable way of using knowledge is to be able to reveal it at a glance. Pattern summarization aims at automatically selecting a small subset of patterns that are representative to other patterns. Again, the concept of attribute matching is integrated with sample-matching to develop a method which summarizes each pattern cluster produced by pattern clustering. It is an extension of the RuleCover pruning algorithm [23] into what we call the AreaCover algorithm which considers both samples and attributes in the selection of the summary patterns for each cluster.

1.4 Organization of the Thesis

There are six chapters in this thesis including this introduction.

To give a better understanding of the research field, a brief review of existing ideas relevant to pattern mining and post-analysis is presented in chapter 2. Discussions of individual approaches follow a general overview of pattern mining and post-analysis. The advantages and disadvantages of these methods are also examined with regard to the goals of this research. The preliminary concepts that will be used throughout the rest of this thesis are also presented. It describes the concepts and definitions of association rule mining and pattern discovery, two commonly used pattern mining methods.

Chapters 3 and 4 embody the major part of this research. In chapter 3, the dual relationship between patterns and data is introduced. Then, a formal analysis of various distance measures between patterns is provided. It explains how item-matching distances and sample-matching distances measure the distances between patterns. The limitations of these existing distances are discussed leading to the necessity of sample-attribute matching and data variation consideration. Based on these concepts, new sample-attribute-matching distances and entropy-based distances based on the dual pattern-data relationship are proposed. Discussion regarding the properties and differences of various distance measures is followed with supportive demonstration.

To demonstrate the efficacy and usefulness of the new distance measures, they are used to build an integrated pattern post-analysis system which consists of pattern pruning, simultaneous pattern and data clustering and pattern summarization. The ultimate goal of the system is to help users to easily and quickly understand and interpret the patterns discovered and obtain useful knowledge from the huge volume of data and patterns.

In chapter 3, the proposed distance measures are first applied to the problem of pattern clustering. A dual process, known as simultaneous pattern and data clustering, is developed for clustering similar patterns and their associated data simultaneously into clusters. Two common clustering algorithms are implemented using the proposed distance measures, namely, hierarchical clustering and *k*-means clustering.

In chapter 4, the concept of sample-attribute matching is used to develop a new type of itemsets, known as generalized itemsets. It has been proven that generalized itemsets are a generalization of closed itemsets and maximal itemsets, two commonly used itemset pruning techniques. Generalized itemsets are a pattern pruning technique to remove redundant patterns. It can be applied before pattern clustering. Moreover, in this chapter, pattern summarization method, known as AreaCover, is developed to summarize each pattern cluster produced by pattern clustering. The method adopts the concept of sample-attribute matching to summarize patterns. It is an extension of the well-known RuleCover algorithm [23]

In chapter 5, the proposed distances based on the dual pattern-data relationship are extensively tested in the context of pattern pruning, clustering and summarization. Both synthetic and real-world data sets were used in the experiments. The experiments are divided into four groups. In the first group, the pruning performance of generalized itemsets is tested and evaluated with ten benchmark data sets. The results are then compared with the well-known closed itemset and maximal itemset methods. How well the system can handle large data sets is investigated. In the second group, seven synthetic data sets were generated to study the differences between the proposed distance measures and the existing ones. In the third group, the new distance measures are implemented in hierarchical and *k*-means clustering algorithms. Their performances are tested and evaluated using the same ten benchmark data sets. The results are compared with the common sample-matching distances. In the fourth group, the summarization performance of the new AreaCover algorithm is tested with ten benchmark data sets. The results are compared with RuleCover algorithm. In all experiments, the speeds of various methods are reported.

Chapter 6 highlights the contribution of this study and suggests the direction of future research in this area.

Chapter 2
Review of Related Works

2.1 An Overview of Pattern Mining and Post-Analysis

This section gives an overview of the development history of pattern mining and its related fields. The development of pattern mining can be traced back to the classical data analysis in statistics. Data analysis has long been recognized as a significant research challenge by statisticians and more recently by researchers in artificial intelligence (AI). Pattern mining comes as an extension of data analysis. Later, it became a part of the activities in the broader disciplines of machine learning. While statisticians focuses on building models from data to characterize system behavior, AI researchers attempt to understand the system better by describing the discovered regularities in a way that humans can easily interpret. Numerous research papers and reports are now available. It is difficult to give a comprehensive comparison of those methods since different methods have different objectives based on different assumptions on the problems. However, they can be categorized along several broad directions from a more general viewpoint.

Pattern mining aims at automatically discovering unknown regularities from data. In the ordinary sense, “discovering regularities” from a system, or a data set, simply involves grouping the data samples into classes according to the similarity of the samples [47]. Hence, discovering patterns is very similar to statistical clustering. However, these methods do not render conceptual descriptions of the clusters nor consider how humans would describe a pattern. In contrast, the AI approaches try to represent the discovered patterns in a form that can be naturally interpreted by humans. Two such commonly used representations of patterns are rules [47] and trees [48]. These representations can be used to support analytical tasks [49] such as classifying a new sample or predicting the missing value of an attribute.

With the demand from applications of expert systems for automatic knowledge acquisition, AI researchers try to teach the machines to discover useful knowledge automatically from data. Unfortunately, traditional manual data analysis techniques are rather *ad hoc* and cannot easily meet the challenges of huge amount of data and the fast growing demand of knowledge. To address these issues, machine learning aims at automatically finding the relations among the attributes and/or among their values. In the literatures, such approaches are referred to as conceptual clustering [47], object classification [48], or rule induction [50]. The importance of learning in AI has been repeatedly and alternatively emphasized by a number of researchers [51] – [54]. There are several forms of learning, ranging from supervised learning to unsupervised learning [55]. In supervised learning, there is a teacher (often represented by class labels) supervising the learning. The learner is told explicitly what is to be learned. In unsupervised learning, there is no teacher (i.e. no class label is given). The learner discovers whatever they think is important (e.g. defined by objective functions). Traditionally, machine learning research pays more attention to supervised learning or classification problem. It only focuses on patterns that are related to the class labels assigned by an external teacher (i.e. the users). The desired performance of supervised learning is apparent, i.e. to improve the prediction accuracy of class membership. In contrast, the performance of unsupervised learning is more difficult to measure. However, unsupervised learning can be applied to a wider range of applications where an external teacher (or class labels) is not given. Ideally, a good learning system should be able to learn patterns with and without an external teacher or explicit classification information. In other words, when classification information is not available, a learning system should be able to perform unsupervised learning. At the same times, it should be able to perform classification tasks when asked.

The techniques of machine learning can be subdivided into two distinct categories, namely the symbolic approaches and the statistical approaches. Better known examples of symbolic techniques include Mitchell's version space algorithm [56] and its later evolution, and the AQ family of algorithms of Michalski [57] including the concept clustering algorithm CLUSTER/2 [47]. Symbolic approaches assume that the learning environment is deterministic [58] – [61] and do not handle noises very well. Hence, their application areas are rather restrictive since most real-world problems involved noises by nature. To address this problem, statistical approaches were developed to handle noisy, incomplete and imperfect data commonly encountered in the real world. Representative works include Breiman's CART [62], Quinlan's ID3 [48] and its variations such as C4.5 [63] and *CDP*[64], Fisher's COBWEB [65], Symth and Goodman's ITRULE [50] and the Bayesian approaches [66]. In these methods, various statistical measures or hypothesis tests are applied to detect pattern and/or rules.

In the past few decades, the explosive growth of technologies in data generation and collections provides a huge amount of information. The overwhelming amounts of data not only make traditional manual methods data analysis virtually infeasible, but also post a tremendous challenge to machine learning. Hence, knowledge discovery from database (KDD) [67] or data mining [1], [2], [68] become a challenging topic for researchers in machine learning, statistics and data analysis [69]. While KDD can be considered as a process from data selection to pattern interpretation/evaluation [69], pattern mining and post-analysis are two major components of the process. A number of statistical and machine learning methods have been adopted and integrated into a data mining system. Compared with traditional machine learning methods which focus more on classification problems, pattern mining and post-analysis in KDD are more general and sensitive to computational complexity due to large amount of data.

2.2 Pattern Mining Methods

In this section, a subset of common techniques in pattern mining are reviewed and discussed. The review is brief but provides insights into the current state-of-the-art of pattern mining. The discussion will be focused on KDD and data mining.

2.2.1 Tree-Based Approaches

Trees are commonly used in decision support tasks such as classification and concept generation. The idea behind is to represent a complex decision into a union of several simpler decisions represented in a tree or a forest. Classical examples are ID3 [48] and CART [62]. ID3 constructs a decision tree using a divide-and-conquer approach [48]. It is simple and effective. In a decision tree, each node partitions the data samples based on the values of a single attribute. An information theoretical measure is employed to choose the attribute whose values improve the classification accuracy of the class membership. The original version of ID3 [59] was designed to include all the positive training samples and to exclude all the negative ones, leading to a potential problem of over-fitting. To avoid over-fitting, algorithms such as ASSISTANT [70], C4 [71] and C4.5 [63] use strategies such as pre-pruning [70], [72] and post-pruning [53], [73] to remove the branches of the decision tree that are too detailed and specific. However, since the decision trees only use univariate splits, they can only be applied to a small portion of the functional models [69]. Complicated patterns such as the XOR problem are difficult to discover if only one attribute is split at each node.

To allow the tree to split at a node when multiple variables are considered, CART [62] was developed. It is able to detect more complex patterns. However, CART is computationally expensive since it needs to generate multiple auxiliary subtrees. Other tree-based classification methods include those reported in [74] – [76]. A comprehensive survey of decision tree can be found in [77]. All decision tree methods are designed for supervised learning or classification problem.

For unsupervised learning, trees are also widely used. Well-known examples are Michalski and Stepp's CLUSTER/2 for conceptual clustering [47]. CLUSTER/2 generates a tree which partitions a set of data samples into *K* groups. A criterion called LEF is employed to guide the clustering process. Each node of the tree is a cluster at the leaf level and is described by logical complexes (a logical product of one or more attribute-value pairs). CLUSTER/2 is not scalable for large data sets and does not handle noise. To deal with noise, COBWEB was proposed by Fisher [65]. COBWEB incorporates a new sample into the class that best matches the samples. A criterion known as category utility is used to direct the clustering process. The generated result is a tree where each node represents a concept and the tree describes the relations between concepts. COBWEB is able to learn patterns from data with noise [78]; however, it does not work well in deterministic environment [79].

2.2.2 Rule-Based Approaches

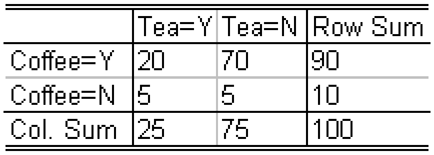
Rules are another commonly used representation in decision support tasks, especially for expert systems. Many researchers such as Smyth and Goodman [50] believe that rules offer a more flexible representation than trees. It is also easier to understand rules than trees especially when the trees are large and complex. Rules can be used for both supervised and unsupervised learning.

In supervised learning, rules classify samples into classes at the consequence side. Typical examples are AQ with its extensions [51], [80] and CN2 [81], [82]. AQ represents classification rules by disjunctive complexes which are easy to understand by human. AQ works well in deterministic environment; however, it is slow and may not perform well in noisy environment. Another problem is that it requires users to have a good understanding of the problems for manipulating its parameters [81]. Such knowledge may not be available. To address some of the problems of AQ, CN2 [81], [82] and GREEDY3 [83] were proposed. CN2 also produces rules in the form of disjunctive complexes. It can handle noisy data because it uses a probabilistic measure to direct the process. However, both CN2 and GREEDY3 are very slow and not scalable for large data sets.

In unsupervised learning, Agrawel and Srikant developed association rule mining which discovers association rules from transaction databases in 90's [4] – [6]. The method uses a user-defined support, which is basically the probability of an itemset, to determine if an itemset is frequent or not. Another user-defined confidence (conditional probability of an association rule) is used to determine if an association rule is strong or not. Association rule mining does not consider negative associations or missing items. To reduce the search space, an important property called the apriori property was used. Based on this property, very efficient algorithms have been developed for very large databases [5], [6]. Association rule mining has been extensively studied and widely used in various real world applications. It is a powerful tool to explore and analyze large data sets.

Association rule mining is well-suited to applications such as market basket analysis. However, Brin et al [15] pointed out that, for some applications where item correlation is required, association rules may be misleading. For example, in Table 1, the association rule [Tea=Y]⇒[Coffee=Y] has 20% support and 80% confidence [15]. With fairly high support and confidence, we may consider it as a valid rule and believe that customers who buy tea will also buy coffee. However, [Tea=Y] and [Coffee=Y] are actually negatively correlated since the ratio $P\{[Tea=Y] \wedge [Coffee=Y]\}/ (P\{[Tea=Y]\} \times P\{[Coffee=Y]\})= 0.2/(0.25 \times 0.9) = 0.89 < 1$.

Table 1. The contingency table of the purchase of tea and coffee [15]



	Tea=Y	Tea=N	Row Sum
Coffee=Y	20	70	90
Coffee=N	5	5	10
Col. Sum	25	75	100

To address this issue, Brin et al [15] proposed the use of chi-squared statistics to detect correlation rules from the contingency tables. However, since the chi-squared statistics obtained from the entire contingency table was designed for testing correlations among random variables rather than among events, correlation rule is less accurate if the contingency table data are sparse.

Pattern discovery moves the hypothesis test from taking the entire contingency table to focusing on its individual cells [10] – [14]. In Table 1, to determine whether [Tea=Y, Coffee=Y] is a significant pattern, it tests the difference between the observed frequency $o = 20$ and the expected frequency under independence assumption $e = 100 \times P\{[Tea=Y]\} \times P\{[Coffee=Y]\} = 100 \times 0.25 \times 0.9 = 22.5$. If the difference $20 - 22.5 = -2.5$ is significant enough, we would conclude that [Tea=Y] and [Coffee=Y] are negatively associated. Since the difference, and hence the hypothesis test, is obtained from an individual cell in the table, pattern discovery can handle sparse contingency table data. The relation between pattern discovery, association rule mining and chi-squared statistics are given in appendixes A and B in more details.

2.2.3 Other Approaches

There are many other pattern mining methods that produce neither trees nor rules. For example, graphical approaches represent probabilistic dependencies using graphs. A graph (pattern) encodes which variables that are dependent on each other. Most of these approaches are based on the Bayesian inference and their models are represented by networks such as Markov networks or Bayesian networks. Bayesian inference has a strong theoretical basis. It also provides a formal framework for reasoning with uncertainty and partial beliefs. Once a probabilistic network is built, the probability of an event conditioned by a set of observations can be derived for classification purpose. Since there is a large set of parameters to be estimated in the network, special methods are required to automatically construct the networks from data. For example, Fung and Crawford developed CONSTRUCTOR [84] to generate a discrete Markov networks from data automatically. Thus, the networks contain both a quantitative (i.e. probabilistic) characterization and a qualitative (i.e. structural) description of the data. The idea behind CONSTRUCTOR is simple. It finds the Markov boundary of each node (i.e. attribute) in the networks so that the effect of other nodes outside the boundary is minimized. The independence between a node and the nodes outside the boundary is tested using high dimensional contingency tables. A heuristics, known as composable distribution, is used to avoid checking for high order dependency. The problem of this approach is that Markov networks cannot represent all kinds of dependencies among variables [85]. More specifically, they cannot represent induced and non-transitive dependencies [85]. Furthermore, the use of contingency tables introduce very heavy computational burden for high order dependency. Most importantly, CONSTRUCTOR is a variable-based method. It is worth pointing out that methods dealing with event-based dependencies such as the rule-based and the tree-based methods are more efficient than variable based methods [61], [86], [87]. From the inference point of view, Markov networks have to attach a matrix of joint probabilities to each edge of the network, otherwise, the original data will be used to estimate the joint probabilities for inference. If the domains of the variables are large, the matrix of joint probabilities will be large. This makes the networks difficult to handle. Since not all the joint events of two variables are significant, it is not necessary to store the information regarding all these events because more events not only require more storage spaces but also involve higher computational complexity.

Since Markov networks cannot represent the induced and non-transitive dependencies [85], Bayesian networks use a richer language of directed graphs to improve the representative power. In Bayesian networks, the directions of the edges permit us to distinguish genuine dependencies from spurious dependencies induced by hypothetical observations [66], [85], [88], [89]. Bayesian networks are also variable-oriented approaches and therefore suffer from the same problems as the other variable-oriented methods.

Many other systems which have been developed cannot be covered in this brief review. But most, if not all, of them can be found similar to one of the categories discussed above. For a good discussion on pattern mining methods, interested readers can refer to [69] for more details.

2.3 Pattern Post-Analysis Methods

Among all pattern mining methods, rule-based approach is one of the most common approaches. It has several advantages over other approaches. First, the rules produced are easy to understand for non-experts such as business managers. Hence, they have been widely used in business and commercial applications. Second, the rule-based approaches assume very little knowledge about the data from the users. Thus, when the users do not have any a priori knowledge about a data set, rule-based approaches are good starting points for them to explore the data. Third, the representation power of rules is strong. Fourth, effective and efficient rule-based mining algorithms, such as association rule mining [4] – [9], correlation rule mining [15], [16] and pattern discovery [10] – [14], are available, which can be applied to large, noisy and incomplete databases.

Common to all rule-based mining methods is the problem of having too many rules or patterns which are often produced by them. Hence, pattern post-analysis, also known as interesting measures [90] – [92], is needed to support the discovery of useful knowledge from huge volume of patterns. In this section, a subset of commonly used techniques in pattern post-analysis are reviewed and discussed. Again, the review is brief but would provide insights into the current state-of-the-art of pattern post-analysis.

2.3.1 Objective Approaches

The problem of handling the overwhelming number of patterns has been widely studied by researchers in the AI, machine learning and data mining areas. Due to the popularity of association rule mining, most methods are designed for itemsets and association rules after its inception. In this thesis, although the proposed methods can be applied to various types of patterns, they will mainly be applied to itemsets [4] – [9] and their variants including closed itemsets [8], [9], [29] – [31] and maximal itemsets [32] – [35].

Pattern post-analysis is generally categorized into two major approaches: objective approaches based on the statistical strengths or properties of the discovered patterns inherent in the data and subjective approaches that are directed by user's beliefs or expectations in their particular problem domain [90], [92]. Objective approaches do not require domain knowledge from the users about the problems, whereas subjective approaches do. Common tasks in objective approaches include pattern pruning [8], [9], [20], [23] – [39], pattern clustering [23], [40] – [42] and pattern summarization [20], [21]. Pattern pruning removes redundant and/or irrelevant patterns from the original set of discovered patterns. Pattern clustering groups similar and/or relevant patterns into clusters. Pattern summarization generates a comprehensive and representative summary for all discovered patterns. Other related tasks include pattern visualization [40].

In pattern post-analysis, pattern pruning is one of the most popular methods. In their well-known paper [23], Toivonen et al observed that the matched samples are a very good source for patterns relationship. They proposed the use of rule cover to prune redundant association rules sharing the same consequent. A greedy algorithm, known as RuleCover, was developed to find the close-to-optimal rule cover. In addition, they also proposed the distance measure d_T in support of the pattern clustering task (see section 3.3 for detailed discussion). Their methods have been widely used for pattern pruning and clustering and were extended by other researchers [24], [25], [40]. Since the RuleCover algorithm is a greedy algorithm, it only guarantees local optimal rule cover. In addition, the stepwise selection of a subsequent rule is dependent on which rules have been previously chosen. Hence, the final rule cover produced is dependent on the ordering of the rules. In [25], an integer programming technique was proposed which always produces the same set of rules (i.e. rule cover) independent of any ordering of the rules and always results in the most optimal rule set. However, the algorithm is slower than RuleCover. In [40], Gupta et al normalized d_T and proposed d_G for pattern clustering and visualization. In [24], d_G is used to prune association rules by representing rule direction as hyperedges. Other works along this line include [26], [27] and [28]. In [26], a distance metric between rules was used to select the most heterogeneous set of rules that together gives a good coverage of the instance space. The method, however, can only be applied to data with uniform distribution and is sensitive to outliers. There is no concrete guidance to specify the values of the three weight parameters in the distance function. In [27], the notion of representative association rules (RR) was introduced. RR is a least set of rules that covers all association rules. Subsequently, a user may be provided with the set of RR's instead of the whole set of association rules. However, when needed, all usual association rules can be generated from the set of RR's by means of a cover operator. In [28], a pruning strategy called redundancy exploitation was proposed. The idea is to prevent continued effort at classifying instances already classified by existing rules with high confidence.

Before Toivonen et al‘s works, the concept of samples matching has been used in the classical closed itemsets for pruning itemsets and in improving the speed of mining association rules [8], [9], [29] – [31]. The advantage of the closed itemsets is that it is lossless from which the original itemsets can be recovered. An extreme case of closed itemsets is the maximal itemsets (also known as long patterns in some literatures such as [32], [33]), which have been used to significantly reduce the number of itemsets regardless of its possible loss of information [32] – [35].

Other methods of pattern pruning include the use of chi-squared statistics to measure the significance of association rules and the insignificant ones are pruned [20]. Bayardo et al [36] proposed to use minimum improvement (min_imp) in confidence to prune association rules. The idea is to mine only those rules whose confidence is at least min_imp greater than the confidence of any of its simplifications, where a simplification of a rule is formed by removing one or more conditions from its antecedent. In [20], [37], [38], a rule is pruned if its confidence is close to that of one of its subrules. In [39], the maximum entropy principle was used to prune redundant association rules.

In the literatures, most pruning methods are designed for association rules. In practice, they can be combined with itemset pruning methods. For example, closed and maximal itemsets can be applied to prune redundant itemsets first. Then, association rule pruning methods can be applied to prune the generated association rules. There are also other types of patterns such as correlation rules [15], [16] and event association patterns [10] – [14]. In [93] – [95], a divide-and-conquer approach was used for analyzing event association patterns discovered by pattern discovery [10] – [14]. In the divide phase, association patterns and their associated data are simultaneously clustered, whereas in the conquer phase, individual clusters are further analyzed.

Another common pattern post-analysis method is pattern clustering. In [23], similar patterns are grouped into clusters using a nonparametric density method. The sample-matching distance measure d_T was first proposed (discussed in details in section 3.3). Later, in [40], a normalized version of d_T, denoted as d_G, was proposed. A dimensionless agglomerative chain clustering was developed to cluster patterns using d_G. Dimensionless agglomerative chain clustering is a special case of agglomerative chain clustering (see chapter 3 in [41]). In this algorithm, a pattern is grouped to its closest neighbor found from the distance matrix. This process is applied to all the patterns resulting in a collection of pattern clusters. Agglomerative chain clustering performs chaining at multiple levels. At the end of the algorithm, a tree structure describing the multiple levels of clustering is produced. It is similar to single link agglomerative clustering [1], [2], but differs in its bias. The tree produced is shorter and the clusters are more uniformly sized. In addition, at each level, more than two patterns can be merged. To visualize patterns using self-organizing map (SOM) [1], [2], the scalar distance d_G between rules must first be converted into an embedded vector space since SOM needs a vector input. Hence, multi-dimensional scaling (MDL) was used to convert the distance information into an embedded space such that the distance information between rules is preserved. The embedded space obtained can then be used in SOM for pattern clustering and summarization.

In [42], the problem of clustering two dimensional association rules was considered. A geometric-based algorithm, known as BitOp, was proposed which uses heuristic methods based on the geometric properties of the two-dimensional grids to cluster association rules in two-dimensional space. The algorithm was designed for segmenting data. The quality of the segmentation was measured by the minimum description length principle of encoding the clusters on several databases. The algorithm is limited to only those rules with two fixed attributes in their antecedents. Another approach reported in [43] lifts the two-dimensional restriction. However, clustering is only based on numeric attributes. In [44], an algorithm, known as Objective Grouping (OG), was proposed. OG groups rules according to the syntactic structure of the rules without using any domain knowledge.

Other pattern post-analysis tasks include pattern summarization. In [20], [21], a method was developed to find a special subset of all the association rules to form a summary of them. This subset of association rules is called the directional setting (DS) rules since they set the directions followed by the rest of the rules. The direction of a rule is the type of correlation it has (i.e. positive correlation, negative correlation or independence) which can be computed using chi-squared test. In experiments, it was shown that the number of DS rules is typically very small. They can be manually analyzed by a human user.

Often, a single method is not adequate to solve the problem of having too many patterns. For example, the number of patterns after pruning may still be too large for human to handle. Other methods such as pattern clustering, summarization and visualization should be applied after pruning. Hence, a hybrid approach combining pattern pruning, clustering, summarizing and visualization is commonly used. In [23], pruning and clustering rules are used together so that only the pruned association rules are grouped. Similarly, in [20], [21], pruning and summarizing rules are used together. In [40], clustering and visualization are used together to visualize the pattern clusters. Table 2 summarizes the hybrid methods found in the literatures.

Table 2 Common hybrid methods for pattern post-analysis

	Approaches	Measures/Methods
Brin et al [15], [16]	1. Searching supported, and 2. correlated itemsets	1. Support 2. Chi-squared statistics
Liu et. al [20], [21]	1. Pruning, and 2. Summarization	1. Chi-squared statistics 2. Chi-squared statistics and rules' direction
Toivonen et al [23]	1. Pruning, and 2. Clustering	1. Rule cover based on the set of matched samples 2. d_r
Gupta et al [40]	1. Clustering, and 2. Visualization	1. d_G 2. Self-organizing map
Wong & Li [93] – [95]	1. Clusterin, and 2. Analysis of individual clusters	1. d_R, d_{RC}, d_O *or* d_D 2. Standard discrete-valued data analysis techniques (eg. Subgrouping tree)

In Table 2, chi-squared test for correlation has been widely used in various methods. The mathematical relationship between chi-squared test and residual analysis used in pattern discovery [10] – [14] is given in appendix B.

2.3.2 Subjective Approaches

All methods described in the previous section do not involve the domain knowledge of the users. They analyze patterns mainly based on the properties of the data sets (e.g. sample matching, supports, chi-squared statistics, minimum improvement in confidence, etc). Such methods are known as objective methods. Alternatively, there are subjective methods which require domain knowledge of the problems. These methods incorporate the domain knowledge of the users, which, if used properly, can significantly improve the effectiveness of the post-analysis and the usefulness of the results produced. For instance, subjective pattern pruning using templates or constraints [96], [106], [107] can usually prune more patterns than objective pattern pruning. If the templates or constraints are properly specified by the users, the results could be more relevant to the users than those produced by objective pattern pruning. However, the difficulty of subjective pruning lies in requiring the users to specify a good constraints, templates or criteria. In many cases, the users may not have a good knowledge about the data. Moreover, it is difficult to verify the knowledge provided by the users. Most existing methods simply assume that the users' knowledge provided is correct. However, in practice, it is often not true since the users' knowledge is simply based on experiences and impression, which is *ad hoc* by nature. To my best knowledge, there is currently no good method to verify the domain knowledge provided by users.

Subjective pattern post-analysis has been intensively studied. In the literatures, most of the research addresses the problem of subjective measures of interestingness of the discovered patterns [90] – [92]]. Piatetsky-Shapiro discussed the general issue of the interestingness of the discovered patterns in [98]. A general study of measures of rule interestingness can be found in [91], [99] – [101]. An overview of the interestingness of an association rule with respect to a set of constraints can be found in [38]. In [102], the authors proposed the method of random worlds and prove that in many important cases it is equivalent to the principle of maximum entropy.

Some researchers suggest that additional specification from the users could be used to select the interesting patterns. In a paper [103], Silberschatz and Tuzhilin argue that interesting associations are those unexpected from the users. They proposed a method that asks the users to specify their existing knowledge and then search those unexpected associations for them. In [38], a rule is considered interesting with respect to some set of beliefs if it contradicts at least one of the rules in the beliefs under the monotonicity assumption. A detailed statistical analysis of interestingness of a rule with respect to a single subrule , and algorithms for finding rules interesting in this setting can be found in [104], [105].

One of the most common approaches in defining interestingness measures is to use templates/constraints to specify interesting or uninteresting patterns. Srikant et al. [96] used item constraints specified by the users to obtain interesting associations. Basically, the item constraints specify which items should appear in the association rules. In [106], Hoschka and Klosgen used templates for defining interesting knowledge. They proposed to use a few fixed statement types and partial ordering of attributes to specify the templates. In [107], Klemettinen et al proposed to use regular expression to specify the templates so that the users can specify what association rules they like. These approaches require users to specify clearly what they know or need.

In [108], Piatetsky-Shapiro and Matheus proposed to group deviations from normative expectation by means of utility functions with the KEFIR system. The utility functions were quite easy to define, as their system was intended to save money in a health-care application. The interestingness measures were based on the actionability of a particular pattern by measuring the savings anticipated from taking a specific action. The system then recommended to the user the most cost-effective approach to take. In [109], Anand et al extends Piatetsky-Shapiro and Matheus‘s deviation approach by providing a methodology for the support of cross-sales in a commercial domain. In both systems, domain knowledge played an important role to determine the effectiveness of the deviation measures.

In [110], Jaroszewicz and Simovici presented a method for pruning itemsets based on background knowledge represented by a Bayesian network. The interestingness of an itemset is defined as the absolute difference between its supports estimated from data and from the Bayesian network.

In [44], a pattern clustering algorithm, known as Subjective Grouping (SG), was developed which incorporates domain knowledge and groups the rules according to the semantic information of the objects in the rules.

Just as in objective approaches, most subjective approaches are sample-matching based. However, there are a few attribute-oriented approaches in subjective approaches. In [111], an attribute-oriented approach is used to prune uninteresting relations. In [112], a similarity measure is defined based on an attribute hierarchy (a tree structure) provided by human expert. By specifying a rule aggregation level, the rules are generalized using the non-leaf nodes at the aggregation level and the rules with the same aggregated rule are grouped together. Hence each group can be described by the aggregated rule. However, this approach requires intensive user interaction during the grouping process, where the user must specify the aggregation level. When the attribute hierarchy is huge, the user may not have a clear idea about what could be the appropriate aggregation level.

Subjective methods such as interestingness measures are interesting topics for pattern post-analysis. Nevertheless, the focus of this thesis is on objective methods.

2.4 Introduction to Association Rule Mining and Pattern Discovery

This section introduces two common types of pattern mining methods, namely, association rule mining (section 2.4.1) and pattern discovery (section 2.4.2) and their related definitions and concepts that will be used throughout the rest of the thesis.

2.4.1 Introduction to Association Rule Mining

This section introduces the concepts of frequent itemsets and association rules [4], [6] – [8] that will be used throughout the thesis. Consider a data set D that contains M data samples. A sample is denoted by x. Each sample is described by N discrete-valued attributes. Let $X=\{X_1, \dots , X_N\}$ represent this attribute set. An item of an attribute X_i is a value of X_i and is denoted by x_i.

To represent a subset of attributes, let s be a subset of integers $\{1 , \dots , N \}$ containing k elements ($k \le N$). Then, X^s is a subset of X. That is, $X^s = \{X_i \mid i \in s\}$ where s is called the attribute index set of X^s. An itemset is a set of items from a subset of attributes X^s and is denoted by x^s. Let o_{x^s} be the observed frequency of occurrences of x^s. The *support* of the itemset x^s is [4] – [8]

$$support(x^s) = \frac{o_{\mathbf{x}^s}}{M} \tag{1}$$

where M is the total number of samples. An itemset x^s is called frequent itemset if its support is greater than a pre-defined minimum support (abbreviated as min_sup).

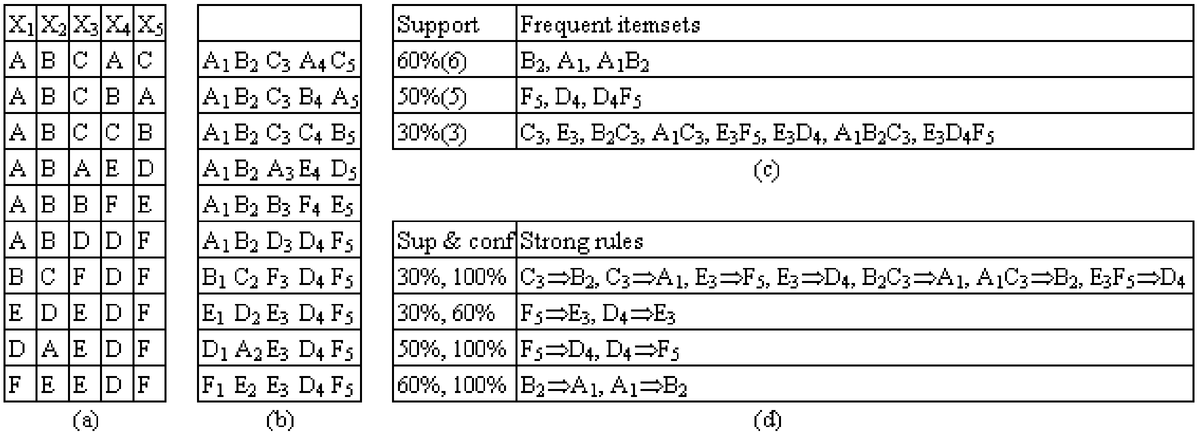
An association rule is an implication of the form $A \Rightarrow B$ which denotes that the observation of A (known as antecedent) infers that B (known as consequent) is probably true. Let a and b be two subsets of integers $\{1, \dots, N\}$ where $a \cap b = \varnothing$. Then X^a and X^b are two disjoint subsets of X. That is,

$$X^a = \{X_j \mid j \in a \} \text{ and } X^b = \{X_j \mid j \in b \} \text{ and } X^a \cap X^b = \varnothing$$

Let x^a and x^b be the values of X^a and X^b. The confidence of an association rule $x^a \Rightarrow x^b$ is

$$confidence(x^a \Rightarrow x^b) = \frac{support(x^a \cup x^b)}{support(x^a)} \tag{2}$$

The support of an association rule $x^a \Rightarrow x^b$ is defined as *support*($x^a \cup x^b$). A rule is called a strong association rule if its support is greater than the minimum support and its confidence is greater than a pre-defined minimum confidence (min_conf). By convention, the support and confidence values occur between 0% and 100%. For example, consider a relational database in Figure 2(a). There are totally 10 samples, each of which is described by 5 attributes $X_1 – X_5$. In Figure 2(b), the relational database is considered as a transactional database by mapping each attribute-value pair in Figure 2(a) to a distinct item. Figure 2(c) presents all frequent itemsets when the minimum support is 30% (i.e. 3 samples). For example, the support of D_4F_5 is 50% because 5 out of the 10 samples contain it. Figure 2(d) gives all strong association rules when the minimum support is 30% and minimum confidence is 60%. For example, the confidence of $F_5 \Rightarrow E_3$ is 60% because 3 out of the 5 samples containing F_5 also contain E_3.



X_1	X_2	X_3	X_4	X_5
A	B	C	A	C
A	B	C	B	A
A	B	C	C	B
A	B	A	E	D
A	B	B	F	E
A	B	D	D	F
B	C	F	D	F
E	D	E	D	F
D	A	E	D	F
F	E	E	D	F

(a)

A_1 B_2 C_3 A_4 C_5
A_1 B_2 C_3 B_4 A_5
A_1 B_2 C_3 C_4 B_5
A_1 B_2 A_3 E_4 D_5
A_1 B_2 B_3 F_4 E_5
A_1 B_2 D_3 D_4 F_5
B_1 C_2 F_3 D_4 F_5
E_1 D_2 E_3 D_4 F_5
D_1 A_2 E_3 D_4 F_5
F_1 E_2 E_3 D_4 F_5

(b)

Support	Frequent itemsets
60%(6)	B_2, A_1, A_1B_2
50%(5)	F_5, D_4, D_4F_5
30%(3)	C_3, E_3, B_2C_3, A_1C_3, E_3F_5, E_3D_4, $A_1B_2C_3$, $E_3D_4F_5$

(c)

Sup & conf	Strong rules
30%, 100%	$C_3 \Rightarrow B_2$, $C_3 \Rightarrow A_1$, $E_3 \Rightarrow F_5$, $E_3 \Rightarrow D_4$, $B_2C_3 \Rightarrow A_1$, $A_1C_3 \Rightarrow B_2$, $E_3F_5 \Rightarrow D_4$
30%, 60%	$F_5 \Rightarrow E_3$, $D_4 \Rightarrow E_3$
50%, 100%	$F_5 \Rightarrow D_4$, $D_4 \Rightarrow F_5$
60%, 100%	$B_2 \Rightarrow A_1$, $A_1 \Rightarrow B_2$

(d)

Figure 2. (a) A relational database (b) The corresponding transactional database (c) All frequent itemsets with min_sup = 30% (d) All strong rules with min_sup=30% and min_conf=60%

2.4.2 Introduction to Pattern Discovery

Pattern discovery uncovers itemsets that do not follow a pre-assumed model (or the null hypothesis). Any default model can be chosen according to the problem domain and the available knowledge. If *a priori* knowledge about the domain is not available, similar to chi-squared (χ^2) statistic, a model assuming the independence of the random variables is normally used. Under this assumption, the expected frequency e_{x^s} of an itemset x^s can be calculated as:

$$e_{x^s} = M \prod_{x_i \in x^s} P(x_i) \tag{3}$$

where M is the sample size and

$$P(x_i) = \frac{o_{x_i}}{M} \tag{4}$$

where o_{x_i} is the observed frequency of the item x_i.

For an itemset x^s, the difference $o_{x^s} - e_{x^s}$ measures how x^s deviates from the independence assumption. However, according to [113], [114], the absolute difference $\left|o_{x^s} - e_{x^s}\right|$ cannot be employed for evaluating the relative size of the discrepancy between o_{x^s} and e_{x^s} because the absolute difference may be affected by the marginal totals. Hence, the residual is first standardized before any analysis is conducted. The standardized residual z_{x^s} is defined by

$$z_{x^s} = \frac{o_{x^s} - e_{x^s}}{\sqrt{e_{x^s}}} \tag{5}$$

Standardized residual has an asymptotic normal distribution with a mean of approximately 0 and a variance of approximately 1. Hence, if z_{x^s} exceeds 1.96, by conventional criteria, we conclude that the items of x^s are "associated" and likely to occur together at 95% confidence level. x^s is referred to as a positive association pattern, or simply a positive pattern. If it is less than -1.96, x^s is referred to as a negative pattern. Standardized residual is considered as normally distributed only when the asymptotic variance of z is close to 1, otherwise, it has to be adjusted by its variance for a more precise analysis. The adjusted residual is expressed as:

$$d_{x^s} = \frac{z_{x^s}}{\sqrt{v_{x^s}}} \tag{6}$$

where v_{x^s} is the maximum likelihood estimate of the variance of z_{x^s}, obtained by:

$$v_{x^s} = Var\left(\frac{o_{x^s} - e_{x^s}}{\sqrt{e_{x^s}}}\right) = 1 - \prod_{x_i \in x^s} P(x_i) \tag{7}$$

The background of residual analysis can be found in [115]. Haberman [116] discussed the properties of residuals for contingency table analysis. The mathematical relation between chi-squared test and residual test is given in appendix B. The details of how v_{x^s} is obtained in (7) can be found in [114]. In appendix A, the mathematical relation between association rule mining and pattern discovery is derived. Finally, in appendix B, the mathematical relation between chi-squared test and residual test is discussed. Here, to conclude this chapter, a simple example is used to illustrate the basic idea of pattern discovery.

Consider an XOR data set containing 1000 data samples. Suppose that each item (e.g. [A=T]) of the data set occurs 500 times. The number of expected occurrences e of the itemset [A=T, B=T, C=F] is

0.5×0.5×0.5×1000 = 125. Suppose that its actual occurrence *o* is 250. The adjusted residual is 15.81, larger than 1.96 which is the value at the 95% significant level. Thus we conclude that this itemset is a positive third-order pattern. Back to the example in Table 1, the adjusted residual of [Tea=Y, Coffee=Y] is -1.92, larger than -1.96 but smaller than -1.65. Hence, we conclude that it is a negative pattern at 90% significant level. While pattern discovery can discover both positive and negative patterns, in this thesis, only the positive patterns are considered. Post-analysis of negative patterns is in the scope of future research.

2.5 Summary

Pattern mining and post-analysis is a board research area. The above discussions are mainly taken from the viewpoint of artificial intelligence and machine learning. Major rationale, focuses and endeavors devoted to the development of new systems from the preceding review are summarized as follow:

- The ability to automatically analyze and organize the huge volume of discovered patterns produced by pattern mining methods is crucial for real-world applications. It is only when the discovered patterns can be interpreted and converted into useful knowledge or actionable plan that the discovered patterns would be meaningful and useful.
- The measurement of the interrelating relationship among patterns, usually formulated as a distance/similarity measures between patterns, is crucial for pattern post-analysis. Once developed, an effective distance measure and/or its related concept can be applied to various post-analysis tasks such as pattern pruning, pattern clustering, pattern summarization and visualization.
- Existing objective pattern post-analysis approaches are mainly based on matching samples, though a few attributed-oriented methods in subjective approaches have been reported.
- Because of the complexity of the real-world problems, it is difficult and time-consuming for the users to specify their knowledge to the system. The objective approaches in pattern post-analysis are therefore highly desirable.
- Most existing methods limit their applications to only a particular problem or a particular type of patterns, and are therefore not general enough to render an integrated post-analysis framework for real-world applications.
- A single pattern post-analysis method is usually inadequate to solve the problem of “too many patterns”. An integrated, hybrid and flexible system combining different methods is needed for real-world problems.

Chapter 3

Simultaneous Pattern and Data Clustering

3.1 Introduction

To solve the too many patterns problem, an integrated pattern post-analysis as shown in Figure 3 is developed in this thesis. It consists of three major components – pattern pruning, pattern clustering and pattern summarization. In this chapter, the pattern clustering component, known as simultaneous pattern and data clustering [93] – [95], or simply pattern clustering, is introduced. Its schematic representation is shown in Figure 4. Patten clustering is introduced first because its fundamental concepts will be used to build the algorithms in pattern pruning and pattern summarization as well, which will be presented in chapter 4.

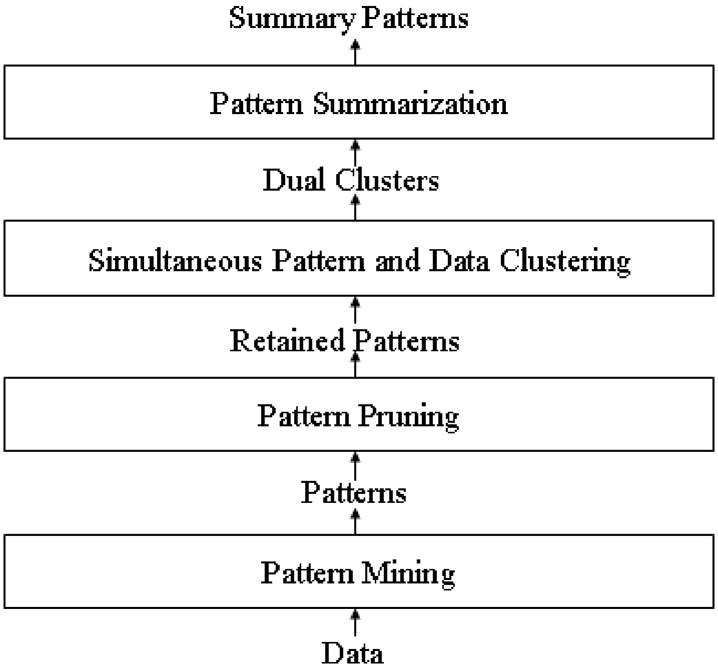


Figure 3. An overview of proposed pattern post-analysis system

Simultaneous pattern and data clustering is a dual clustering process which is able to simultaneously cluster the discovered patterns and their associated data into clusters for pattern management, analysis and interpretation [93] – [95]. In Figure 4, once patterns (e.g. frequent itemsets [4] – [9], association rules [4], [6] – [8] or event association patterns [10] – [14]) are discovered, the simultaneous pattern and data clustering method will cluster the patterns based on a distance measure derived from their associated data. When two patterns are clustered into a pattern cluster, the data associated with them will be simultaneously merged into a data cluster. Thus, both the patterns and their associated data are clustered and the relationship between them is made explicit through their one-to-one correspondence. At the end of the clustering process, the algorithm produces pattern clusters as well as their corresponding data clusters. The two types of clusters are collectively referred to as dual clusters. An important advantage of dual cluster is that the relationship between pattern clusters and their associated data clusters is made explicit. The explicit one-to-one correspondence makes it possible to analyze each dual cluster individually. This enables the summarization of each pattern cluster individually later by the method introduced in chapter 4.2.

Since the proposed method clusters patterns and their associated data by a distance measure derived from data, the notion of distance measures between patterns is first addressed in sections 3.2 – 3.5. Sections 3.2 and 3.3 review the existing item-matching distances and the widely used sample-matching distances respectively. Sections 3.4 and 3.5 then introduce the concepts of sample-attribute matching and data variation and derive the relationship between them and the existing distances. Later, in chapter 4, the concept of sample-attribute matching is used to develop efficient and generalized methods for pattern pruning and pattern summarization. In section 3.6, two commonly used clustering algorithms, namely hierarchical clustering (section 3.6.1) and *k*-means clustering (section 3.6.2), are used to implement the dual clustering process.

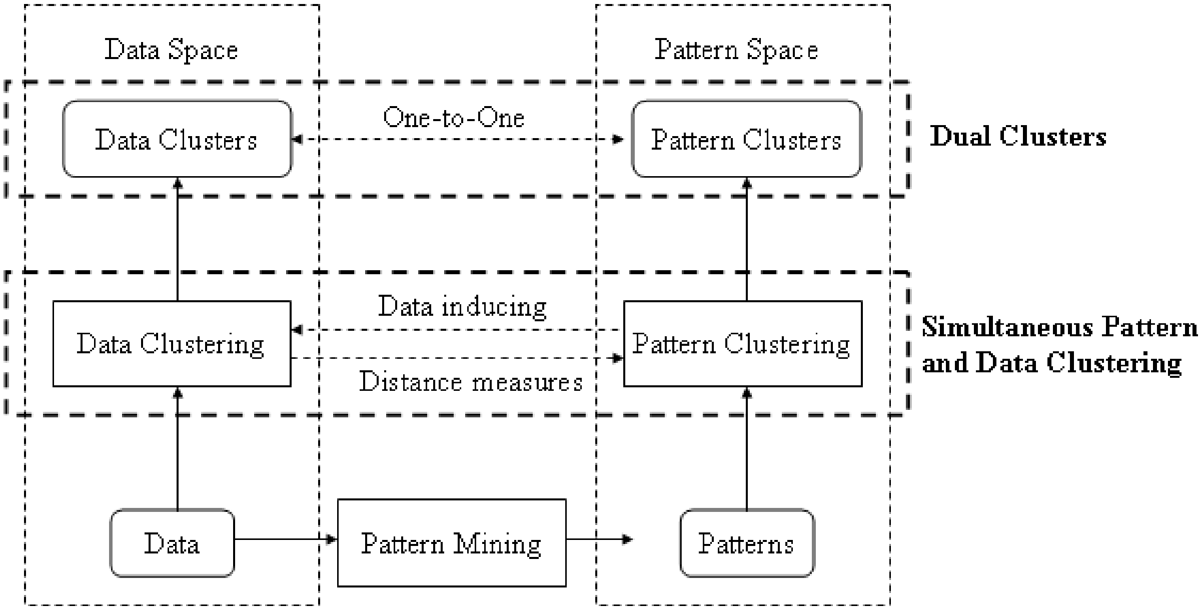


Figure 4. The dual clustering process of simultaneous pattern and data clustering

3.2 Item-Matching Distances

When defining distance between patterns, most distances in the literatures do not consider the direction of rules since direction is usually either considered as irrelevant to rule distance or can be dealt with separately. For instance, the rule [computer] => [science] is considered as an itemset [computer, science]. If we follow this view, the discussion on distance measures will be significantly simplified. This view also allows us to compare distance measures originally designed for different types of patterns (e.g. itemsets, association rules [4], [6] – [8], correlation rules [15], [16] and event association patterns [10] – [14]). Hence, from now on, we focus on non-directional patterns.

A naïve approach to measure distance between patterns is by counting the number of common items shared by them. For example, in a text mining application, the patterns [computer, science] and [computer, language] share the item [computer] and so their distance is 1. However, this approach has two drawbacks. First, related patterns may not contain common items. For instance, [computer, science] and [programming, language] do not share any common items but programming language is an important subject in computer science. Second, unrelated patterns may contain common items. For instance, [computer, science] and [social, science] share a common item but computer science and social science are two separate fields. Hence, counting the number of common items may miss certain relationship between patterns and may produce misleading results.

3.3 Sample-Matching Distances

Due to the problems of item-matching distances, most pattern distances nowadays are based on sample matching. The idea is simple: it counts the number of samples where the patterns share or differ. A well-known example of sample-matching distances was proposed by Toivonen et. al [23]. Suppose that a set of patterns $\{x_1^{s_1}, x_2^{s_2}, ..., x_n^{s_n}\}$ is discovered. Then, the set of samples matched by a pattern $x_i^{s_i}$ is denoted by *m*(*i*) $=\{x \in D \mid x \supseteq x_i^{s_i}\}$. In Figure 5, *m*(*i*) and *m*(*j*) are matched by patterns $x_i^{s_i}$ and $x_j^{s_j}$ respectively. Note that $\frac{|m(i)|}{M}$ = *support*($x_i^{s_i}$). Let r_i (r_j) be the number of samples matched by $x_i^{s_i}$ ($x_j^{s_j}$) but not matched by $x_j^{s_j}$ ($x_i^{s_i}$). That is, $r_i = | m(i) \setminus m(j)|$ and $r_j = | m(j) \setminus m(i)|$. Let r_{ij} be the number of samples matched by both $x_i^{s_i}$ and $x_j^{s_j}$. That is, $r_{ij} = | m(i) \cap m(j)|$. The distance proposed by Toivonen et. al. is defined as [23]:

$$d_T(i,j)=r_i + r_j \tag{8}$$

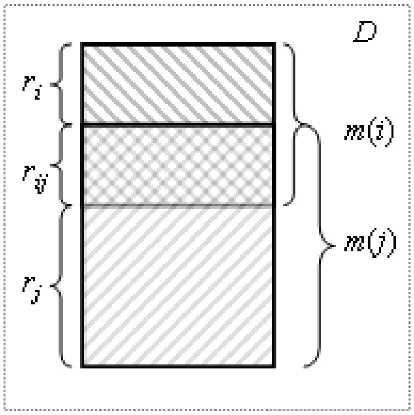


Figure 5. Sample-matching distances

In [40], Gupta et. al. pointed out that d_T tends to give higher values for rules that are matched by more samples (i.e. high $|m(i)|$). For example, two pairs of rules, both consisting of non-overlapping rules, may have different distances. Pairs with higher $|m(i)|$ will have a greater distance. To address this problem, they proposed a normalized distance [40]:

$$d_G(i, j) = 1 - \frac{r_{ij}}{r_i + r_j + r_{ij}} \tag{9}$$

Note that $0 \leq d_G \leq 1$. $d_G = 0$ if the two rules have an identical set of matched samples and $d_G = 1$ if they have non-overlapping sets of matched samples. Obviously, one can have other variants of distances/similarity based on the set of matched samples. For instance, we find that the ratio:

$$d_R(i, j) = \frac{r_i + r_j}{r_{ij}} \tag{10}$$

works fairly well since it captures both the similarity (r_{ij}) and dissimilarity (r_i + r_j).

3.4 Sample-Attribute-Matching Distances

Sample-matching distances do not give special consideration to the attributes where two patterns share or differ. As an illustration, consider two pairs of patterns $x_i^{s_i}$, $x_j^{s_j}$ and $x_p^{s_p}$, $x_q^{s_q}$ in Figure 6 (a) and (b) respectively. Let c_i, c_j and c_{ij} bear the same meaning for the set of matched attributes as r_i, r_j and r_{ij} for the set of matched samples. Now, suppose that $r_i = r_p$, $r_j = r_q$ and $r_{ij} = r_{pq}$, and $c_{ij} > 0$, $c_{pq}= 0$. The measures d_T, d_G and d_R will yield the same value for both pairs of patterns since $r_i = r_p$, $r_j = r_q$ and $r_{ij} = r_{pq}$. However, it seems more reasonable to consider that $x_i^{s_i}$ and $x_j^{s_j}$ are more similar since they share certain attributes ($c_{ij} > 0$) while $x_p^{s_p}$ and $x_q^{s_q}$ do not ($c_{pq} = 0$).

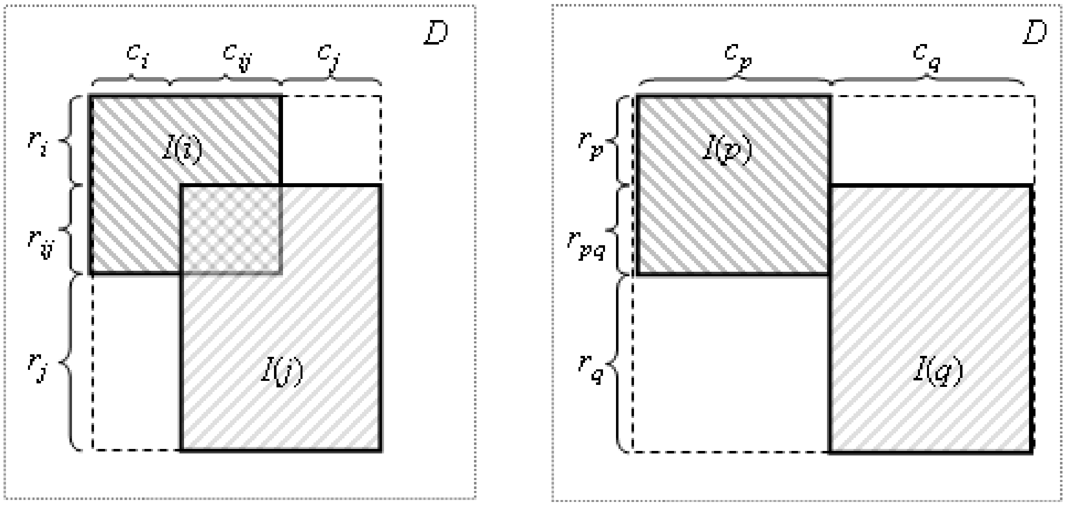
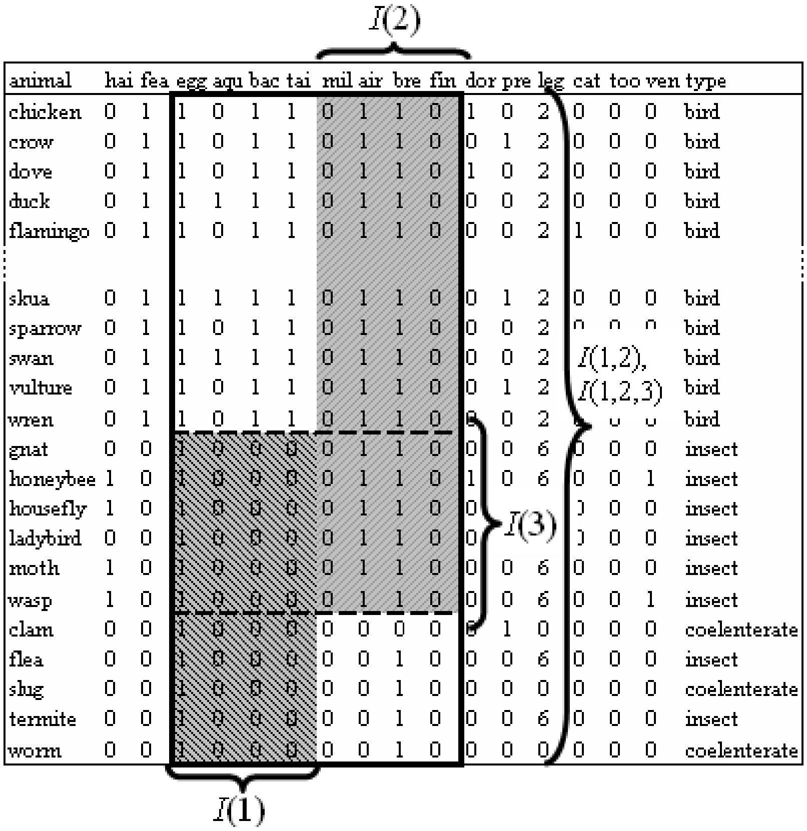


Figure 6. Sample-attribute matching distances

This motivates us to introduce the concept of pattern-induced data cluster, which consider both the sets of matched samples and matched attributes. A pattern-induced data cluster of a pattern $x_i^{s_i}$, or simply a data cluster, is a set of items containing $x_i^{s_i}$. It is defined as:

$$I(i)=\{x^s \subseteq x \mid x \in m(i), s=s_i\} \tag{11}$$

As an example, in Figure 7, *I*(1) is a data cluster (the dark shaded block) induced by a pattern [eggs=1, aquatic=0, backbone=0, tail=0]. *I*(2) (the light shaded block) is induced by [milk=0, airborne=1, breathes=1, fins=0] and *I*(3) (the block bound by dashed lines) is induced by [eggs=1, aquatic=0, backbone=0, tail=0, milk=0, airborne=1, breathes=1, fins=0].


I(2)
I(1,2),
I(1,2,3)
I(3)
I(1)

animal	hai	fea	egg	aqu	bac	tai	mil	air	bre	fin	dor	pre	leg	cat	too	ven	type
chicken	0	1	1	0	1	1	0	1	1	0	1	0	2	0	0	0	bird
crow	0	1	1	0	1	1	0	1	1	0	0	1	2	0	0	0	bird
dove	0	1	1	0	1	1	0	1	1	0	1	0	2	0	0	0	bird
duck	0	1	1	1	1	1	0	1	1	0	0	0	2	0	0	0	bird
flamingo	0	1	1	0	1	1	0	1	1	0	0	0	2	1	0	0	bird
skua	0	1	1	1	1	1	0	1	1	0	0	1	2	0	0	0	bird
sparrow	0	1	1	0	1	1	0	1	1	0	0	0	2	0	0	0	bird
swan	0	1	1	1	1	1	0	1	1	0	0	0	2	[illegible]	[illegible]	[illegible]	bird
vulture	0	1	1	0	1	1	0	1	1	0	0	1	2	[illegible]	[illegible]	[illegible]	bird
wren	0	1	1	0	1	1	0	1	1	0	0	0	2	[illegible]	0	0	bird
gnat	0	0	1	0	0	0	0	1	1	0	0	0	6	0	0	0	insect
honeybee	1	0	1	0	0	0	0	1	1	0	1	0	6	0	0	1	insect
housefly	1	0	1	0	0	0	0	1	1	0	0	[illegible]	[illegible]	0	0	0	insect
ladybird	0	0	1	0	0	0	0	1	1	0	0	[illegible]	[illegible]	0	0	0	insect
moth	1	0	1	0	0	0	0	1	1	0	0	0	6	0	0	0	insect
wasp	1	0	1	0	0	0	0	1	1	0	0	0	6	0	0	1	insect
clam	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	coelenterate
flea	0	0	1	0	0	0	0	0	1	0	0	0	6	0	0	0	insect
slug	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	coelenterate
termite	0	0	1	0	0	0	0	0	1	0	0	0	6	0	0	0	insect
worm	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	coelenterate

Figure 7. Pattern-induced data clusters

To measure the distance between two patterns, we need to merge individual data clusters. Let *I*(*i*) and *I*(*j*) be two data clusters induced by patterns $x_i^{s_i}$ and $x_j^{s_j}$ respectively. The merged data cluster of *I*(*i*) and *I*(*j*) is the union of their matched samples and matched attributes, expressed as:

$$I(i,j)=\{x^s \subseteq x \mid x \in m(i) \cup m(j), s=s_i \cup s_j\} \qquad (12)$$

The definition can be generalized to *n* patterns, i.e., $I(1,\ldots,n)=\{x^s \subseteq x \mid x \in m(1) \cup\ldots\cup m(n), s=s_1\cup\ldots\cup s_n\}$. For instance, in Figure 7, *I*(1, 2) (the thick lined block) is merged from *I*(1) and *I*(2). Note that *I*(1,2,3), being merged from the 3 data clusters, is the same as *I*(1, 2). As another example, in Figure 6, the 4 highlighted rectangles are actually induced data clusters *I*(*i*),*I*(*j*), *I*(*p*) and *I*(*q*). When two data clusters, say *I*(*i*) and *I*(*j*), are merged, the items in the top-right and bottom-left corners are added into the merged data cluster.

The pattern-induced data cluster establishes an explicit one-to-one correspondence between patterns and their associated data. In this thesis, such one-to-one correspondence is referred to as the dual relationship between patterns and data. The dual relationship will be used to develop a dual clustering algorithm later in this chapter as well as other efficient pattern pruning and summarization methods in chapter 4.

One possible measure that takes into account both the matched samples and attributes is:

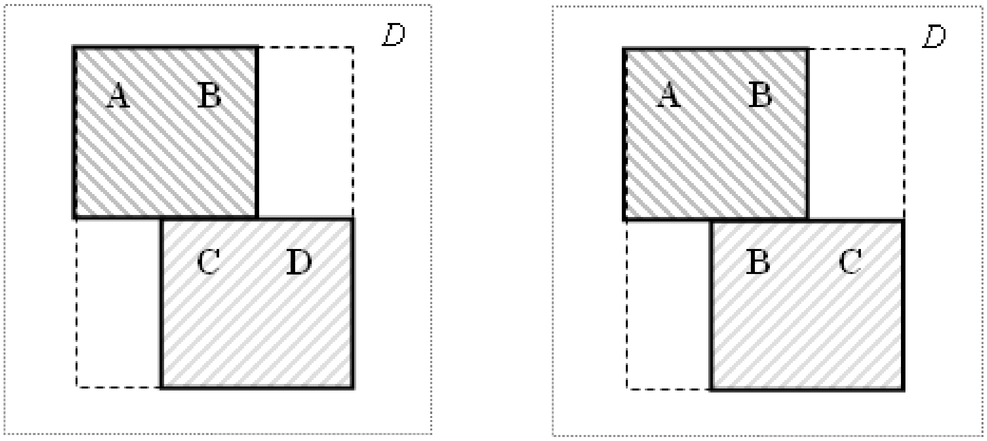
$$d_{RC}(i, j) = w_r \frac{r_i + r_j}{r_{ij}} + w_c \frac{c_i + c_j}{c_{ij}} \tag{13}$$

where w_r and w_c are the weight of rows (samples) and columns (attributes) respectively. If we consider matched samples and matched attributes as equally important, we may set w_r and w_c to 0.5.

3.5 Entropy-Based Distances

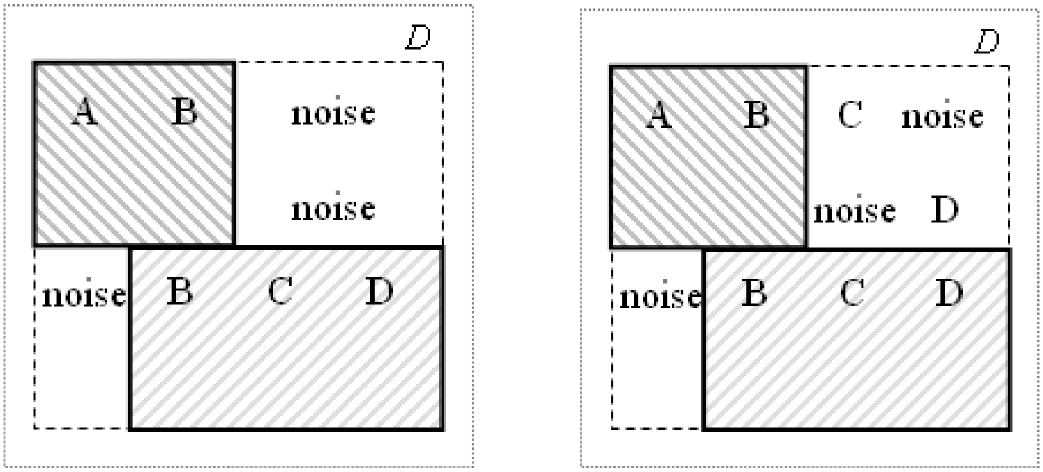
One problem of the sample-attribute-matching distance measure d_{RC} is that it does not consider the variation within the data clusters. The measures, including d_T, d_G, d_R and d_{RC}, discussed so far are designed to measure the distance among patterns. However, they do not measure the distance among the data associated with the patterns. It is worth stating at this point that clustering patterns without simultaneously clustering their associated data may lose insights about how patterns and data are interrelated. On the one hand, while most data analysis techniques are inapplicable to analyze pattern clusters, they can be applied to their associated data clusters once we know how patterns are realized in data. This duality allows us to analyze pattern clusters via their associated data clusters using standard data analysis techniques. In chapter 4, we also introduce an efficient pattern summarization method to summarize individual pattern clusters based on the data cluster associated with a pattern cluster (see [93] for other possible standard analysis techniques that have been used to analyze dual clusters.). On the other hand, knowing which set of patterns induce which data clusters allows users to use understandable patterns to interpret and validate the data clusters. In view of this, it is desirable to simultaneously cluster patterns and data and keep their relationship explicitly for further post-analysis.

Since sample-matching and sample-attribute-matching distances (e.g. d_T, d_G, d_R and d_{RC}) do not measure the variation inside the data clusters, they actually overlook certain very important factors when measuring distances between patterns and their data. For example, in Figures 8 (a) and (b), if the areas of the corner regions are the same for the two pairs of patterns, sample-matching and sample-attribute-matching distances will consider the two pairs as having equal distance. However, it is obvious that the second pair is closer than the first one because the second pair shares the same item B. As another example, consider two identical pattern pairs in Figure 9. The corners of the first pair contain mainly noises whereas the top-right corner of the second pair contains mainly items C and D with some noises. Even though the two pairs are the same, the second pair should have a closer distance than the first one because the items in the top-right corner are more consistent with the items among the patterns. In later experiments (section 5.2), artificial data are generated to further study the limitations of sample-matching and sample-attribute matching distances.



(a) (b)

Figure 8. The first pattern pair



(a) (b)

Figure 9. The second pattern pair

The above observation motivates us to take into account the variation within data clusters in pattern distance measure. A common measure of variation/uncertainty for discrete-valued data is entropy. In general, the entropy of a data cluster $I(1, \ldots, n)$ can be expressed as:

$$H(I) = -\sum_{x^s \in I} P(x^s)\log P(x^s) \tag{14}$$

where $P(x^s)$ is the joint probability distribution of the itemset x^s and I is the abbreviation of $I(1,\ldots,n)$. The computation complexity of $P(x^s)$ is exponential (i.e. $O(2^{|s|})$). To reduce the complexity, we adopt a naïve assumption that the attributes are conditionally independent given a data cluster. The joint entropy is then estimated by summing up the entropy of individual attribute:

$$H(I) = -\sum_{i \in s} \sum_{x_i \in x^s, x^s \in I} P(x_i) \log P(x_i) \tag{15}$$

where *s* is the attribute index set of *I*. $P(x_i)$ is estimated by:

$$P(x_i) = \frac{o_{x_i}}{|I|} \tag{16}$$

where o_{x_i} is the observed frequency of x_i in *I* and | *I* | is the number of itemsets in *I*. The computation complexity of *H*(*I*) in (15) is O(| *I* | | *s* |). Since | *s* | is usually much smaller than | *I* |, the complexity could be taken as O(| *I* |), which is linear. From (15), all constant clusters have zero entropy. For example, in Figure 7, we have *H*(*I*(1)) = 0 and *H*(*I*(2)) = 0. Note that *H*(*I*(1,2,3)) = *H*(*I*(1, 2)) = 3.66 since merging *I*(1, 2) and *I*(3) results in the same data cluster as *I*(1, 2) (i.e. same amount of uncertainty). When all values in each attribute are equiprobable, the entropy is maximal.

Note that $0 \le H(I) \le \sum_{i \in s} \log m_i$, where m_i is the number of possible values of the *i*th attribute. Hence, *H*(*I*) in (15) can be normalized as:

$$\underline{H}(I) = \frac{H(I)}{\sum_{i \in s} \log m_i} \tag{17}$$

$\underline{H}(I)$ does not explicitly consider the numbers of matched samples and matched attributes. Hence, it should be weighted by the area of *I*. The distance measure then becomes:

$$d_o(I) = |I| \, |s| \, \underline{H}(I) \tag{18}$$

An appropriate weighting is important for comparing the normalized entropies in regions with different sizes. Intuitively, a larger region should have a greater impact than a smaller region and thus should be assigned with greater weight. For example, it may be acceptable to have a small region with high variation, but not as acceptable to have a large region with small variation.

Table 3 summarizes the properties of all the distance measures.

Table 3. Comparison of properties of patterns distance measures d_T, d_G, d_R, d_{RC} and d_O.

	Samples Matching	Attributes Matching	Data Variation	Computation Complexity
d_T [23]	Yes	No	No	O(| *I* |)
d_G [40]	Yes	No	No	O(| *I* |)
d_R	Yes	No	No	O(| *I* |)
d_{RC}	Yes	Yes	No	O(| *I* | + | s |)
d_O	Yes	Yes	Yes	O(| *I* | | s |)

A final remark is that this section considers only those measures that ignore the rule direction and hence a rule can be considered as a pattern. Although this is true for most pattern distances, it is still possible to include rule direction. In [24], rule direction is encoded by directed hyperedges. d_G is then used to measure distances between two rules. Hence, to measure rule distance, rule direction can be augmented with the measures discussed in this section. However, in this paper, we focus on non-directional pattern distances. Hence, rule direction is not used.

3.6 The Clustering Algorithms

A very important task in pattern post-analysis is pattern clustering. To solve the too many patterns problem after pattern mining, a natural task to follow is to organize the discovered patterns as well as their associated data into similar pattern clusters for easy management, analysis and interpretation. In the past, because of the lack of good pattern distance measures, this task was not very successful. In particular, both item-matching distances and sample-matching distance overlook certain important factors as discussed above. Hence, they miss to capture their subtle interacting relationship inherent in the data. It was this challenging demand that motivates the development of the new distance measures based on the dual pattern-data relationship as presented earlier in this chapter. Once the entropic distance measure is defined, it not only can enhance pattern clustering, but its related concept of sample-attribute matching can also be applied in the development of more efficient and generalized methods in other post-analysis tasks such as pattern pruning and summarization.

In this chapter, to demonstrate the effectiveness and efficacy of the new distance measure in pattern clustering, both the hierarchical clustering and *k*-means clustering algorithms are implemented as dual clustering processes which cluster patterns as directed by the coherence of their associated data clusters based on the entropic distance developed. Similar to clustering on data, hierarchical pattern clustering produces the entire clustering hierarchy and always produces the same result given the same distance measure. Hence, it is ideal for studying and comparing different distances. However, hierarchical pattern clustering is not scalable. In contrast, *k*-means pattern clustering is fast and scalable since it only produces a partition rather than the entire hierarchy. However, it requires users to set the number of clusters. It also involves a random cluster initialization. Hence, different executions of *k*-means will produce different clustering results, making evaluation and comparison difficult. Despite the limitations of the two methods, they are commonly used in data clustering. I implemented both methods using the proposed distance measures.

3.6.1 Hierarchical clustering

Recalling that $I(i)$ is a data cluster induced by pattern $x_i^{s_i}$. Since there is a one-one correspondence between a pattern and its induced data cluster, $I(i)$ is addressed as the pattern $x_i^{s_i}$ in the algorithm. The major steps in hierarchical pattern clustering are contained in the following procedures [1] – [3], where c is the desired number of final clusters.

```
initialize $c$, $c' := n$, $I(i)$, $i = 1,\ldots, n$
do $c' := c - 1$
  find nearest pattern clusters, say, $I(i)$ and $I(j)$
  merge $I(i)$ and $I(j)$
until $c = c'$
return $c$ clusters
```

Since hierarchical clustering is well-studied, we present it here for reference only. More details can be found in [1], [2].

3.6.2 *K*-means Clustering

The major steps in *k*-means pattern clustering are contained below[1], [2].

```
1.   initialize $k$, $I(i)$, $i = 1,\ldots, n$
2.   take the first $k$ $I(i)$, $i = 1,\ldots, k$ as single-element pattern clusters $c(i)$
3.   for each of the remaining $I(i)$, $i = k+1,\ldots,n$ do
4.     find the nearest pattern cluster $c(j)$ for $I(i)$ and merge them
5.   end {for}
6.   repeat
7.     for $i := 1$ to $n$ do
8.       find the nearest pattern cluster $c(j)$ for $I(i)$
9.       if $I(i)$ is not contained in $c(j)$ then
10.        remove $I(i)$ from its current pattern cluster, say $c(k)$
11.        merge $I$(i) and $c(j)$
12.      end {if}
13.    end {for}
14.  until no new remove and merge operation occurs
15.  return $k$ clusters $c(i)$, $i = 1,\ldots, k$
```

For evaluation and comparison of different pattern distances in pattern clustering experiments, it is desirable to remove the randomness of *k*-means. Hence, in steps 2 – 5, we use a systematical approach to initialize pattern clusters. This approach takes the first *k* patterns as initial single-element pattern clusters and merges the remaining (*n*-*k*) patterns to the nearest pattern clusters. This may not be the best approach for cluster initialization. In fact, a more common one is to randomly initialize the clusters. However, this approach allows us to fairly compare the performance of different pattern distances and hence will be used in our experiments later. Steps 6 – 15 are standard *k*-means procedures [1], [2].

Chapter 4

Pattern Pruning and Summarization

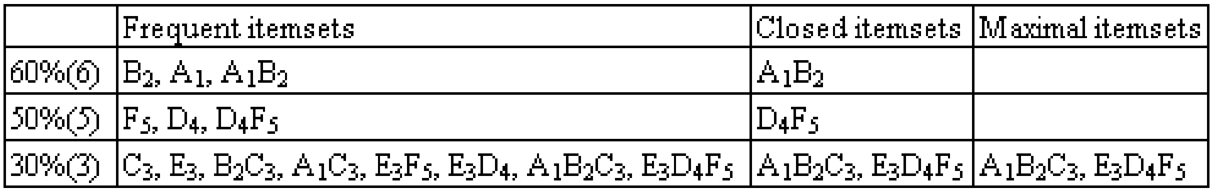
This chapter presents pattern pruning and summarization methods based on the concept of sample-attribute matching described in section 3.4. In section 4.1, an effective pattern pruning method, known as generalized itemset pruning, is developed which allows the users to control the tradeoff between information loss after pruning and the number of patterns pruned. In section 4.2, an effective pattern summarization method is developed which automatically generates an informative yet concise summary for a pattern cluster.

4.1 Pattern Pruning

In practice, pattern clustering itself is not adequate because the number of patterns in each pattern cluster can still be overwhelming. Hence, pattern pruning is usually performed before pattern clustering [8], [9], [20], [23] – [39] or summarization [20], [21]. In this section, section 4.1.1 first reviews the classical closed itemsets pruning [8], [9], [29] – [31] and maximal itemsets pruning [32] – [35] commonly used in pattern pruning in a graphical approach similar to the one in chapter 3. Then, a generalized itemsets pruning method is developed and presented in section 4.1.2. Using the concept of sample-attribute matching introduced in chapter 3, a new and unified perspective of closed itemset and maximal itemsets is formulated from the viewpoint of information loss. Such perspective helps to generalize both closed itemsets pruning and maximal itemsets pruning by considering them as two extreme special cases. It also explains why one can reconstruct all the original frequent itemsets from the closed itemsets alone – closed itemsets pruning does not lose any information. It also indicates that maximal itemsets pruning allows the maximal amount of information loss under certain conditions. The proposed generalized itemset pruning provides an alternative to closed itemset pruning and maximal itemset pruning by allowing the user to control the tradeoff between information loss and the number of patterns being pruned.

4.1.1 Closed and Maximal Itemsets

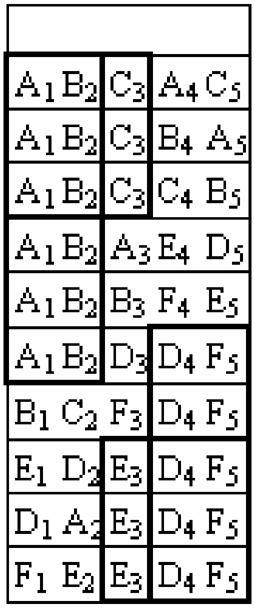
Two common pruning techniques are closed itemset pruning [8], [9], [29] – [31] and maximal itemset pruning [32] – [35]. Both of them are based on sample matching. A frequent itemset $x_i^{s_i}$ is called maximal if it is not a subset of any other frequent itemsets [8]. A frequent itemset $x_i^{s_i}$ is called closed if there exists no proper superset $x_j^{s_j} \supset x_i^{s_i}$ with $|m(i)| = |m(j)|$ [8]. In Figure 10, the 3 frequent itemsets B_2, A_1 and A_1B_2 in Figure 2(c) are reduced to one frequent closed itemset A_1B_2 since all of them are contained in the first 6 samples. By the same token, the 14 itemsets in Figure 2(c) are reduced to only 4 closed itemsets. Hence, 71% of itemsets are pruned. Furthermore, the closed itemsets A_1B_2 and $A_1B_2C_3$ are reduced to one maximal itemset $A_1B_2C_3$ since it is not a subset of other itemsets. Similarly, D_4F_5 and $E_3D_4F_5$ are reduced to $E_3D_4F_5$. Thus, 50% of closed itemsets are pruned.



	Frequent itemsets	Closed itemsets	Maximal itemsets
60%(6)	B_2, A_1, A_1B_2	A_1B_2	
50%(5)	F_5, D_4, D_4F_5	D_4F_5	
30%(3)	C_3, E_3, B_2C_3, A_1C_3, E_3F_5, E_3D_4, $A_1B_2C_3$, $E_3D_4F_5$	$A_1B_2C_3$, $E_3D_4F_5$	$A_1B_2C_3$, $E_3D_4F_5$

Figure 10. An example of frequent itemsets, closed itemsets and maximal itemsets.

It is more intuitive to represent closed and maximal itemsets graphically. In Figure 11, the pattern-induced data clusters of the 4 closed itemsets in Figure 10 are shown as 4 highlighted blocks. Each data cluster contains the pattern-induced data clusters of all its corresponding frequent itemsets. For example, the data cluster of A_1B_2 contains the data clusters of A_1 and B_2. Similarly, the data cluster of $A_1B_2C_3$ contains the data clusters of C_3, B_2C_3 and A_1C_3. In this sense, closed itemset pruning will not lose any information. This is one major reason why it is widely used in pattern pruning. More precisely, closed itemset pruning does not lose information because we can always recover all the pruned frequent itemsets (see the algorithm in [8]). In this paper, we also prove that clustering closed itemsets using distance measures d_T [23], d_G [40], d_R, d_{RC} or d_O will produce equivalent results to clustering all frequent itemsets (see appendix I). Hence, clustering closed itemsets is always better than clustering all frequent itemsets since it is faster and produces same clustering results. This is a good news since closed itemset pruning can prune fairly large number of itemsets, even though the actual number of itemsets pruned depends on the nature of the data sets. In our simple example, closed itemset pruning can reduce 71% of frequent itemsets. In practice, it was found that the number of frequent closed itemsets is usually of an order of magnitude lower than the number of frequent itemsets if the data set is dense [29] – [31].



A_1	B_2	C_3	A_4	C_5
A_1	B_2	C_3	B_4	A_5
A_1	B_2	C_3	C_4	B_5
A_1	B_2	A_3	E_4	D_5
A_1	B_2	B_3	F_4	E_5
A_1	B_2	D_3	D_4	F_5
B_1	C_2	F_3	D_4	F_5
E_1	D_2	E_3	D_4	F_5
D_1	A_2	E_3	D_4	F_5
F_1	E_2	E_3	D_4	F_5

Figure 11. Pattern-induced data clusters of the closed itemsets and maximal itemsets in Figure 10.

The data clusters of maximal itemsets, however, do not contain the data clusters of all its corresponding frequent itemsets. For example, the data cluster of $A_1B_2C_3$ does not contain the data clusters of A_1B_2, A_1 or B_2. Similarly, the data cluster of $E_3D_4F_5$ does not contain the data clusters of D_4F_5, D_4 or F_5. Maximal itemset pruning will lose information since we cannot recover the pruned itemsets [32] – [35] and clustering maximal itemsets does not produce equivalent results to clustering all itemsets. However, maximal itemset pruning prunes more patterns than closed itemsets, even though the number of patterns pruned is also dependent on the data nature.

4.1.2 Generalized Itemsets

While closed itemset pruning is very conservative and information is preserved, maximal itemset pruning is more aggressive and can reduce more patterns (in many cases, many more patterns) than closed itemsets. However, maximal itemset pruning may lose too much information. Hence, we present a new type of itemsets that generalizes these two types of itemsets by including them as two special cases and providing a tradeoff between information loss and pattern number.

We take the approach in section 3 to analyze closed and maximal itemsets. Figure 12 shows 2 pattern-induced data clusters $I(i)$ and $I(j)$ (the 2 highlighted blocks) induced by itemsets $x_i^{s_i}$ and $x_j^{s_j}$ respectively, where s_j is a superset of s_i. If $x_i^{s_i}$ is pruned and $x_j^{s_j}$ is left behind, the shadow area ($r_i \times c_{ij}$) represents the information loss. Hence, one possible measure of information loss is:

$$loss(i, j) = \frac{r_i \times c_{ij}}{\left(r_{ij} + r_i\right) \times c_{ij}} = \frac{r_i}{r_{ij} + r_i} \qquad (19)$$

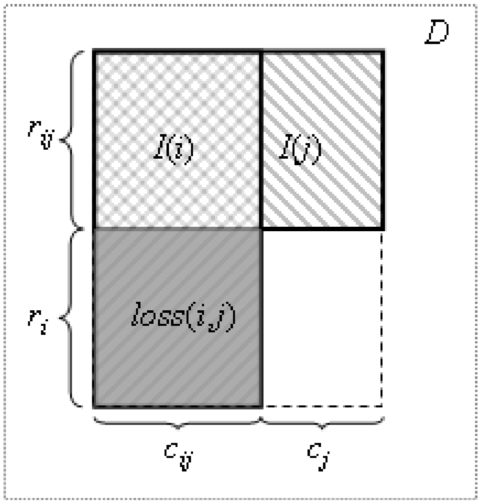


Figure 12. Generalized itemests.

When *loss*(*i*, *j*) = 0 (i.e. no information loss is allowed), $x_j^{s_j}$ becomes a closed itemset. When *loss*(*i*, *j*) = 1 (i.e. maximal information loss is allowed), $x_j^{s_j}$ becomes a maximal itemset (see appendix D for proofs). When 0 < *loss*(*i*, *j*) < 1, $x_j^{s_j}$ is called a generalized itemset. *loss*(*i*, *j*) allows us to control the tradeoff between information loss and number of patterns. Using *loss*(*i*, *j*), a simple algorithm is developed to generate generalized itemsets. The major steps in the algorithm are shown below, where *c* is a user-specified threshold for controlling the maximal information loss allowed. When *c* is 0, the algorithm produces closed itemsets (i.e. 0% information loss is allowed). When *c* is 1, it produces maximal itemsets (i.e. a maximum of 100% information loss is allowed). When 0 < *c* < 1, it produces generalized itemsets having at most *c*×100% information loss.

```
initialize $c$, $x_i^{s_i}$, $i = 1,\ldots, n$
for $i := 1$ to $n$ do
  for $j := 1$ to $n$, $i \neq j$ do
    if $x_i^{s_i}$ is a superset of $x_j^{s_j}$ and $loss(i, j) < c$ then delete $x_j^{s_j}$
  end {for}
```

```
6.    end {for}
7.    return remaining itemsets
```

The complexity of the above algorithm is O(n^2) where n is the number of patterns. The complexity of the operation for checking whether $x_i^{s_i}$ is a superset of $x_j^{s_j}$ is O(|s_i| |s_j|) where |s_i| is the number of items in $x_i^{s_i}$. Since |s_i| is very small (e.g. |s_i| = 3 for $A_1B_2C_3$), the running time of this operation is negligible. According to (19), the complexity of *loss*(i, j) is O(| I |), which is usually larger than O(|s_i| |s_j|) even though it is linear. In step 4, if $x_i^{s_i}$ is not a superset of $x_j^{s_j}$, it is not necessary to calculate *loss*(i, j) and thus the complexity is negligible (or O(|s_i| |s_j|)). It is only when $x_i^{s_i}$ is a superset of $x_j^{s_j}$, the complexity of step 4 is O(| I |). Since each pattern $x_i^{s_i}$ has only a few subset patterns, such situation only occurs a few times for each pattern. Otherwise, step 4 is very fast. Hence, the above algorithm is very efficient for pruning a large number of patterns.

4.2 Pattern Summarization

After pattern pruning, the remaining patterns are clustered using methods introduced in chapter 3. Although it is much easier to interpret the pattern clusters than the original set of patterns, each pattern cluster could still contain quite a large number of patterns. In practice, for a large data set, it is not uncommon that the number of frequent itemsets is more than 20,000 (e.g. Wine, Anneal and Letter data sets obtained from UCI). Pattern pruning could probably reduce half of the itemsets (the pruning performance will be extensively tested in later experiments in section 5.1). However, the number of remaining itemsets could still be more than 10,000. If these itemsets are clustered into 10 clusters, on average, each cluster could contain 1000 itemsets. Hence, if the number of patterns after pruning is still large, methods are needed to summarize each pattern cluster.

As mentioned in chapter 3, an important advantage of the proposed simultaneous pattern and data clustering method is that the dual clusters produced contain an explicit one-to-one correspondence between the pattern clusters and the data clusters. Hence, while it is difficult to analyze the pattern clusters directly, we can analyze their associated data clusters more easily. In this section, effective summarization techniques are proposed to summarize each individual pattern cluster. In section 4.2.1, the classical RuleCover pruning algorithm is reviewed. How the algorithm can be used for summarization and its limitation is also discussed. Section 4.2.2 then introduces a simple method to overcome the limitation of RuleCover algorithm when applying to summarizing pattern clusters. In section 4.2.3, the concept of sample-attribute-matching introduced in chapter 3 is used to develop an effective pattern summarization method. It is referred to as AreaCover, which is adapted from RuleCover. Finally, in section 4.2.4, the algorithm complexity of the summarization methods is discussed. A simple optimization technique is introduced to improve the algorithm speed.

4.2.1 RuleCover Summarization

The objective of pattern summarization is to obtain a small subset of patterns that are representative to other patterns. In the literatures, most research works focus on pattern pruning rather than pattern summarization, although these two problems are highly related. Pattern summarization can be considered as a very aggressive pruning method where most patterns, except for a few representative patterns, are pruned. In practice, patterns are not really pruned. Instead, a few representative patterns are selected to summarize the other patterns. In [23], the RuleCover method was proposed to prune a group of association rules sharing the same consequent. A greedy algorithm was developed to find the close-to-optimal cover. In each iteration, the algorithm selects the rule that covers the largest number of samples which have not been covered by the rules in a rule cover Δ. The selected rule is then put in Δ. The algorithm stops when all samples matched by the original rules in Γ are matched by the rules in the rule cover Δ. For example, in Figure 13, the pattern-induced data clusters *I*(1, 2, … ,7) of a group of 7 rules sharing the consequent A is shown. RuleCover first selects rule 3 which induces *I*(3) since *I*(3) covers the largest number of samples (or rows). RuleCover then selects rule 1 since *I*(1) covers the largest number of samples not covered by rule 3. Note that rules 3 and 1 altogether have covered most samples in *I*(1, 2, … ,7). The remaining uncovered samples are marked by (*) in Figure 13. The matched samples of other rules such as rules 4, 5, 6 and 7 have been covered by rules 3 and 1. Hence, RuleCover finally selects rule 2 which covers the uncovered samples in (*). It then stops since all samples in *I*(1, 2, …,7) have been covered by the rules in the rule cover Δ = {3, 1, 2}.

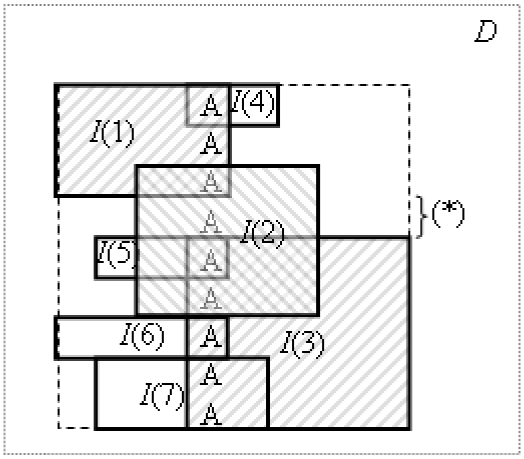


Figure 13. RuleCover

The major steps in the RuleCover algorithm are shown below [23], where Γ is the original set of rules, *m*(*i*) is the set of matched samples of the rule $x_i^{s_i}$ and *ε* is the user-specified constant for specifying the desired portion of samples covered by the rule cover Δ:

```
initialize $\varepsilon$, $\Gamma := \{ x_i^{s_i} \mid i = 1, \ldots, n\}$, $m(i)$, $i = 1, \ldots, n$
$\Delta := \varnothing$
$u := \bigcup_{i=1}^{n} m(i)$
$o = | u | \times (1 - \varepsilon)$
for $i := 1$ to $n$ do
  $u_i := m(i)$
end {for}
repeat
  choose $x_i^{s_i} \in \Gamma$ so that $| u_i |$ is largest
  $\Delta := \Delta \cup x_i^{s_i}$
  $\Gamma := \Gamma \setminus x_i^{s_i}$
  for all $x_i^{s_i} \in \Gamma$ do
    $u_i := u_i \setminus m(i)$
  end {for}
  $u := u \setminus m(i)$
until $| u | \leq o$
return the rule cover $\Delta$
```

The above algorithm gets as input the original set Γ of rules and the set of rows matched by each of these rules $m(i)$. Rule cover Δ is initialized to an empty set. The set u is used to store those samples that are not matched by rules in Δ whereas the sets u_i stores those samples in u that are matched by the rule $x_i^{s_i}$. Iteratively, the rule in Γ that matches most of the samples in u is moved from the rule set Γ to the rule cover Δ. The samples matched by this rule are removed from u. This is repeated until the rules in Δ cover at least $\varepsilon \times 100\%$ of the samples.

The only parameter in RuleCover algorithm is ε, which is called minimum coverage. It specifies the minimum percentage of samples covered by the rule cover Δ. For example, in our example in Figure 13, ε is 100%, indicating that the rule cover will cover 100% of samples in $I(1, 2, \ldots, 7)$. RuleCover is a greedy algorithm. It always selects the rule covering the largest number of samples. Later selected rules will always cover less than the former selected ones. It can be useful to loose ε so that rules covering small uncovered samples are not included in the rule cover. For example, in Figure 13, rule 3 covers around 50% of the samples while rule 1 covers around 30% of the samples. It may be desirable to only cover 80% of the samples and not to move rule 2 to Δ since rule 2 only cover around 20% more samples as marked by (*). The time complexity of RuleCover is polynomial with respect to $\left|\bigcup_{i=1}^{n} m(i)\right|$ [23].

Given a set of rules, RuleCover prunes most rules and only a few patterns that cover the largest portions of samples are retained. RuleCover could be a good method for summarizing patterns. For example, in Figure 13, intuitively, rules 1, 2 and 3 can be used as a high-level summary of all the 7 rules because they cover the 3 major portions of samples (rows) in I(1, 2, …,7). Rules 4, 5, 6, and 7 matches some small portions of samples in I(1, 2, …, 7). They are the details not included in the summary. For convenience, the rules or patterns in the cover Δ are called summary rules or summary patterns respectively.

RuleCover is most useful when applying to a small group of rules/patterns since it prunes aggressively and only a few rules/patterns are retained. That is one reason why it was originally applied to a group of association rules sharing the same consequent instead of all association rules [23]. In effect, the association rules are clustered accordingly to the consequent before applying RuleCover. The condition of same consequent is a strict grouping criterion since only rules sharing the same consequent are grouped together. Such criterion results in many small groups of rules, each of which is pruned by RuleCover separately. Hence, although RuleCover prunes most rules in each group, the total number of rules retains will not be very small. In general, the criterion of same consequent is not necessarily appropriate for clustering rules. This criterion is only used together with RuleCover algorithm to prune association rules. In [23], after applying RuleCover to prune association rules, the pattern distance d_T was used for pattern clustering. The criterion of same consequent is not used in pattern clustering.

In our case, we cluster patterns as described in chapter 3 before applying RuleCover. The criterion of same consequent (more precisely, consequent should be item here since a rule is considered as a pattern as discussed in section 3.2) is not used since it is not a good criterion for clustering similar patterns. In fact, such criterion is essentially comparing the common items shared by two patterns. As mentioned earlier, such approach may miss certain relationship between patterns and may produce misleading results. Moreover, ideally, like the number of classes of most real-world data sets is usually small, the number of clusters produced should not be too large. Otherwise, it is very time-consuming to analyze all of them. However, the criterion may produce many small pattern clusters.

4.2.2 Multi-level RuleCover Summarization

When applying RuleCover to a pattern cluster, it could be too aggressive for large clusters (e.g. a cluster of 100 patterns). It may select only a few large and trivial patterns and miss other important patterns in a pattern cluster. In Figure 14, patterns 8 and 9 are added to the 7 patterns (patterns and rules are considered the same as discussed in section 3.2) in Figure 13. Note that the pattern clusters produced by the clustering methods in chapter 3 can contain patterns not sharing any item. RuleCover will produce either Δ = {9, 1} or Δ = {9, 8} depending on the implementation of the greedy algorithm. Regardless of which Δs it produces, the rule cover Δ, as a summary, misses many important patterns (e.g. patterns 2 and 3) in the cluster. The problem is that pattern 9 covers most patterns, including patterns 2 and 3. Just like trivial and large clusters may cover other clusters in data clustering, large patterns may cover other patterns in pattern summarization. The resulting summary may be trivial or not detailed enough.

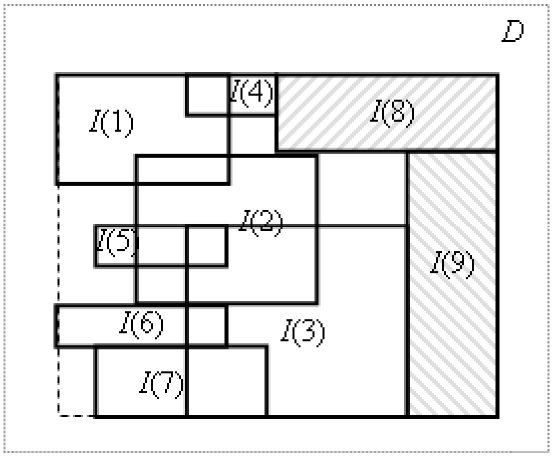


Figure 14. Problem of RuleCover

One simple method to solve the above problem is to remove the rules produced by the first run of RuleCover and then apply RuleCover again to find the second level of rules that summarize the other rules. For example, in the above example, suppose the RuleCover produces $\Delta_1 = \{9, 8\}$ in the first run (the subscript 1 in Δ_1 indicate that this rule cover is produced in the first run). Then, after removing rules 8 and 9, RuleCover will produce $\Delta_2 = \{3, 2, 1\}$ in the second run. Note that if RuleCover produce $\Delta_1 = \{9, 1\}$ in the first run, it will produce $\Delta_2 = \{3, 2, 8\}$ in the second run. This procedure can be repeated. In the third run, RuleCover will produce $\Delta_3 = \{4, 5, 6, 7\}$. Note that the rules in Δ_3 do not cover all samples $I(1, 2,\ldots, 7)$. The resulting algorithm is called multi-level RuleCover since it runs RuleCover with different levels (each run of RuleCover is one level).

The major steps in multi-level RuleCover are contained in the following procedures where L is the level (the Lth run) of RuleCover:

```
initialize $\varepsilon$, $L$, $\Gamma := \{ x_i^{s_i} \mid i = 1, \ldots, n\}$, $m(i)$, $i = 1, \ldots, n$
**for** $i := 1$ to $L$ **do**
  $\Delta_i :=$ RuleCover($\varepsilon$, $\Gamma$, $\{m(i)\}$)
  $\Gamma := \Gamma \setminus \Delta_i$
**end** {for}
return $\Delta_i$ , $i = 1, \ldots, L$
```

The algorithm runs RuleCover for L times. After each run of RuleCover, it removes the patterns in Δ_i from Γ so that the patterns in Δ_i will not be selected again. If L is large enough, eventually all patterns will be selected. In effect, in the above algorithm, L controls the depth of the search and ε controls the width of the search at each level.

4.2.3 AreaCover Summarization

RuleCover is a sample-matching based method since it only covers the samples in the cluster. It can be easily extended to cover both the sets of matched samples and matched attributes. The resulting algorithm is called AreaCover since it considers the both the matched samples and matched attributes (i.e. an area) of each pattern. The major steps in AreaCover are contained in the following procedures:

```
initialize $\varepsilon$, $\Gamma := \{ x_i^{s_i} \mid i = 1, \ldots, n\}$, $I(i)$, $i = 1, \ldots, n$
$\Delta := \varnothing$
$u := I(1, \ldots, n)$
$o = | u | \times (1 - \varepsilon)$
**for** $i := 1$ to $n$ **do**
  $u_i := I(i)$
**end** {for}
**repeat**
  choose $x_i^{s_i} \in \Gamma$ so that $| u_i |$ is largest
  $\Delta := \Delta \cup x_i^{s_i}$
  $\Gamma := \Gamma \setminus x_i^{s_i}$
  **for** all $x_i^{s_i} \in \Gamma$ **do**
```

```
13.       $u_i := u_i \setminus I(i)$
14.     **end** {for}
15.     $u := u \setminus I(i)$
16.  **until** $|\, u \,| \leq o$
17.  return the rule cover $\Delta$
```

Basically, AreaCover is very similar to RuleCover. The only difference is that *m*(*i*) is replaced by *I*(*i*) since AreaCover considers both the set of matched samples and matched attributes. The algorithm is shown here for completeness.

As an illustration of AreaCover, in Figure 13, AreaCover will select, in the order of the sequences, *I*(3), *I*(1), *I*(2), *I*(7), *I*(6), *I*(4) | *I*(5), where *I*(4) | *I*(5) means that either *I*(4) or *I*(5) is selected depending on the implementation of the algorithm since their uncovered areas are the same. Note that ε can be used to control the number of patterns in area cover Δ. For example, since *I*(1), *I*(2) and *I*(3) occupies approximately 60% of the total area of *I*(1, …, 7), ε = 60% will produce Δ = {3, 1, 2}. Moreover, in Figure 14, *I*(9) will not hide the patterns *I*(2), *I*(3), *I*(5), *I*(6) and *I*(7) if AreaCover is used. Hence, AreaCover is less prone to the problem of being hidden by large patterns since it considers matched samples as well as matched attributes. However, if necessary, AreaCover can also run multiple times as in multi-level RuleCover. The algorithm is exactly the same as the algorithm in multi-level RuleCover except that RuleCover is replaced by AreaCover.

4.2.4 Algorithm Complexity

A final remark in this section is about the time complexity of RuleCover and AreaCover. The time complexity of RuleCover reported in [23] is polynomial with respect to $\left|\bigcup_{i=1}^{n} m(i)\right|$. However, a simple optimization can be done in both RuleCover and AreaCover by adding the following statement after updating u_i (between step 13 and step 14 in RuleCover and AreaCover):

```
A 、          **if** $|\, u_i \,| = 0$ **then** $\Gamma := \Gamma \setminus x_i^{s_i}$
```

The above statement removes pattern $x_i^{s_i}$ from Γ if the number of the uncovered matched samples | u_i | is 0 (| u_i | is the size of the uncovered matached area in AreaCover). The above operation is only of constant time. However, it could significantly speeds up the RuleCover (or AreaCover) algorithm depending on how the patterns overlap in the cluster. For example, in Figure 14, in level 1 (first run) of RuleCover, once pattern 9 is moved to Δ_1, patterns 2, 3, 5, 6 and 7 will be removed from Γ by the above statement since their uncovered matched samples | u_i | is 0. Likewise, in level 2 (second run), once pattern 3 is moved to Δ_2, patterns 5, 6 and 7 are removed immediately. The effect of the above statement is most significant in the lower levels (i.e. the first few runs). If there are a few large patterns in a given level, RuleCover and AreaCover can be significantly speeded up. However, in the worst case where all patterns have their own uncovered samples, the above statement cannot speed up the algorithm. A possible worst case scenario is shown in Figure 15 for RuleCover. An example of worst case scenario for AreaCover is Figure 14.

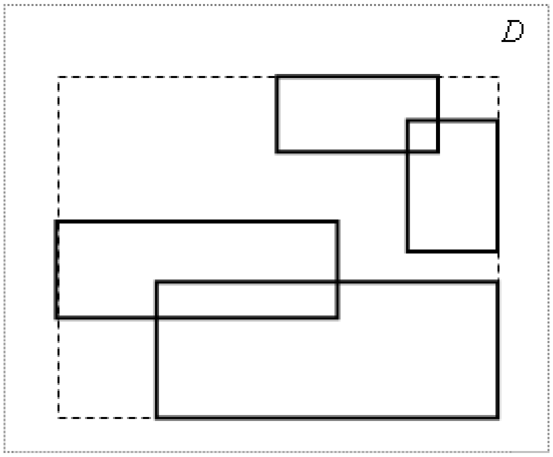


Figure 15. Worst case scenario of the optimization method for RuleCover

Chapter 5
Experiments

The experimental results of generalized itemsets, closed itemsets [8], [9], [29] – [31] and maximal [32] – [35] itemsets are provided for 10 real-world data sets in section 5.1. In section 5.2, artificial data is generated to study the difference of various distance measures. In section 5.3, 10 real-world data is used to test the distance measures using both hierarchical clustering and *k*-means.

5.1 Experiments of Pattern Pruning on Real-World Data Sets

Table 4 reports the pruning results of closed itemset pruning (or simply closed pruning), generalized itemset pruning (or generalized pruning) and maximal itemset pruning (maximal pruning) on 10 real-world data sets obtained from UCI Repository [19] when minimum support is 2%. The table consists of 6 columns. Column 1 shows the data set names. Column 2 gives the number of frequent itemsets. Column 3 gives the number of closed itemsets. Column 4 gives the number of generalized itemsets with maximal information loss (i.e. *c*× 100%) set to 25%, 50% and 75%. Column 5 gives the number of maximal itemsets. Finally, column 6 reports the average running times.

Many data sets contain numeric attributes. We discretize these attributes into intervals. There are a number of discretization algorithms in literatures for this purpose. We use MLC++ [117] with the default settings. The experiments were run on a laptop computer with Intel Premium 1.73GHz and 1GB RAM. For pruning, all experiments finished in 4 minutes. In Tables 4 and 5, the number of 75% generalized itemsets is almost the same as the number of maximal itemsets. This is reasonable because 100% information loss means there is no common area between the induced data clusters of the pruned itemsets and the renaming maximal itemsets. This is impossible since the pruned itemests are the subset of maximal itemsets, implying that their induced data clusters must have some common areas. The results show that most pruned patterns share at least 25% of the common data with the maximal itemsets. The experimental results are summarized below:

Car: This is a car evaluation data set consisting of 1782 samples, 6 attributes and 4 classes. A car is evaluated based on factors such as price, maintenance fee, etc. In Table 4, closed pruning reduces about half ((1156 – 557) / 1156 × 100% = 52%) of the frequent itemsets. This result is desirable since we can cluster only half of the itemsets and obtain equivalent results as clustering all of them. If further pruning is preferred, maximal pruning reduces 83% of the itemsets. It is interesting to observe that 25% generalized pruning reduces 76%. Thus, maximal pruning sacrifices a possible 75% more information loss only for pruning an additional 6% (or 276–202=74) of the itemsets.

Tic-Tac: This data set encodes the complete set of possible board configurations at the end of tic-tac-toe games. It has 958 samples, 9 attributes and 2 classes. This data set is a good example where closed pruning does not perform well. In fact, from Tables 4 and 5, closed pruning cannot prune any itemsets since the data set is not dense. In Table 4, maximal pruning can reduce 40% of the itemsets. However, it suffers from large amount of information loss. In contrast, 50% generalized pruning has already pruned 36%. This may be a better choice if we want to avoid too much information loss.

Liver-disorder: This data set consists of 5 blood tests results and other information to classify liver disorders arisen from excessive alcohol consumption. It consists of 345 samples, 7 attributes and 2 classes. From Table 4, very few frequent itemsets (only 1%) are pruned in closed pruning. In contrast, maximal pruning can prune 53% of the frequent itemsets. A better choice may be 50% generalized pruning since it has already pruned 44% of the itemsets. This data set shows that, when not many itemsets are pruned in closed pruning and too many itemsets are pruned in maximal pruning, generalized pruning provides an alternative method for us to control the tradeoff.

Wine: This data set contains a chemical analysis result of wines grown in the same region in Italy but derived from 3 different cultivars. The data is used to determine the quantities of 13 constituents found in each of the 3 types of wines. It has 178 samples, 13 attributes and 3 classes. From Table 4, closed pruning reduces 33% of the itemsets, while maximal pruning reduces 44%. In between, 25% generalized pruning have already pruned 43%. This may be a better choice than maximal pruning.

Glass: This is a glass identification data set consisting of 214 samples, 9 attributes and 6 classes (an Id# attribute was removed). The study of glass types was motivated by criminological investigation. At the scene of the crime, a correctly identified glass can be used as evidence. From Table 4, closed pruning reduces 37% of the itemsets while maximal pruning reduces 65%. 50% generalized pruning only prunes 1 itemset less than maximal pruning. This is definitely more preferable. 25% generalized pruning may also be a good choice since it only prunes 26 less than maximal pruning.

Breast Cancer: This data set consists of 683 samples, 9 attributes and 2 classes. It is used to classify whether a patient has breast cancer or not. The 2 classes are known to be linearly separable. From Table 4, closed pruning reduces 11% of the frequent itemsets while maximal pruning reduces 61%. Again, a better choice balancing the tradeoff between information loss and number of itemsets may be 25% or 50% generalized pruning, which reduces 58% and 61% of the itemsets respectively.

Auto: This data set is about city-cycle fuel consumption and is used in the 1983 American Statistical Association Exposition. It has 398 samples, 8 attributes and no class attribute. From Table 4, closed pruning reduces 44% of the frequent itemsets while maximal pruning reduces 78%. 50% generalized pruning reduces 77%, which is evidently better than maximal pruning.

Anneal: This data set contains 798 samples, 38 attributes and 6 classes. From Table 4, closed pruning reduces 76.2% of itemsets while maximal pruning reduces 80.3%. This data set is an example where closed pruning performs very well (76.2% is a huge pruning) and maximal pruning cannot prune many more (only 4.1% more). Thus, closed pruning is a good choice. Alternatively, 25% generalized pruning could be used since it only prunes 0.1% less than maximal pruning.

Letter: This is a data set of characters image features. It has 20000 samples, 16 attributes and 26 classes. From Table 4, 0.4% of the itemsets are pruned in closed pruning while 49.1% are pruned in maximal pruning. The pruning results are opposite to that obtained in Anneal data. In this case, the pruning result of closed pruning is not that good while that of maximal pruning looks very good. However, 25% and 50% generalized pruning reduces 39.4% and 48.9% of the itemsets respectively. Both of them seem better than maximal pruning if we want to avoid too much information loss. In particular, 50% generalized pruning only prunes 0.2% less than maximal pruning.

Iris: It consists of 150 samples, 4 attributes and 3 classes (Setosa, Versicolor and Virginica). The classes Versicolor and Virginica are highly overlapped while the class Setosa is linearly separable from the other two. Only 90 itemsets are discovered. Given the small number of itemsets, pruning is not necessary. However, it is still good to use the 50 closed itemsets since clustering them produces the same results as clustering all itemsets and it is easier to interrupt less itemsets. This data set is included in the experiments mainly because it is probably the best known data set in the literatures.

Table 4. Number of Patterns Remained after Pruning (minimum support = 2%)

Data sets	Freq Itemsets	Closed Itemsets	Generalized itemsets			Maximal Itemsets	Average execution time (sec)
		0%	25%	50%	75%	100%	
Car	1156	**557**	276	202	202	202	0.032
Tic-Tac	4024	4024	3908	2556	2404	2404	3.235
Liver disorder	645	**636**	568	360	303	303	0.020
Wine	27503	18303	15406	15289	15289	15289	169.807
Glass	3791	**2045**	1437	1311	1310	1310	0.224
Breast Cancer	3195	2854	**1353**	1253	1252	1252	0.516
Auto	1157	**646**	402	269	256	256	0.031
Anneal	23797	13509	12963	12955	12955	12955	89.886
Letter	21340	21244	12936	10910	10869	10869	190.068
Iris	90	**50**	32	16	15	15	0.010

Highlighted number marks the set of itemsets that will be used for hierarchical clustering later.

Table 5 reports the pruning results when minimum support is 3%. The number of itemsets is reduced. The behavior of the results in this table is very similar to Table 4.

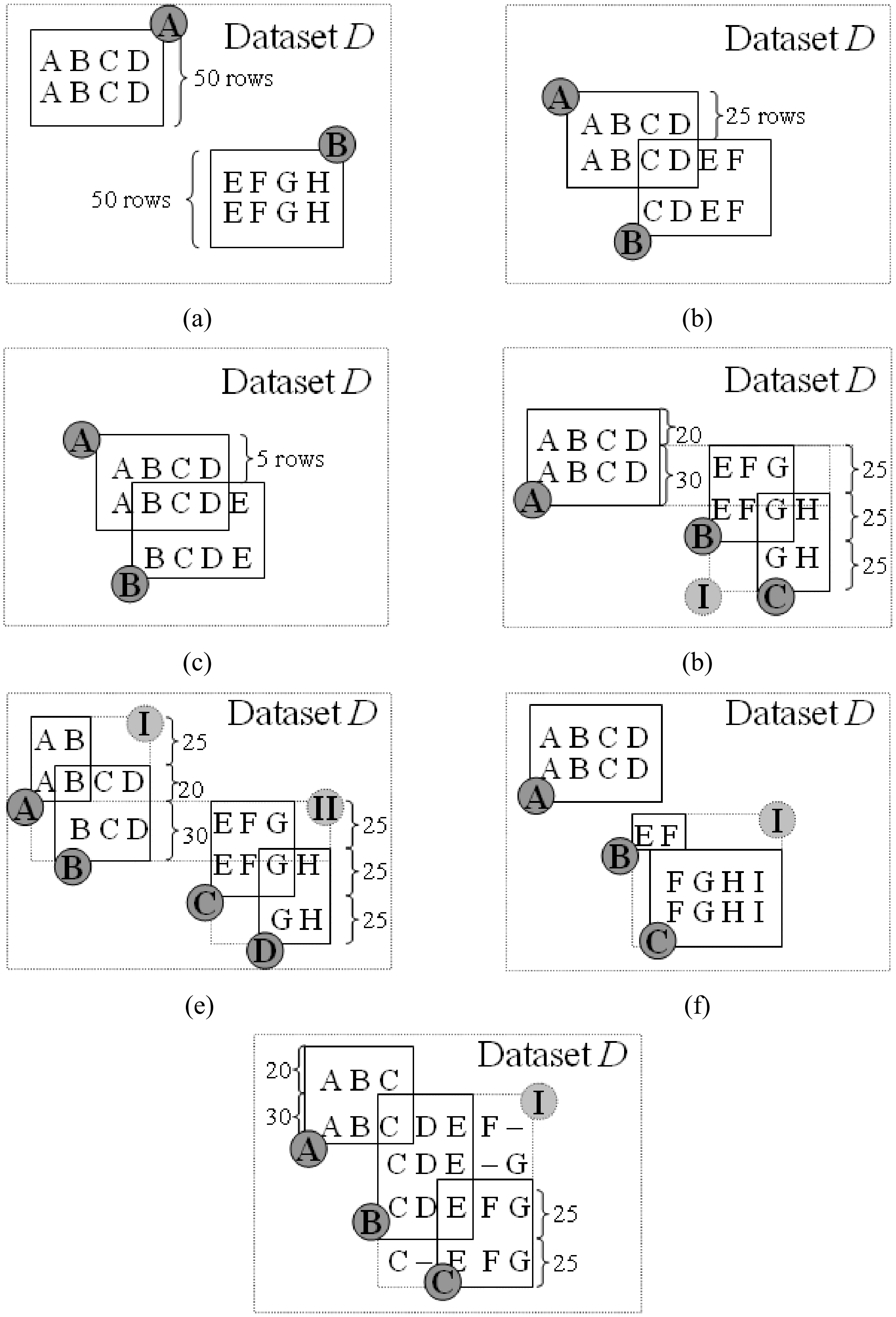
Table 5. Number of Patterns Remained after Pruning (minimum support = 3%)

Data sets	Freq itemsets	Closed itemsets	Generalized itemsets			Maximal itemsets	Average execution time (sec)
		0%	25%	50%	75%	100%	
Car	593	313	144	102	102	102	0.015
Tic-Tac	2080	**2080**	2060	1322	1174	1174	0.267
Liver disorder	335	335	311	219	200	200	0.005
Wine	17845	11842	9365	9300	9300	9300	58.870
Glass	2590	1457	946	866	865	865	0.114
Breast Cancer	2020	1973	845	804	804	804	0.219
Auto	384	384	236	158	154	154	0.016
Anneal	14657	7943	7454	7448	7448	7448	30.578
Letter	9807	9785	5743	4841	4828	4821	43.193
Iris	86	48	30	14	13	13	0.001

Highlighted number marks the set of itemsets that will be used for hierarchical clustering later.

5.2 Experiments of Pattern Clustering on Artificial Data Sets

In this section, 7 artificial data sets are generated to study the 5 measures. This provides insights into why one measure may work better than others in some circumstances. In Figure 16, the 7 data sets have 15 attributes and 500 samples. In Figure 16(a), the data set contains 2 well-separated clusters A and B. Both clusters have 4 attributes and 50 samples, with 10% (uniformly distributed) random noises added. As expected, all measures consider clusters A and B as separated. The 2 clusters are then made closer to each other. They share 2 attributes and 25 samples in Figure 16(b) and 3 attributes and 75 samples in Figure 16(c). It turns out that all measures can still separate them.



(a)

(b)

(c)

(b)

(e)

(f)

(g)

Figure 16. Seven artificial data sets for the study of different distance measures.

In Figure 16(d), the data set contains 3 clusters A, B and C. Clusters B and C are originally taken from a larger cluster I which consists of 4 attributes and 75 samples (the enclosed dashed-line rectangle). Clusters B and C are obtained by adding random noises to cluster I. We want to see whether or not the measures can differentiate that clusters B and C come from cluster I. It turns out that d_T [23], d_G [40] and d_R group clusters A and B together even though they do not share any attribute. Cluster C is left alone. This is because clusters A and B shares 30 samples, more than the sample shared between clusters B and C (25 samples). In contrast, d_{RC} and d_O groups cluster B and C together. d_{RC} does so because clusters B and C share 1 attribute while clusters A and B do not share any. d_O also does so not only because clusters B and C share 1 attribute but also the common attribute share the same value G, giving lower entropy. By the same token, in Figure 16(e), d_T [23], d_G [40] and d_R group clusters B and C together even though they do not share any attribute. Cluster A and D are left alone. In contrast, d_{RC} and d_O groups cluster A and B as well as C and D together. Figures 16(d) and 16(e) shows that sample-matching distances may be misleading because they overlook the contribution from attributes and data variation.

In Figure 16(f), clusters B and C are originally taken from cluster I and share the attribute value F. Although clusters A and B also share 1 attribute, their attribute values (D in cluster A and E in cluster B) are different. d_T [23], d_G [40] and d_R consider clusters A and B having equal distances as clusters B and C because both cluster pairs do not share any sample. Hence, depending on the order of input patterns, these measures may group clusters A and B first, leaving cluster C alone. d_{RC} also consider the 2 cluster pairs as having equal distance since they both share 1 attribute. In contrast, d_O groups clusters B and C first because their shared attribute have the same value F.

In Figure 16(g), again, clusters B and C come from cluster I with random noises added to its corners. d_T [23], d_G [40] and d_R group clusters A and B and leave cluster C alone because clusters A and B share 30 samples, more than the samples shared between clusters B and C (25 samples). d_{RC} also makes the same mistake because both cluster pairs share 1 attribute. However, measures d_O always groups clusters B and C first because the values in the top-right and bottom-left corners between clusters B and C are more consistent with the clusters, giving lower entropy. The last 2 data sets in Figures 16(f) and 16(g) show that measures based on data variation can distinguish difference that may be overlooked by measures based only on matched samples and/or attributes.

5.3 Experiments of Pattern Clustering on Real-World Data Sets

To evaluate the proposed distances, the 10 real-world data sets in section 5.1 are used. Two popular measures of cluster "goodness" or quality, namely, cluster entropy [1] and Minkowski scores (*MS*) [1], [118], are reported for performance evaluation. Cluster entropy and *MS* scores are measures of the quality of a clustering solution given the true clustering. In all experiments, class labels are deleted and are only used for evaluation purpose. Suppose that a data set containing *S* classes is clustered into *K* clusters. Let n_k be the number of samples in the *k*th cluster and n_{sk} be the number of samples from the *s*th class in the *k*th cluster. The entropy of the *k*th cluster is defined as:

$$H_k = -\sum_{s=1}^{S} \frac{n_{sk}}{n_k} \log\left(\frac{n_{sk}}{n_k}\right) \tag{20}$$

The total entropy for the set of *K* clusters is given by

$$Ent = \sum_{k=1}^{K} \frac{n_k}{M} H_k \tag{21}$$

where *M* is the total number of samples in the data set.

The cluster entropy is a measure of the class purity of the clusters. The optimum value is 0, with lower values being "better". *MS* score measures the consistency between a clustering solution and a given true clustering. To define *MS*, let *T* be the "true" solution and *C* the solution we wish to measure. Let n_{11} denote the number of pairs of samples that are in the same cluster in both *C* and *T*. Let n_{01} denote the number of pairs that are in the same cluster only in *C* and n_{10} denote the number of pairs that are in the same cluster in *T*. Minkowski score (*MS*) is then defined as

$$MS(T,C) = \sqrt{\frac{n_{01} + n_{10}}{n_{11} + n_{10}}} \tag{22}$$

For *MS*, the optimum score is also 0, with lower scores being "better".

Both hierarchical clustering and *k*-mean described in section 3.6 are implemented using the 5 distance measures. To remove the randomness in *k*-means, we use the systematic cluster initialization procedures described in section 3.6.2. Ideally, closed itemsets should be used for pattern clustering since clustering them produces equivalent results to clustering all itemsets. However, the numbers of closed itemsets of some data sets are too large for hierarchical clustering. To demonstrate that the distance measures can be applied to hierarchical-type clustering methods, we select around 2,000 itemsets for hierarchical clustering. Later, *k*-means is used to cluster all closed itemsets. The selected itemsets are highlighted in Tables 4 and 5. For Wine, Anneal and Letter data, a hard limit of 2,000 is set to obtain the 2,000 closed itemsets having the highest supports. The number of itemsets used for clustering is shown in the second column of Table 6.

Cluster entropy and *MS* score have different values for different number of clusters. In particular, cluster entropy tends to be smaller for more clusters since more clusters imply smaller and purer clusters. Hence, to fairly compare the cluster entropy and *MS* scores of the 5 measures, the cluster number must be the same. Hence, we set the cluster number as the number of classes when running hierarchical clustering. In later experiments, the cluster number will be automatically determined. Here, we prescribe the cluster number only for the sake of comparison. Table 6 reports the cluster entropy (*Ent*) and *MS* scores of the 5 distance measures. The table consists of 8 columns. Column 1 is the data set name. Column 2 is the number of itemsets used for clustering. Column 3 is the prescribed number of clusters. Columns 4 – 8 contain both *Ent* and *MS* scores for each distance measure. Note that we set the cluster number as 5 for Anneal data even though Anneal data has 6 classes since class 4 has no sample. In addition, for d_{RC}, w_r and w_c are set to 0.5 in all experiments.

From Table 6, for Car data, the *Ent* of d_G and d_R are the best while the *MS* of d_O is the best. Hence, the clusters obtained by d_G and d_R are purer while the clusters obtained by d_O best match the class labels. No single distance measure obtains both the best *Ent* and *MS*. For Tic-Tac data, the *MS* of d_T is the best. d_O obtains the best *Ent* and the second best *MS*. Considering both *Ent* and *MS*, the performance of d_O is the best. For Liver data, the *Ent* of all distance measures are very closed (with only 0.004 difference). d_O obtains the best *MS* score. For Wine data, both the *Ent* and *MS* values of all measures are very closed (with 0.01 difference in *Ent* and 0.005 difference in *MS*). Hence, all of them perform comparably. For Glass data, d_{RC} obtains the best *Ent* and *MS*. For Cancer, Auto and Iris data, d_O obtains the best *Ent* and *MS*. For Anneal data, d_T obtains the best *Ent* while d_{RC} obtains the best *MS*. Finally, for Letter data, d_{RC} obtains the best *Ent* while d_O obtains the best *MS*.

Among the 10 data sets, d_O performs the best in terms of *Ent* and *MS* in 3 data sets and works fairly well in other data sets. It is not known why d_T seems to perform better than d_G in some cases since d_G is just a normalized version of d_T. One interesting observation is that the performance of d_G and d_R are very similar. This can be explained by the fact that they measure distances very similarly. The only difference is that d_R takes the dissimilarity into account while d_G does not. However, the dissimilarity has been mostly reflected by the similarity. Another interesting observation is that d_{RC} obtains the same *Ent* and *MS* as d_G and d_R in Iris data. It is not a coincidence because Iris data only has 4 attributes. Recall that d_{RC} is a sample-attribute-matching distance. When the dimension is low, it behaves like a sample-matching distance such as d_G and d_R.

Table 6. Cluster Entropy and MS (Hierarchical Clustering and Prescribed Cluster Numbers)

Data sets	Item sets#	Clu #	d_T [23]		d_G [40]		d_R		d_{RC}		d_O	
			Ent	*MS*	*Ent*	*MS*	*Ent*	*MS*	*Ent*	*MS*	*Ent*	*MS*
Car	557	4	0.892	1.074	**0.872**	1.084	**0.872**	1.084	0.897	1.094	0.901	**0.920**
TicTac	2080	2	0.639	**0.901**	0.645	1.112	0.645	1.102	0.647	1.031	**0.543**	0.911
Liver	636	2	**0.677**	1.028	0.679	1.178	0.679	1.178	0.680	0.898	0.681	**0.694**
Wine	2000	3	**0.968**	1.203	0.978	1.199	0.978	**1.198**	0.969	1.201	0.973	1.199
Glass	2045	6	1.436	1.685	1.352	1.688	1.352	1.688	**1.206**	**1.508**	1.396	1.689
Cancer	1353	2	0.527	0.725	0.640	0.909	0.640	0.909	0.645	0.909	**0.428**	**0.650**
Auto	646	5	1.507	1.841	1.537	2.000	1.537	2.000	1.550	2.000	**1.382**	**1.724**
Anneal	2000	5	**0.710**	0.797	0.798	0.786	0.798	0.788	0.799	**0.725**	0.750	0.766
Letter	2000	26	3.102	1.923	3.005	2.101	3.005	2.012	**2.966**	1.901	3.166	**1.823**
Iris	50	3	0.570	0.757	0.532	0.865	0.532	0.865	0.532	0.865	**0.364**	**0.611**

Highlighted number marks the lowest *Ent* or *MS* values among the 5 distances for each data set.

The above results are obtained by prescribing the number of clusters. In practice, we do not assume that the number of classes is known. In hierarchical clustering, a common and simple method to determine the number of clusters is to stop clustering if the distance is noticeably increased after merging. The method is implemented to automatically stop hierarchical clustering. The resulting cluster entropy (*Ent*) and *MS* scores are reported in Table 7.

From Table 7, for Car data, d_O automatically produce 4 clusters, which is the actual number of clusters. It also obtains the best *MS*. The correct cluster number and best *MS* values indicate that d_O obtains the best clustering results. Note that, although d_{RC} obtains the best *Ent*, it produces 98 clusters. Since more clusters tend to have smaller and purer clusters, its lowest *Ent* does not mean that it is the best. Similar situation occurs in Liver, Wine, Cancer, Auto and Iris where d_O obtains the closest cluster numbers and the best *MS* (the second best *MS* in Wine), indicating that it performs very well in these data sets (in Liver and Auto data, no distance measure obtains the correct cluster number. However, d_O gets the closest one.). In some data sets (e.g. Liver, Cancer and Iris), d_O even obtains the best *Ent*. In this set of experiments, it is evident that d_O is better than the other measures. For the rest of the data sets, in Tic-Tac data, d_T automatically produce the correct number of clusters (2). However, d_O obtains 3, very close to 2. It also obtains the best *MS*. Hence, d_O performs fairly well in Tic-Tac data. For Glass data, d_{RC} obtains the correct cluster number and the best *MS*. Hence, it performs the best in this data set. For Anneal data, no distance obtains the correct cluster number. However, d_R and d_{RC} obtain the closest ones with d_{RC} having the best *MS*. For Letter data, d_R obtains the closest cluster number while d_O obtains the best *MS*.

Table 7. Cluster Entropy and MS (Hierarchical Clustering and Automatic Termination)

Data sets	d_T [23]			d_G [40]			d_R			d_{RC}			d_O		
	#	*Ent*	*MS*	#	*Ent*	*MS*	#	*Ent*	*MS*	#	*Ent*	*MS*	#	*Ent*	*MS*
Car	3	0.892	1.084	23	0.672	1.084	5	0.824	1.084	98	**0.612**	0.994	**4**	0.901	**0.920**
TicTac	**2**	0.639	0.901	58	0.628	0.871	4	0.644	0.818	83	**0.600**	0.911	3	0.643	**0.621**
Liver	**3**	0.671	0.895	25	0.635	0.988	4	0.674	0.976	6	0.632	0.974	**3**	**0.479**	**0.778**
Wine	2	0.995	1.397	12	**0.813**	1.372	**3**	0.978	**1.198**	5	0.920	1.396	**3**	0.973	1.199
Glass	7	1.447	1.681	41	**1.158**	1.528	3	1.465	1.657	**6**	1.206	**1.508**	3	1.237	1.518
Cancer	**2**	0.527	0.725	9	0.594	0.861	3	0.631	0.909	8	0.503	0.910	**2**	**0.428**	**0.650**
Auto	3	1.507	1.841	8	**1.162**	1.405	2	1.600	2.000	7	1.208	2.000	**4**	1.316	**1.351**
Anneal	3	0.712	0.816	11	**0.450**	0.997	**4**	0.779	0.726	**6**	0.595	**0.719**	3	0.783	0.817
Letter	9	3.451	1.293	39	2.924	1.194	**23**	3.141	1.215	47	**2.856**	1.274	21	3.234	**1.189**
Iris	**3**	0.570	0.757	5	0.516	0.837	**3**	0.532	0.865	4	0.514	0.813	**3**	**0.364**	**0.611**

Highlighted number marks the cluster number closest to the actual cluster number and the lowest *Ent* or *MS* values of each data set.

The problem of hierarchical clustering is that it is not scalable. Hence, we implemented *k*-means clustering for clustering a large set of itemsets. Table 8 reports the cluster entropy (*Ent*) and *MS* scores of the 5 distance measures when *k*-means is used to cluster all closed itemsets with 2% minimum support. In practice, the cluster number *k* can be determined by a number of methods. A simple and common one is to try different values of *k* and choose the best one. Other more sophisticated methods include ISODATA and its variants [1], [2]. All these methods involve heuristics that make the comparison of distance measures difficult. For comparison purpose, we simply set *k* equal to the class number. However, it is possible for applying the measures to these methods.

From Table 8, among the 10 data sets, d_O obtains the best *Ent* in 5 data sets and the best *MS* in 4 sets. In particular, it obtains both the best *Ent* and *MS* in Cancer and Iris data. Although d_O does not obtain the best scores in other data sets, it still performs fairly well. For example, it obtains the second best *Ent* and *MS* in 5 data sets.

Table 8. Cluster Entropy and MS (*K*-means and Prescribed Cluster Numbers)

Data sets	Item sets #	Clu #	d_T [23]		d_G [40]		d_R		d_{RC}		d_O	
			Ent	*MS*	*Ent*	*MS*	*Ent*	*MS*	*Ent*	*MS*	*Ent*	*MS*
Car	557	4	0.896	1.028	0.879	**1.010**	0.879	**1.010**	0.867	1.019	**0.854**	1.016
TicTac	4024	2	**0.644**	1.056	0.651	0.911	0.649	0.910	0.646	0.813	0.646	**0.654**
Liver	636	2	0.682	0.978	**0.680**	0.977	0.682	0.963	0.681	**0.961**	0.683	0.963
Wine	18303	3	1.061	1.382	1.051	1.396	1.049	1.340	1.049	**1.315**	**1.043**	1.332
Glass	2045	6	**1.472**	1.686	1.516	1.689	1.514	1.691	1.504	1.689	1.490	**1.675**
Cancer	2854	2	0.581	0.828	0.589	0.771	0.594	0.765	0.590	0.773	**0.541**	**0.678**
Auto	646	5	1.505	1.864	1.479	**1.767**	1.485	1.771	1.488	1.788	**1.461**	1.774
Anneal	13509	5	0.822	**0.807**	0.825	0.826	**0.819**	0.815	0.891	0.817	0.820	0.812
Letter	21244	26	**2.089**	1.542	2.215	1.732	2.221	1.725	2.207	**1.532**	2.104	1.549
Iris	50	3	0.581	0.730	0.483	0.680	0.483	0.680	0.481	0.655	**0.462**	**0.589**

Highlighted number marks the lowest *Ent* or *MS* values among the 5 distances for each data set.

For Tables 6 and 8, the execution times of the 2 clustering algorithms are reported in Table 9. We report the execution times of the experiments in these 2 tables because they have the same cluster number so that we can compare their speeds. The table consists of 9 columns. Column 1 is the data set names. Column 2 is the number of execution of the distance measures in hierarchical clustering for each data set. Note that the number of executions of the 5 distance measures is the same in hierarchical clustering. Column 3 is the average execution times of 3 sample-matching distances. We take an average of the execution times of the 3 distances because their times are close to each other. Columns 4 and 5 are the execution times of d_{RC} and d_O respectively. Column 6 is the average number of executions of the 5 distance measures in *k*-means. We take an average because the numbers of executions are not the same for different measures in *k*-means. Columns 7 – 8 are the (average) execution times of the 5 distance measures. From columns 2 and 6 in Table 9, the number of distance executions in hierarchical clustering is significantly larger than that in *k*-means. For the same number of executions, d_T, d_G and d_R is the fastest whereas d_O is the lowest.

Table 9. Execution Times of Clustering Algorithms (seconds)

	Hierarchical Clustering in Table 6				*K*-means in Table 8			
Data Sets	# of dist. cal.	d_T [23], d_G [40], d_R	d_{RC}	d_O	Ave # of dist cal	d_T [23], d_G [40], d_R	d_{RC}	d_O
Car	309135	1.797	1.807	3.191	26727	0.047	0.061	0.114
TicTac	4322241	40.6973	50.476	93.927	48284	0.407	0.598	1.171
Liver	403225	1.931	2.267	3.150	19076	0.095	0.120	0.261
Wine	4015716	32.0587	51.842	64.347	274536	1.358	1.541	2.758
Glass	4177926	30.247	39.720	62.692	159474	0.605	0.879	1.414
Cancer	1827904	14.5407	21.276	30.871	22820	0.187	0.225	0.410
Auto	416019	2.4196	3.314	4.759	48425	0.219	0.315	0.530
Anneal	4015716	46.148	67.783	97.948	405245	14.996	17.240	31.231
Letter	4015716	287.689	394.174	512.881	859854	31.315	42.131	72.231
Iris	2208	0.016	0.016	0.018	391	0.001	0.001	0.001

In summary, the proposed measures d_O perform fairly well in most cases in our experiments, even though it is the slowest. In some cases, they even achieve the best clustering performance in terms of *Ent* and *MS*. However, the results may not reflect the whole picture of clustering quality. First, cluster entropy and *MS* scores cannot capture all aspects of clustering quality. In fact, many clustering quality may not be measurable at all. Second, the class labels may not necessarily be the "true" clustering. In fact, it is difficult to define "true" clustering. However, the experiment results suggest that the proposed measures are better than sample-matching distances in many cases.

5.4 Experiments of Pattern Summarization on Real-World Data Sets

Three pattern summarization methods including RuleCover[23], multi-level RuleCover and AreaCover are implemented. To evaluate these methods, the 10 real-world data sets are used again. To compare the performance of different methods in the 10 data sets, all summarization methods are applied to the set of pattern clusters produced by *K*-means using d_O as distance measure with prescribed cluster numbers (i.e. the cluster entropy and *MS* of these sets of patterns clusters are shown in the last column of Table 8). In other words, for each data set, all closed itemsets are clustered before summarization. The summarization methods can be applied to pattern clusters produced by other clustering algorithms (e.g. hierarchical clustering) as well as other pattern distances (e.g. d_T, d_G, d_R and d_{RC}). The pattern clusters in Table 8 are chosen because all closed itemsets are used. Hence, the speed of summarization methods can be tested against all of them. If hierarchical clustering is used, only around 2,000 itemsets are selected (see section 5.3 for details). d_O is chosen as distance measures because it performs very well in our experiments. In practice, other measures can also be used.

Table 10 gives the summarization results of RuleCover [23] with 100% minimum coverage. The table consists of 7 columns. As in Table 8, the first three columns are the data set names, number of closed itemsets used for summarization and the prescribed number of clusters respectively. Column 4 is the number of itemsets having 2 or less items (i.e. |s|-itemsets, |s| ≤2). We explicitly show this number because it is easier to interpret itemsets with fewer items. Hence, it is of interest to know how many short summary itemsets we have. We show both the total number of itemsets in all clusters and the average number of itemsets in each cluster. Column 5 is the number of all itemsets. Again, both the total number and the average number in each cluster are shown. Column 6 gives the average number of items in each itemset. Finally, column 7 gives the execution times of the summarization methods. We set 100% minimum coverage so that all summary itemsets are extracted. In practice, we often loose the minimum coverage to reduce the number of summary itemsets. Table 11 gives the summarization results of RuleCover [23] with 80% minimum coverage. From column 5 in Tables 10 and 11, the average number of all summary itemsets is significantly reduced from 133 to 28 whereas the average number of all summary itemsets per cluster is reduced from 16.50 to 4.64. This indicates that, on average, each pattern cluster is summarized by 4.64 patterns if the minimum coverage is 80%. This small set of patterns can be interpreted manually by domain experts. Note that the number of summary itemsets in Letter is much more than other data sets. If Letter data is not considered, the average numbers of all summary itemsets per cluster for Tables 10 and 11 are 14.42 and 4.50 respectively.

Another observation from Tables 10 and 11 is that most summary itemsets have only a few items. For example, in Car data, of the 56 summary itemsets, 53 itemsets have 2 or less items. This is also reflected by the fact that the average number of items is only 2.07. For the 10 data sets, the average number of items in Tables 10 and 11 are 2.11 and 2.08 respectively. This indicates that most itemsets have only 2 items. On one the hand, short itemsets (itemsets having only a few items) are desirable since it is easier to interpret than long itemsets (itemsets having many items). On the other hand, short itemsets are less informative than long itemsets. For example, 2-itemsets only describe the relationship between 2 items. In Figure 14, Δ = {9, 8}. Such summary is too general and not informative enough because both patterns 9 and 8 do not describe the first several attributes in the cluster. The problem is that pattern 9 is a short summary itemset. This only provides very general and high-level description of the cluster. Many details are missed. In practices, it is desirable to have longer summary itemsets because they are more specific and informative.

Table 10. RuleCover [23] Summarization Results (Minimum Coverage = 100%)

Data sets	Item sets #	Cluster #	Summary itemsets # ($\lvert s\rvert\le 2$)		All summary itemsets # (for all $\lvert s\rvert$)		Ave items #	Exec times (sec)
			All	Per Cluster	All	Per Cluster		
Car	557	4	53	13.25	56	14	2.07	0.001
TicTac	4024	2	37	18.5	37	18.5	2	0.028
Liver	636	2	59	29.5	59	29.5	2	0.001
Wine	18303	3	27	9	28	9.33	2.04	0.137
Glass	2045	6	48	8	50	8.33	2.06	0.001
Cancer	2854	2	39	19.5	39	19.5	2	0.036
Auto	646	5	86	17.2	90	18	2.04	0.001
Anneal	13509	5	30	6	43	8.6	2.61	0.089
Letter	21244	26	763	29.35	916	35.23	2.29	348.52
Iris	50	3	12	4	12	4	2	0.001
Total	6387	5.8	115.4	15.43	133	16.50	2.11	34.88

Table 11. RuleCover [23] Summarization Results (Minimum Coverage = 80%)

Data sets	Item sets #	Cluster #	Summary itemsets # ($\lvert s\rvert\le 2$)		All summary itemsets # (for all $\lvert s\rvert$)		Ave items #	Exec times (sec)
			All	Per Cluster	All	Per Cluster		
Car	557	4	14	3.5	15	3.75	2.07	0.001
TicTac	4024	2	16	8	16	8	2	0.018
Liver	636	2	20	10	20	10	2	0.001
Wine	18303	3	9	3	9	3	2	0.121
Glass	2045	6	17	2.83	17	2.83	2	0.001
Cancer	2854	2	7	3.5	7	3.5	2	0.031
Auto	646	5	26	5.2	27	5.4	2.04	0.001
Anneal	13509	5	7	1.4	10	2	2.6	0.083
Letter	21244	26	142	5.46	153	5.89	2.10	191.53
Iris	50	3	6	2	6	2	2	0.001
Total	6387	5.8	26.4	4.49	28	4.64	2.08	19.18

Tables 12 and 13 give the summarization results of multi-level RuleCover with 100% and 80% minimum coverages respectively. The average number of items is slightly higher than that in Tables 10 and 11. This indicates that longer itemsets are extracted after removing the trivial and general itemsets discovered from the first run of RuleCover. However, the number of items is still small. This reflects that the summary is still general and not specific. While general summary is easy to understand, longer itemsets are required to obtain more specific and informative information.

Table 12. Multi-level RuleCover Summarization Results (Minimum Coverage = 100%, Level = 2)

Data sets	Item sets #	Cluster #	Summary itemsets # (|s|≤2)		All summary itemsets # (for all |s|)		Ave items #	Exec times (sec)
			All	Per Cluster	All	Per Cluster		
Car	557	4	85	21.25	119	29.75	2.30	0.001
TicTac	4024	2	74	37	74	37	2	0.042
Liver	636	2	116	58	124	62	2.07	0.003
Wine	18303	3	54	18	60	20	2.13	0.312
Glass	2045	6	69	11.5	111	18.5	2.41	0.006
Cancer	2854	2	69	34.5	75	37.5	2.08	0.027
Auto	646	5	134	26.8	168	33.6	2.20	0.003
Anneal	13509	5	39	7.8	86	17.2	2.92	0.142
Letter	21244	26	1288	49.54	2470	95.00	2.62	681.131
Iris	50	3	17	5.67	21	7	2.19	0.001
Total	6387	5.8	194.5	27.0	330.8	35.8	2.29	68.167

Table 13. Multi-level RuleCover Summarization Results (Minimum Coverage = 80%, Level = 2)

Data sets	Item sets #	Cluster #	Summary itemsets # (|s|≤2)		All summary itemsets # (for all |s|)		Ave items #	Exec times (sec)
			All	Per Cluster	All	Per Cluster		
Car	557	4	32	8	36	9	2.11	0.001
TicTac	4024	2	32	16	32	16	2	0.046
Liver	636	2	43	21.5	44	22	2.02	0.001
Wine	18303	3	17	5.67	18	6	2.06	0.231
Glass	2045	6	36	6	44	7.33	2.21	0.004
Cancer	2854	2	16	8	16	8	2	0.019
Auto	646	5	58	11.6	68	13.6	2.15	0.001
Anneal	13509	5	15	3	22	4.4	2.59	0.119
Letter	21244	26	308	11.85	344	13.23	2.15	358.413
Iris	50	3	11	3.67	12	4	2.08	0.001
Average	6387	5.8	56.8	9.53	63.6	10.36	2.14	35.884

Tables 14 and 15 give the summarization results of AreaCover with 100% and 60% minimum coverage respectively. The average number of items is higher than that in Tables 10 – 13. It indicates that AreaCover extracts longer itemsets than RuleCover [23] and multi-level RuleCover. It can be explained by the fact that AreaCover considers matched attributes neglected by RuleCover. Hence, it tends to extract summary itemsets from all attributes. In RuleCover, it stops extracting summary itemsets if a certain percentage of samples (as specified by minimum coverage) are covered by the itemsets in RuleCover. However, in AreaCover, the algorithm will continue to extract summary itemsets as long as the items in other attributes have not been covered. Hence, the summary itemsets will cover most attributes of the data cluster and tends to produce longer itemsets. Such desirable properties are not guaranteed by RuleCover or multi-level RuleCover.

Moreover, in AreaCover, the summary patterns tend to cover different attributes in the cluster. As a result, most attributes in the clusters are described. In contrast, in RuleCover, it is possible that only a small subset of attributes is described. Such summary is not informative because it does not provide information about some of the attributes. Another advantage of AreaCover is that each pattern tends to describe different attribute values when compared with RuleCover and multi-level RuleCover. It can be explained by the fact that once an attribute value is covered by the patterns in Δ, pattern covering it do not contribute additional coverage. Hence, the description given by the summary patterns produced by AreaCover is less redundant than those produced by RuleCover and multi-level RuleCover.

Table 14. AreaCover Summarization Results (Minimum Coverage = 100%)

Data sets	Item sets #	Cluster #	Summary itemsets # (|s|≤2)		All summary itemsets # (for all |s|)		Ave items #	Exec times (sec)
			All	Per Cluster	All	Per Cluster		
Car	557	4	91	22.75	178	44.5	2.68	0.024
TicTac	4024	2	122	61	228	114	2.48	0.073
Liver	636	2	171	85.5	194	97	2.13	0.021
Wine	18303	3	71	23.67	191	63.67	3.32	2.875
Glass	2045	6	73	12.17	255	42.5	3.14	0.061
Cancer	2854	2	122	61	151	75.5	2.34	0.321
Auto	646	5	154	30.8	243	48.6	2.49	0.022
Anneal	13509	5	25	5	271	54.2	4.14	3.121
Letter	21244	26	1999	76.89	3053	117.42	2.52	739.781
Iris	50	3	14	4.67	24	8	2.5	0.001
Total	6387	5.8	284.2	38.35	478.8	66.54	2.77	74.63

Table 15. AreaCover Summarization Results (Minimum Coverage = 60%)

Data sets	Item sets #	Cluster #	Summary itemsets # (|s|≤2)		All summary itemsets # (for all |s|)		Ave items #	Exec times (sec)
			All	Per Cluster	All	Per Cluster		
Car	557	4	22	5.5	60	15	2.92	0.021
TicTac	4024	2	38	19	50	25	2.24	0.061
Liver	636	2	46	23	47	23.5	2.02	0.012
Wine	18303	3	26	8.67	62	20.67	3.21	2.691
Glass	2045	6	20	3.33	75	12.5	3.35	0.058
Cancer	2854	2	11	5.5	22	11	3.05	0.213
Auto	646	5	65	13	92	18.4	2.49	0.019
Anneal	13509	5	16	3.2	258	51.6	4.23	3.211
Letter	21244	26	1412	54.31	2053	78.96	2.52	683.636
Iris	50	3	3	1	7	2.33	2.57	0.001
Total	6387	5.8	165.9	13.65	272.6	25.90	2.86	68.992

Chapter 6
Conclusion and Future Research

The work described in this dissertation was motivated by the recognition of: (1) increasingly large amounts of raw data and discovered patterns which require the facilitation of an automatic pattern post-analysis system for a better understanding of the problem domain reflected by the data; (2) the pressing need to develop intelligent system which are able to support knowledge discovery and decision making from the huge volume of discovered patterns; (3) the potential applications of pattern post-analysis in both scientific and business worlds; (4) the inability of most available post-analysis methods to cope with large volume of patterns which are overlapping and entangling with each other; and (5) the application limitation of most existing systems which solve only a particular problem and therefore, not general enough to render an integrated post-analysis framework for real-world applications.

The research presented in this thesis is an integrated and flexible framework for pattern post-analysis. It proposes a new system of pattern post-analysis including pattern pruning, pattern clustering and pattern summarization to support effective analysis and interpretation of the discovered patterns. A dual space approach is introduced in which the explicit correspondence between patterns and data is maintained throughout the entire framework. Once this relationship is established and made explicit, the system is able to meaningfully and accurately measure the distances between patterns via the differences measured from their induced data. While the patterns provide human-friendly expression to describe the statistical nature of the data, the data associated with the patterns provide a basis to analyze the patterns. Furthermore, the sample-attribute matching distances and the entropy-based distances in support of the dual space approach are developed. These distances are naturally extended from the existing sample-matching distances.

In pattern pruning, using the concept of sample-attribute matching, a generalized itemset pruning method is developed which generalize the classical closed [8], [9], [29] – [31] and maximal itemset pruning [32] – [35]. The proposed method allows the users to control the tradeoff between the number of patterns being pruned and the amount of information loss after pruning. It furnishes a middle alternative, as closed itemset pruning removes not too many patterns while maximal itemset pruning loses too much information. For pattern clustering, a simultaneous pattern and data clustering method is developed which is able to cluster patterns as well as their associated data. The resulting dual clusters maintain an explicit one-to-one correspondence between patterns and data. Such explicit correspondence enables further analysis of individual dual clusters. The dual clustering process is implemented in two common clustering algorithms: *k*-mean clustering and hierarchical clustering. Both algorithms, using the entropy-based distances developed, are able to capture important factors that are overlooked by sample-matching distances. In our experiments on both synthetic and real data, it is found that the entropy-based distance performs the best, but it is the slowest. Finally, the concept of sample-attribute matching is used to develop an AreaCover summarization method. AreaCover method does not suffer from the problem of being sensitive to trivial large patterns as RuleCover does.

Hence, it can produce a concise yet detailed enough summary for each pattern cluster. The resulting summary can be easily interpreted by a human user.

6.1 Summary of Contribution

A good way to summarize the contributions of this research is to evaluate its outcomes against its objectives as stated at the outset of this thesis. The following statements are an account corresponding to the points outlined in section 1.2.2.

1. The current research brings forth a general and versatile pattern post-analysis system [93] – [95], [126] and [127].

   In the system, redundant patterns are first pruned by generalized itemset pruning. To meet the needs of the complex real-world problems, the amount of information loss and the number of pattern being pruned can now be controlled by the users. Then, the retaining patterns and their associated data are simultaneously clustered using either sample-attribute-matching distances or entropy-based distances. To support the human interpretation, AreaCover is developed to generate a concise and descriptive summary for each pattern cluster. The summary produced can be manually inspected by a human user. The system is general enough to apply to various real-world problems with different types of patterns including directional or non-directional patterns. By allowing users to control the tradeoff between speed and quality, the system is flexible enough to meet the complex needs of different real-world problems. Two fruitful contributions of the proposed works are: 1) the dual pattern-data relationship that furnishes robust and effective distance measures between patterns through their associated data, and 2) the applications of the dual relationship to pattern pruning, simultaneous pattern and data clustering and pattern summarization. The former provides the strong base for effective analysis of discovered patterns while the latter renders an integrated, flexible and automatic system to handle the huge volume of discovered patterns.

2. An effective simultaneous pattern and data clustering algorithm is developed. The proposed pattern-data relationship furnishes the new sample-attribute-matching distances and the entropy-based distances which are able to capture important factors overlooked by existing sample-matching distances for more accurate distance measures.

   By taking into consideration of both the matched samples and matched attributes of patterns, the sample-attribute-matching distances capture the attribute aspect of information that is overlooked by sample-matching distances. By further considering the data variation within the data associated with the pattern clusters, entropy-based distances takes into account the important factor reflecting the degree of variations/uncertainty in the data during the pattern clustering process. Such factors account for the subtle yet crucial differences between pattern/data clusters. By considering these additional factors, the proposed distance measures are more accurate and robust than the sample-matching distances. The dual clustering algorithm, supported by the proposed distance measures, is able to achieve effective pattern and data clustering.

3. A new type of generalized itemsets is proposed based on the concept of sample-attribute matching.

From the standpoint of information loss, the generalized itemsets generalize the classical closed itemsets and maximal itemsets by taking them as two special cases. While closed itemset pruning does not lose any information, maximal itemset pruning allows maximal amount of information loss. A nice property of generalized itemset pruning is that it allows the user to control the tradeoff between information loss and the amount of patterns retained. It thus furnishes a good alternative when closed itemset pruning produces too many patterns while maximal itemset pruning loses too much information.

4. The multi-level RuleCover and AreaCover summarization methods proposed in this thesis render for each pattern cluster a concise and representative summary which can be interpreted manually by a human user.

By applying the well-known RuleCover pruning algorithm to select a few patterns in each pattern cluster, we propose an effective algorithm for pattern summarization. However, as observed, RuleCover is prone to render trivial patterns which cover large number of samples. Hence, the summarization results are less useful or interesting. To address this problem, a pattern summarization algorithm based on multi-level RuleCover is developed in this dissertation work. However, although the multi-level RuleCover algorithm yield better data coverage for pattern clusters than a single level RuleCover algorithm, the results are still not satisfactory. Using the concept of sample-attribute matching, AreaCover is developed in this thesis to further improve pattern summarization. By considering the attribute aspect as well as the sample aspect in the cluster, AreaCover has several advantages over the other methods. First, it tends to obtain longer and more descriptive patterns than those selected by RuleCover and multi-level RuleCover. Second, the patterns produced by AreaCover tend to take in different attribute values – richer in the pattern description. Hence, the descriptive pattern obtained by AreaCover is less redundant and more informative than those obtained by RuleCover and multi-level RuleCover. Third, AreaCover tends to cover all attributes in the clusters. Hence, the description of each attribute will not be missed in the summarization. Hence, AreaCover is able to provide concise yet descriptive summary of each complex pattern cluster.

5. The information loss after pruning and the data not covered in the summarization can both be measured and controlled by the user.

The information loss after generalized itemset pruning can now be measured and controlled so that the user can decide on the tradeoff between the amount of information loss and the number of patterns retained. Similarly, the amount of data covered by RuleCover, multi-level RuleCover and AreaCover can be quantified and controlled by the user. Hence, the user has a sense of how representative the patterns are in the cover obtained. Such ability is important since information loss is inevitable during pattern pruning and summarization.

6. Experiments on synthetic and real-world data sets indicated that the system is superior to many existing methods in term of performance of pruning, accuracy of clustering and quality of summarization.

For pattern pruning, empirical tests on real-world data sets indicated that the generalized itemset pruning provides a good alternative to closed and maximal itemset pruning. It allows the user to control the tradeoff of information loss and the amount of the patterns retained. For pattern clustering, empirical tests on synthetic and real-world data sets indicated that the proposed sample-attribute-matching and entropy-based distance measures based on the dual pattern-data relationship are superior to existing sample-matching distances. Finally, for pattern summarization, experimental results showed that AreaCover produces more concise and informative description than other methods and is not prone to produce large trivial patterns covering a large number of samples

7. A prototype of the pattern post-analysis software system has been implemented. The potential applications of such system are numerous.

The prototyped system can analyze patterns discovered by various pattern mining methods including both association rule mining [4] – [9] and pattern discovery [10] – [14]. It can automatically prune redundant patterns, cluster similar patterns into clusters and/or generate concise and descriptive summary for each pattern cluster at the request of the user. This system works well especially in situations where 1) no a *priori* knowledge is available, 2) the decision maker needs more information than that of a single pattern, and 3) transparent results are important. Some current applications of the system are listed below:

- Analyzing patterns discovered from the operational data from an oil sand plant for improved performance, safety (including root cause analysis), environmental management and increased return on investment.
- Analyzing student behavioral patterns discovered from academic record data from a school board for studying students' learning behaviors and the efficiency of computer-based tests such as Canadian Cognitive Abilities Test (CCAT), Canadian Achievement Test (CAT) and EQAO (Education Quality and Accountability Office) tests.
- Analyzing patterns discovered from legal documents database from a law firm for automatic, effective and intelligent organization and management of legal documents.
- Analyzing patterns discovered from biological data such as DNA sequences and protein sequences for discovering biologically meaningful patterns and structures.

Some other possible applications of the system are listed below:

- Analyzing patterns from large databases such as stock market records, connection records of telecommunication companies and basket data of department stores for business planning.
- Analyzing patterns for quality control, diagnosis, and decision support systems; and
- Organizing, encoding and retrieving information from various data sets or databases.

In addition to these contributions, there are some other points worth mentioning:

- The proposed methods can be applied to a wide variety of patterns including directional and non-directional patterns. In this thesis, due to the popularity of frequent itemsets, the proposed methods are applied to them to demonstrate its capability. However, in [93] – [95], it has also been applied to event association patterns generated by pattern discovery [10] – [14]. Other possible applicable patterns include correlation rules.

- In practice, depending on the needs of the problems, the pattern pruning, clustering and summarization methods in the proposed system can be used either together or separately. For example, pattern clustering and summarization can be performed without pruning. Another example is that summarization can be applied to all patterns to generate a high level summary of all patterns.

6.2 Suggested Future Research

Several interesting problems related to this research are still open for future investigation. The following is a list of some possible directions presented as the conclusion of this thesis.

1. Subjective pattern post-analysis using the proposed distance measures

   Although this thesis focuses on the objective pattern post-analysis approaches, the proposed dual pattern-data relationship and the developed distance measures can also be adapted and used in subjective post-analysis approaches. For example, objective pattern pruning either prune a limited number of patterns or suffer from great information loss due to over pruning. To prune more patterns without losing the interesting ones, the use of domain knowledge or specification of the users' expectation, if applicable, are significant. Existing subjective pruning algorithms focuses on the use of templates, constraints, or attribute hierarchy to specify the domain knowledge of the users (see section 2.3.2). However, it is difficult and time-consuming for the users to specify such templates, especially for complex real-world problems. One possible solution for subjective pattern clustering or pruning related to this dissertation work is to use a semi-supervised learning approach [2] for subjective pattern clustering or pattern pruning. Semi-supervised learning has been widely used in data analysis. Effective semi-supervised clustering algorithms such as those in [2], [119] – [123] exist. With the proposed distance measures, a semi-supervised pattern clustering algorithm could be developed. The users can tell the algorithms whether certain patterns are interesting or uninteresting by just labeling them as "interesting" or "uninteresting" respectively. Then, patterns similar to uninteresting patterns are pruned, retaining only those patterns that are similar to the interesting patterns. The advantage of such approach is that it tolerates incomplete knowledge from the user. It makes the user much comfortable as he/she does not need to label all patterns. Instead, after the user labels a small portion of interesting and/or uninteresting patterns (e.g. 10% uninteresting patterns and 5% interesting patterns), the algorithm will fill in the missing labels of the rest (i.e. the 85% of the patterns). Moreover, the labeling can be aided by objective approaches. For example, the user can label the summary patterns produced by pattern summarization and the algorithm could determine the labels of the rest. Since summary patterns are representative to all other patterns, they are good candidates for labeling. In practice, we find that it is easier for a user to criticize than to construct (to discriminate than to ascertain). Hence, users prefer to label patterns as "uninteresting". Nevertheless, in theory, this approach can incorporate both positive and negative comments from the users.

2. Pattern visualization using the proposed distance measures

The proposed distance measures can be used to support visualizing patterns and pattern clusters. One possible approach is to use multi-dimensional scaling (MDL) [1], [40] to convert the distances between patterns into an embedded vector space such that the distance information is preserved. Then self-organizing map can be used to visualize the patterns or pattern clusters in a two-dimensional grid. The problem is that we have to reduce the dimensions to two or three for visualization purposes since direct visualization ability of human eyes is limited to only three dimensions. If the patterns themselves by nature are not separable into two or three dimensions, multi-dimensional scaling will produce non-separable set of patterns. The quality of the scaling can be reflected by the stress factor [1], [40]. More research is needed to investigate how to visualize patterns that are not separable into two or three dimensional grids.

3. Clustering of unseen samples to existing pattern clusters

   Once pattern clusters are formed, each of them is well described by its summary patterns. If the pattern clusters can be interpreted by domain experts, they become useful knowledge. When new, unseen samples come in, it is highly desirable to group them to the existing well-understood pattern clusters to investigate the properties of the new samples. To achieve this, a similarity measure between a sample and a pattern cluster has to be developed. One possible similarity measure is the number of patterns in a pattern cluster contained in a new sample. Intuitively, if there are many patterns in a pattern cluster contained in a new sample, the pattern cluster is similar to the sample. More sophisticated methods to relate new samples to pattern clusters include weighting of the sample-attribute-matching region of each pattern.

4. Discretization of continuous attributes and mixed-mode data analysis

   Discussion in the current research is limited to discrete attributes or discretized continuous attributes. The choice of discretization algorithms for continuous attributes is therefore very important for the performance of both the pattern mining and post-analysis processes. Although the earlier works [124], [125] yielded some insights into this problem, further investigations is definitely necessary. To analyze mixed-mode data, we either discretize the continuous attribute and then apply discrete analytical algorithms, or to analyze mixed-mode data directly without discretization. The tradeoff of the two choices is worth studying.

5. More efficient implementation of the distance measures

   It is possible to integrate the pattern post-analysis system with the pattern mining system so that redundant computation can be eliminated. For example, the counting of the number of samples containing a pattern (i.e. supports) is usually available in the pattern mining process. Such information can be used to calculate the distance measures. Another method is to develop an effective data structure to store the counts for all possible attribute value pairs and their intersections or unions

There are many other valuable research possibilities that are not mentioned here. Due to the challenging topics identified and the tremendous potential applications, pattern post-analysis will continue to gain more and more attention in both the scientific and the industrial worlds.

Appendices

Appendix A
Mathematical Relationship between Association Rule Mining and Pattern Discovery

Association rule mining and pattern discovery are two highly related problems. In association rule mining, an itemset x^s is said to be frequent if its observed frequency of occurrences o_{x^s} is greater than a user specified threshold *c*. That is,

$$o_{x^s} \geq c \tag{23}$$

In pattern discovery, recall from (5) and (6) that an itemset x^s is said to be associated if

$$o_{x^s} \geq \left|d_{x^s}\right|\sqrt{e_{x^s} v_{x^s}} + e_{x^s} \tag{24}$$

Comparing (23) and (24), association rule mining uses a constant threshold *c* to detect frequent itemsets whereas pattern discovery uses an adaptive threshold $\left|d_{x^s}\right|\sqrt{e_{x^s} v_{x^s}} + e_{x^s}$ for each itemset. Since d_{x^s} is a constant and, from (3) and (7), e_{x^s} and v_{x^s} are dependent on the term $\prod_{x_i \in x^s} P(x_i)$ and a constant *M*, $\left|d_{x^s}\right|\sqrt{e_{x^s} v_{x^s}} + e_{x^s}$ becomes a same constant for all itemsets x^s if $\prod_{x_i \in x^s} P(x_i)$ are the same for all x^s. More precisely, the criterion of detecting significant patterns in pattern discovery is equivalent to the criterion of detecting frequent itemsets in association rule mining if

$$\prod_{x_i \in x^s} P(x_i) = \text{constant}, \forall x^s \tag{25}$$

Hence, pattern discovery uses different thresholds for different itemsets x^s, while association rule mining uses a fixed threshold for all itemsets. However, the cost of such customization is the lack of the important Apriori property. Hence, pattern discovery is slower than frequent itemset mining despite the use of several effective heuristics based on the statistical properties of contingency tables [10] – [14].

Appendix B

Mathematical Relationship between Chi-squared Test and Residual Test

The chi-squared test for correlation has been widely used in various methods. It may be of interest to derive the relation between chi-squared statistics and the residuals in (5) and (6). The chi-squared statistics has the form of:

$$\chi^2 = \sum_{x^s} \frac{\left(o_{x^s} - e_{x^s}\right)^2}{e_{x^s}} = \sum_{x^s} \left(\frac{o_{x^s} - e_{x^s}}{\sqrt{e_{x^s}}} \right)^2 = \sum_{x^s} z_{x^s}^2 \qquad (26)$$

where z_{x^s} is the standardized residual in (5). Hence, standardized residual is the square root of values of the individual cells of χ^2. Since χ^2 distribution is the sum of squared standard normal distribution, z_{x^s} is normally distributed with zero mean and unit variance. To ensure z_{x^s} have unit variance z_{x^s} is normalized by its estimated variance v_{x^s} in (6) to obtain the adjusted residual. Hence, while chi-squared test is a test for correlation among attributes, the residual test is a test for correlation among the values of the attributes (i.e. items).

Appendix C

Equivalent Clustering Results Produced by Clustering Closed Itemsets and Clustering All Itemsets

If an itemset $x_i^{s_i}$ is the closed itemset of $x_j^{s_j}$, then, by the definition of closed itemsets (see section 4.1.1), $x_j^{s_j} \supset x_i^{s_i}$ and $|m(i)| = |m(j)|$. Hence, $m(i) = m(j)$. By the definition of pattern-induced data clusters (see (11)), the data cluster induced by $x_i^{s_i}$ completely contains the data cluster induced by $x_j^{s_j}$. That is, $I(i) \supset I(j)$. Hence, the measures d_T, d_G, d_R, d_{RC} and d_O are all equal to 0. Since the distances between the 2 itemsets is 0, they must be first merged at the beginning of clustering. Moreover, the resulting merged data cluster will be the same as $I(i)$. That is, $I(i, j) = I(i)$ since $s_i \supset s_j$ and $m(i) = m(j)$ (see (12)). Thus, all itemsets will be merged to their corresponding closed itemsets at the beginning of clustering and the resulting merged data cluster is just the data cluster induced by the closed itemsets. From then on, the clustering algorithm will cluster the data cluster of closed itemsets. Hence, clustering closed itemsets will produce equivalent results as clustering all frequent itemsets.

Appendix D

Generalization Relation between the Generalized Itemsets and the Closed and the Maximal Itemsets

Consider two itemsets $x_i^{s_i}$ and $x_j^{s_j}$ where $s_j \supset s_i$. If $loss(i, j) = 0$, by (14), $r_i = 0$. Hence, $|m(i)| = |m(j)|$. Hence, by definition (see section 4.1.1), $x_j^{s_j}$ is a closed itemset of $x_i^{s_i}$. If $loss(i, j) = 1$, by (19), $r_{ij} = 0$, indicating that it is not necessary for $x_i^{s_i}$ and $x_j^{s_j}$ to share any samples. Hence, by definition, $x_j^{s_j}$ is a maximal itemset of $x_i^{s_i}$ since $s_j \supset s_i$.

Bibliography

[1] J. Han and M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2001.

[2] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, Wiley, 2000.

[3] J. Ghosh. *Handbook of Data Mining*, Lawrence Erlbaum Assoc., 2003.

[4] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” *Proc. of the 20th Int. Conf. on Very Large Data Bases*, Santiago, Chile, pp. 487–499, 1994.

[5] J. Han, J. Pei, Y. Yin, “Mining Frequent Patterns without Candidate Generation,” *Proc of ACM SIGMOD Intl. Conference on Management of Data*, 2000

[6] J.Hipp, U.Gűntzer, and G.Nakhaeizadeh, “Algorithms for Association Rule Mining – General Survey and Comparsion,” *ACM SIGKDD Explorations Newsletter*, vol.2, no.1, pp.58 – 64,2000.

[7] R. Aggarwal, T. Imielinski, A. Swami, “Mining Association Rules Between Sets of Items in Large Databases,” *Proc. ACM SIGMOD Conf. Management of Data (SIGMOD’93)*, pp. 207-216, 1993.

[8] N. Pasquier, Y. Bastide, R. Taouil and L. Lakhal, “Efficient Mining of Association Rules Using Closed Itemset Lattices,” *Information Systems*, vol. 24, no. 1, pp. 25-46, 1999.

[9] M. J. Zaki, C. J. Hsiao, “Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure, “ *IEEE Trans. on Knowledge and Data Engineering*, vol.17, no.14, pp.462–477,2005

[10] A.K.C. Wong and Y. Wang, “High Order Pattern Discovery from Discrete-Valued Data,” *IEEE Trans. on Knowledge and Data Eng.*, vol. 9, no. 6, pp. 877-893, 1997.

[11] A.K.C. Wong and Y. Wang, “Pattern Discovery: A Data Driven Approach to Decision Support,” *IEEE Trans. on Syst., Man, Cybern. – Part C*, vol. 33, no. 1, pp. 114-124, 2003.

[12] T. Chau, & A.K.C. Wong, “Pattern Discovery by Residual Analysis and Recursive Partitioning,” *IEEE Trans. on Knowledge and Data Eng.*, vol. 11, no. 6, pp. 833-852, 1999.

[13] Andrew K. C. Wong, Wang Yang and Gary C. L. Li, “Pattern Discovery as Event Association,“ to appear in *Encyclopedia of Data Warehousing and Mining - 2nd Edition*

[14] Y. Wang and A. K. C. Wong, “From Association to Classification: Inference Using Weight of Evidence,” *IEEE Trans. on Knowledge and Data Eng*., vol. 15, no. 3, pp. 914-925, 2003.

[15] S. Brin, R. Motwani and R. Silverstein, “Beyond Market Basket: Generalizing Association Rules to Correlations,” *Proc. of ACM SIGMOD Conf. Management of Data (SIGMOD’97)*, pp. 265-276, 1997.

[16] K. M. Ahmed, N. M. El-Makky and Y. Taha, “A note on “Beyond Market Baskets: Generalizing Association Rules to Correlations”,” *ACM SIGKDD Explorations*, vol. 1, no. 2, pp. 46 - 48, 2000.

[17] G. Dong, J. Li, “Efficient Mining of Emerging Patterns: Discovering Trends and Differences, “*Proc. 5th Int. Conf. Knowledge Discovery and Data Mining (KDD’99)*, pp. 43-52, 1999.

[18] H. Xiong, “Hyperclique pattern discovery, “Data Mining and Knowledge Discovery, vol. 13, no. 2, pp. 219 - 242, 2006.

[19] P.M. Murph and D.W. Aha, “UCI Repository of Machine Learning Databases,” Dept. of Information and Computer Science, Univ. of California, Irvine, 1987.

[20] B. Liu, W. Hsu, and Y. Ma, “Pruning and Summarizing the Discovered Associations,” *Proc. 5th Int. Conf. Knowledge Discovery and Data Mining (KDD’99)*, pp. 125-134. 1999.

[21] B. Liu, W. Hsu, and Y. Ma, “Summarizing the discovered associations using direction setting rules, “ Technical Report, SoC, 1999.

[22] B. Liu, W. Hsu, and Y. Ma, “Integrating classification and association rule mining, “*Proc. 4th Int. Conf. Knowledge Discovery and Data Mining (KDD’98)*, 1998.

[23] H. Toivonen, M. Klemetinen, P. Ronkaninen, K. Hatonen, and H. Mannila, “Pruning and Grouping Discovered Association Rules,” *Minet Workshop on Statistics, Machine Learning, and Discovery in Databases*, pp. 47-52, 1995.

[24] S. Chawla and J. Davis, *On Local Pruning of Association Rules Using Directed Hypergraphs*, Technical Report 537, School of Information Technologies, University of Sydney, 2003.

[25] T. Brijs, K. Vanhoof and G. Wets, “Reducing redundancy in characteristic rule discovery by using integer programming techniques,” *Intelligent Data Analysis*, vol. 4, pp. 229-240, 2000.

[26] P. Gago and C. Bento, “A metric for selection of the most promising rules,” *Proc. 2nd European Symposium, PKDD98, Lecture Notes in Artificial Intelligence 1510*, pp. 19–27, 1998.

[27] M. Kryszkiewicz, “Representative association rules and Minimum condition maximum consequence association rules,” *Proc. Principles of Data Mining and Knowledge Discovery Conference (PKDD’98)*, Nantes, pp. 361–369, 1998.

[28] R.J. Bayardo, Jr., “Brute-force mining of high-confidence classification rules,” *Proc. 3rd Int. Conf. of Knowledge Discovery and Data Mining*, The AAAI Press, pp. 123–126, 1997.

[29] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal, “Mining frequent patterns with counting inference,” *SIGKDD Explorations*, vol. 2, no. 2, 2000.

[30] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, “Discovering frequent closed itemsets for association rules,” *In Proc. of 7th Int. Conf. on Database Theory*, 1999.

[31] J. Pei, J. Han, and R. Mao, “Closet: An efficient algorithm for mining frequent closed itemsets,” *In SIGMOD Int. Workshop on Data Mining and Knowledge Discovery*, 2000.

[32] R. Agrawal, C. Aggarwal, and V. Prasad, “Depth First Generation of Long Patterns,” *Proc. of 7th Int. Conf. on Knowledge Discovery and Data Mining*, August 2000.

[33] R. J. Bayardo, “Efficiently mining long patterns from databases,” *In Proc. of ACM SIGMOD Conf. Management of Data*, June 1998.

[34] D. Burdick, M. Calimlim, and J. Gehrke, “MAFIA: a maximal frequent itemset algorithm for transactional databases,” *In Proc. of Int Conf. on Data Engineering*. April 2001.

[35] K. Gouda and M. J. Zaki, “Efficiently mining maximal frequent itemsets,” *In Proc. of 1st IEEE Int. Conf. on Data Mining*, November 2001.

[36] R. Bayardo, R. Agrawal, and D. Gunopulos, “Constraint-Based Rule Mining in Large, Dense Databases,” *In Proc. 15th Int. Conf. Data Engineering*, pp. 188-197, 1999.

[37] Y. Aumann and Y. Lindell, “A statistical theory for quantitative association rules, “ *Knowledge Discovery and Data Mining*, pp. 261-270, 1999.

[38] B. Padmanabhan and A. Tuzhilin, “ Small is beautiful: discovering the minimal set of unexpected patterns,” Proc. 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD00), pp. 54-63, N. Y., 2000.

[39] S. Jaroszewicz, D. A. Simovici, “Pruning Redundant Association Rules Using Maximum Entropy Principle,” *Proc. Advances in Knowledge Discovery and Data Mining, 6th Pacific-Asia Conference (PAKDD'02)*, 2002

[40] G. K. Gupta, A. Strehi and J. Ghosh, “Distance Based Clustering of Association Rules,” *Proc. Int. Conf. Artificial Neural Networks in Engineering*, vol. 9, pp. 759–764, 1999.

[41] B. Everitt. Cluster Analysis, 2nd Edition. Halsted Press, 1980.

[42] B. Lent, A. Swami and J. Widom, “Clustering Association Rules,” *Proc. Int. Conf. Data Engineering (ICDE'97)*, pp. 220--231, Birmingham, England, 1997

[43] K. Wang, S.H.W. Tau and B. Liu, ”Interestingness based interval merger for numeric association rules”, *Proc. 4th Int. Conf. Knowledge Discovery and Data Mining (KDD’98)*, 1998.

[44] A. An, S. Khan and X. Huang, “Objective and Subjective Algorithms for Grouping Association Rules”, Proc. Third IEEE International Conference on Data Mining (ICDM'03), pp. 477 - 480, 2003.

[45] J. S. Gero and R. Stanton. *Artificial Intelligence Development and Applications. North Holland*, 1987.

[46] W. A. Woods. Knowledge representation: What's important about it? In N. Cercone and G. McCalla, editors, The Knowledge Frontier: Essays in the Representation of Knowledge, pp. 44-79. Springer-Verlag, New York, 1987.

[47] R. S. Michalski and P. Stepp. Automated construction of classifications: Conceptual clustering versus numerical taxonomy. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 5, no. 4, pp. 396 - 409, 1983.

[48] J. R. Quinlan. Induction of decision trees. Machine Learning, vol. 1, no. 1, pp. 81 - 106, 1986.

[49] C. J. Thornton. Technique in Computational Learning. Chapman & Hall, London, UK, 1992.

[50] P. Smyth and R. M. Goodman. Information theoretic approach to rule induction from database. *IEEE Trans. on Knowledge and Data Engineering*, vol. 4, no. 4, pp. 301-316, 1992.

[51] R. S. Michalski, "A theory and methodology of inductive learning," *Artificial Intelligence*, vol. 20, no. 2, pp. 111-161, 1983.

[52] H. A. Simon, "Why should machine learn?" In J. G. Michalski, R. S. Carbonell and T. M. Mitschell, editors, Machine Learning: An Artificial Intelligence Approach, vol. 1, ch. 2, pp. 25 - 38, Tioga Publishing Co., 1983.

[53] J. R. Quinlan, "Simplifying decision trees, " International Journal of Man-Machine Studies, vol. 27, no. 2, pp. 221-234, 1987.

[54] J. G. Carbonell and P. Langley. Learning, machine. In C. S. Shapiro, editor, Encyclopedia of Artificial Intelligence, vol. 1, pp. 464-488, Wiley, 1987.

[55] I. Bratko, "Machine Learning in Artificial Intelligence, " Artificial Intelligence in Engineering, vol. 8, no. 3, pp. 159-164, 1993.

[56] T. M. Michell, “Version spaces: A candidate elimination approach to rule learning, “ *Proc. 5th Int. Joint Conf. on Artificial Intelligence*, pp. 305-316, 1977.

[57] R. Michalski and R. Chilauski, “Knowledge acquisition by encoding expert rules versus computer induction from examples: A case study involving soybean pathology, “ International Journal of Man-Machine Studies, vol. 12, pp. 63-87, 1980.

[58] F. Hayes-Roth and J. McDermott, “An interface matching technique for inducting abstractions, “Communications of the ACM, vol. 21, no. 5, pp. 401-410, 1978.

[59] J. R. Quinlan, “Discovering rules by induction from large collections of examples” In D. Michie, editor, *Expert Systems in the Micro-Electronic Age*, pp. 168-210. Edinberg University Press, 1979

[60] T. M. Mitchell, “Generalization as search, “Artificial Intelligence, vol. 18, no. 2, pp. 203-226, 1982.

[61] K. C. C. Chan, Induction Learning in the Presence of Uncertainty. PhD thesis, Department of Systems Design, University of Waterloo, 1989.

[62] L. Breiman, J. H. Freidman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Wadsworth Belmont, 1984.

[63] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, 1993.

[64] R. Agrawal, T. Imielinski, and A. Swami, “Database mining: A performance perspective, “ IEEE Trans. on Knowledge and Data Engineering, vol. 5, no. 6, pp. 914-925, 1993.

[65] D. H. Fisher, “Knowledge acquisition via incremental conceptual clustering, “ Machine Learning, vol. 2, no. 2, pp. 139-172, 1987.

[66] D. Heckerman, “Bayesian networks for knowledge discovery, “ In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, ch 11, pp. 273-305, AAAI Press/MIT Press, 1996.

[67] W. J. Frawley, G. Piatesky-Shapiro, and C. J. Matheus, “Knowledge discovery in databases: An overview, “ In G. Piatetsky-Shapiro and W. J. Frawley, editors, Knowledge Discovery in Databases. AAAI/MIT Press, 1991.

[68] J. M. Zytkow and J. Baker, “Interactive mining of regularities in databases, “ In G. Piatetsky-Shapiro and W. J. Frawley, editors, Knowledge Discovery in Databases, ch. 2, pp. 31-53. AAAI/MIT Press, 1991.

[69] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery: An overview, “ In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, ch. 1, pp. 1-34. AAAI/MIT Press, 1996.

[70] I. Brakto and I. Kononenko, “Learning diagnostic rules from incomplete and noisy data, “ In B. Phelps, editor, *Interactions in Artificial Intelligence and Statistical Methods*, pp. 142-153. Technical Aldershot, 1987.

[71] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazarus, “Inductive knowledge acquisition: A case study, “ In J. R. Quinlan, editor, *Applications of Expert Systems*, pp. 157-173. Addison-Wesley, Australia, 1987.

[72] B. Cestnik, I. Kononenko, and I. Bratko, “ASSISTANT 86: A knowledge elicitation tool for sophisticated users,” In I. Bratko and N. Larvac, editors, *Progress in Machine Learning: Proc. of EWSL 87*, pp. 31-45, 1987.

[73] T. Niblett, “Constructing decision trees in noisy domains, “ In I. Bratko and N. Larvac, editors, *Progress in Machine Learning: Proc. of EWSL 87*, pp. 67-78, 1987.

[74] S. N. Gelfand, C. S. Ravishankar, and E. J. Delp, “An interactive growing and pruning algorithm for classification tree design, “ *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, pp. 163-174, 1991.

[75] R. M. Goodman and P. Smyth, “Decision tree design from a communication theory standpoint, “ IEEE Trans. on Information Theory, vol. 34, no. 5, pp. 979-994, 1988.

[76] Q. R. Wang, C. Y. Suen, “Large tree classifier with heuristic search and global training, “ IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 9, no. 1, pp. 91-102, 1987.

[77] A. R. Safavian and D. Landgrede, “A survey of decision tree classifier methodology, “ IEEE Trans. on Systems, Man, and Cybernetics, vol. 21, no. 3, pp. 660-674, 1991.

[78] D. F. Fisher, “Conceptual clustering, learning from examples, and inference, “ *Proc. 4th Int. Workshop on Machine Learning*, pp. 38-49, 1987.

[79] S. B. Thrun, The MONK’s problems: A performance comparison of different learning algorithms. Technical Report CS-CMU-91-197, Carnegie Mellon University, 1991.

[80] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac, The AQ15 inductive learning system: An overview and experiments, Technical Report UIUCDCS-R-86-1260, Department of Computer Science, University of Illinois at Urbana-Champaign, 1986.

[81] P. Clark and T. Niblett, “Induction in noisy domains, “ In I. Bratko and N. Larvac, editors, *Progress in Machine Learning: Proc 2nd European Workshop Session on Learning*, pp. 11-30, Sigma Press, 1987.

[82] P. Clark and T. Niblett, “Learning if-then rules in noisy domains, “ In B. Phelps, editor, Interaction in AI and Statistical Methods, pp. 154-168, Technical Aldershot, Hants, England, 1987.

[83] G. Pagallo and D. Haussler, “Two algorithms that learn DNF by discovering relevant features, “ *Proc. of Int. Workshop on Machine Learning*, pp. 119-123, Morgan Kaufmann, 1989.

[84] R. M. Fung and S. L. Crawford, “Constructor: A system for the induction of probabilistic models, “ Prof. 8th National Conf. on Artificial Intelligence (AAAI’90), vol. 2, pp. 762-769, 1990.

[85] J. Pearl. Probabilistic reasoning in Intelligent Systems: networks of plausible Inference. Morgan Kaufmann, 1988.

[86] D. K. Y. Chiu. Pattern Analysis Using Event-Covering, PhD thesis, Department of Systems Design, University of Waterloo, 1986.

[87] A. K. C. Wong and D. K. Y. Chiu, "An event-covering method for effective probabilistic inference, " *Pattern Recognition*, vol. 20, no. 2, pp. 245-255, 1987.

[88] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks form data, "Machine Learning, vol. 9, no. 4, pp. 309-347, 1992.

[89] D. Heckerman, D. Geiger, and D. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," Machine Learning, vol. 20, no. 3, pp. 197-244, 1995.

[90] K. Mcgarry, "A survey of interestingness measures for knowledge discovery, " The Knowledge Engineering Review, vol. 20, no. 1, pp. 39-61, 2005.

[91] P. N. Tan, V. Kumar, and J. Srivastava, "Selecting the right interestingness Measure for Association Patterns, "Proc. 8th ACM SIGKDD Int. Conf, on Knowledge Discovery and Data Mining, July 2002.

[92] L. Geng and H. J. Hamilton, " Interestingness measures for data mining: A survey, " *ACM Computing Surveys*, vol. 38, no. 3, 2006.

[93] Andrew K. C. Wong and Gary C. L. Li, "Simultaneous Pattern and Data Clustering for Pattern Cluster Analysis," In *IEEE Trans. on Knowledge and Data Engineering*, vol. 20, no. 7, pp. 911-923, 2008.

[94] Andrew K. C. Wong and Gary C. L. Li, "Using Association Patterns for Discrete-Valued Data Clustering," In *Proceedings of the 25th IASTED International Conference on Artificial Intelligence and Applications (AIA 2007)*, Innsbruck, Austria, pp. 410-416, 2007

[95] Andrew K. C. Wong and Gary C. L. Li, “Pattern Clustering and Data Grouping,” a poster paper in *The UW and IEEE Kitchener-Waterloo Section Joint Workshop on Knowledge and Data Mining*, 2006

[96] R. Srikant, Q. Vu, and R. Agrawal, “Mining Association Rules with Item Constraints, “ *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining (KDD’97)*, pp. 67-73, 1997.

[97] A. Silberschatz and A. Tuzhilin, “On subjective measures of interestingness in knowledge discovery, “Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining (KDD’95), 1995.

[98] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, Knowledge Discovery in Databases, pp. 229-248. AAAI/MIT Press, 1991.

[99] R. J. Bayardo and R. Agrawal, “Mining the most interesting rules, “ *Proc. 5th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 145-154, 1999.

[100] R. Hilderman and H. Hamilton, Knowledge discovery and interestingness measure: A survey. Technical Report CS 99-04, Department of Computer Science, University of Regina, 1999.

[101] S. Jaroszewicz and D. A. Simovici, “A general measure of rule interestingness, “ *Proc. 5th European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD 01)*, pp. 253-265, 2001.

[102] A. J. Grove, J. Y. Halpern, and D. Koller, “Random worlds and maximum entropy, “ Journal of Artificial Intelligence Research, vol. 2, pp. 33-88, 1994.

[103] A. Silberschatz and A. Tuzhilin, “What Makes Patterns Interesting in Knowledge Discovery Systems,” *IEEE Trans. on Knowledge and Data Eng.*, vol. 8, no. 6, pp. 970-974, 1996.

[104] E. Suzuki, “Autonomous discovery of reliable exception rules, “ *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining(KDD’97)*, pp. 259, 1997.

[105] E. Suzuki and Y. Kodratoff, “Discovery of surprising exception rules based on intensity of implication, “ *Proc. 2th European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD 98)* , pp. 10-18, 1998.

[106] P. Hoschka and Willi Klosgen, “A support system for interpreting statistical data, “ In Gregory Piatetsky-Shapiro and William J. Frawley, editors, Knowledge Discovery in Databases, pp. 325 – 345, AAAI/MIT Press, Menlo Park, CA, 1991.

[107] M. Klemettinen, H. Mannila, P. Ronkainen, T. Toivonen, and A. Verkamo, “Finding interesting rules from large sets of discovered association rules, “*Proc. off the 3rd Int. Conf. on Information and Knowledge Management (CIKM'94).*, pp. 401-407, Gaithersburg, Maryland, November 1994.

[108] G. Piatetsky-Shapiro and C. J. Matheus, “The interestingness of deviations, “ In Usama M. Fayyad and Ramasamy Uthurusamy, editors, AAAI Workshop on Knowledge Discovery in Databases, pp. 25-36, Seattle, Washington, 1994.

[109] S. Anand, A. Patrick, J. Hughes and D. Bell, “A data mining methodology for cross-sales, “ *Knowledge based Systems*, vol. 10, pp. 449 – 461, 1998.

[110] S. Jaroszewicz, D. A. Simovici, “Interestingness of frequent itemsets using Bayesian networks as background knowledge, “*Proc. 10th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pp. 178 – 186 , 2004.

[111] J. Han, Y. Cai and N. Cercone, “Knowledge discovery in databases: an attribute-oriented approach, “ *Proc. 18th Int. Conf. on Very Large Databases (VLDB)*, pp. 547-559, 1992.

[112] G. Adomavicius and A. Tuzhilin, “Expert-driven validation of rule-based user models in personalization applications”, *Data Mining and Knowledge Discovery*, vol.5, no.1/2, 2001.

[113] N. Wrigley, *Categorical Data Analysis for Geographers and Environmental Scientists*, Longman, 1985.

[114] Yang Wang, High-order pattern discovery and analysis of discrete-valued data sets. PhD thesis, Department of Systems Design, University of Waterloo, 2000.

[115] D.R. Cox and E.J.A. Snell, “General Definition of Residuals,” *Journal of Royal Statistical Society*, B-30, pp. 248-265, 1968.

[116] S. J. Haberman, “The Analysis of Residuals in Cross-Classified Tables,” *Biometrics*, 29, pp. 205 – 220, 1973.

[117] R. Kohavi, D. Sommerfield, and J. Dougherty, *Data Mining Using MLC++: A Machine Learning Library in C++. Tools with Artificial Intelligence*. IEEE CS Press, 1996.

[118] A. Ben-Hur and I. Guyon, *Detecting Stable Clusters using Principal Component Analysis in Methods in Molecular Biology*, Humana press, Clifton, UK, 2003.

[119] S. Basu, A. Banerjee, and R. J. Mooney, “Semi-supervised clustering by seeding, “ *Proc. 19th Int. Conf. on Machine Learning (ICML’ 02)*, 2002.

[120] A. Demiriz, K. P. Bennett, M. J. Embrechts, “Semi-supervised clustering using genetic algorithms, “*Artificial Neural Networks in Engineering (ANNIE’99)*, 1999.

[121] D. Klein, S. D. Kamvar, and C. Manning, “From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering, “ *Proc. 9th Int. Conf. on Machine Learning (ICML’ 02)*, 2002.

[122] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, “Constrained k-means clustering with background knowledge, “ Proc. 18th Int. Conf. on Machine Learning (ICML’01), 2001.

[123] E. P. Xing, A. Y. Ng, M. I. Jordan and S. Russell, “Distance metric learning, with application to clustering with side-information, “*Advances in Neural Information Processing Systems*, 2003.

[124] J. Y. Ching, A. K. C. Wong, and K. C. C. Chan, “Class-Dependent Discretization for Inductive Learning from Continuous and Mixed-Mode Data,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 631-641, 1995.

[125] A. K. C. Wong and D. K. Y. Chiu, “Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 9, no. 8, pp. 796-805, 1987.

[126] Gary C. L. Li and Andrew K. C. Wong, “Analysis of Pattern Distance Measures in Categorical Data for Pattern Pruning and Clustering, “ submitted to *Pattern Recognition*, under second revision.

[127] Andrew K. C. Wong and Gary C. L. Li, “Pattern Pruning, Pattern Clustering and Summarization, “ submitted to Frontiers in Data Mining Research, Advanced Information and Knowledge Processing, Springer.