Al al-Bayt University

Prince Hussein bin Abdullah College of Information Technology

Department of computer science

A Hybrid Strategy for Minimizing Association Rules

by:

Saif Mohammed Ali

Supervisor:

Dr. Mamoun Al-Rababaa

Co-supervisor:

Prof.Dr. Alaa Hussain Al-Hamami

2009/2010

Dedication

To the person who taught me to give without return...to the person that I'm carrying his name with pride…. I ask God to prolong your life so that you can see the fruits of your efforts, and after a long time, your words will always become starts that guided me and till today, tomorrow and forever…. to my dear father.

To my angel in life, to the meaning of love, to the source of compassion and dedication.... to the smile of life and it's secret, to the person who prayed for my success to the dearest person…. my beloved mother.

To my soul mate, to my life companion…. to the person who has the purest heart, and sincere intentions.... to the candle that lights the darkness of my life.... my Loyal wife.

Saif

Acknowledgment

Praise is to Allah and peace is upon our prophet Mohammed, I would like to offer sincere thanks and gratitude to my supervisors Prof.Dr. Alaa Hussain Al-Hamami and Dr. Mamoun Al-Rababaa for their support, and I would like to thank everyone who lighted a candle in the way of knowledge and help to give an idea to educate us, To all professors in the College of Information Technology**.**

I would like to thank the examining committee, all my colleagues and relative for supporting me from the beginning. Consequently many thanks go to all lecturers, and the administrative staff of AL al-Bayt University for their support.

All thanks to my father and mother for their prayers and courage, and special thanks to my adored wife who from the beginning had confidence in my abilities not only to attain a degree, but also to attain in with excellence. Also a lot of thanks to my brother and sisters**.**

Table of Contents

Chapter One Introduction

Chapter Two Associative Classification

Chapter Three The Proposed Strategy

Chapter Four Implementation and Experiment Results

Chapter Five Conclusions and Future Work

List of Tables

List of Figures

Abstract

Associative classification is an integrating association rules mining and classification. Associative Classification discovered patterns for useful information are important and in great demand in science, engineering and business. Today, effective associative classification methods have been developed and widely used in various challenging industrial and business applications. These methods attempt to provide significant and useful information for decision makers. Paradoxically, associative classification itself can produce such huge amounts of rules that poses a knowledge management problem: to tackle thousands or even more rules discovered in a dataset. In order to solve the rules number problem, in this study associative classification process involves two stages. First: the data is preprocess to improve the classification ability and the efficiency of the classification process. Second: by using concept of sample itemset matching, the hybrid system of data post-analysis is proposed. The hybrid system includes data pruning, data clustering and data summarization to support effective analysis and interpretation of the discovered rules. In data pruning the frequent and redundant itemsets are prune and data clustering using to grouping those itemsets results from data pruning into similar clusters. Finally, data summarization using to prune itemsets in each clusters those results from data clustering. In the Hybrid Strategy approach, we use the composite criteria of minimum support and minimum confidence as the rule weight to indicate the significance of the rule. Through our study, we find it is important to find a good combination of these two rule interestingness threshold values.

By using datasets from UCI Repository, we compared the Hybrid Strategy results with some well-known associative classification algorithms such as CMAR, CPAR, TFPC and Classification by Bagged Consistent Itemset Rules. The results that achieved by Hybrid Strategy not only give better classification accuracy but also have better results of rules number. Hybrid Strategy generates a much smaller set of high-quality predictive rules with high classification accuracy compare with these algorithms.

Keywords: Data Mining, Associative Classification, Association Rules.

Chapter One

Introduction

1.1 Overview

In the last years the development of Information Technology has motivated a parallel growing of facilities to store and manage database. The largest amount of stored data, the more important the demand of extracting the implicit information they contain to aid decision-making in business, health care services, research, etc. and thus to obtain useful knowledge from data in large repositories, i.e. the "Knowledge Discovery" , is recognized as a basic necessity in many areas.

Since the nineties of last century the research area named "Data Mining" has become central topic in Databases and Artificial Intelligence. Data mining is the process of automatically discovering useful information in large data repositories. Data mining techniques are deployed to scour large database in order to find novel and useful patterns that might otherwise remain unknown. They also provide capabilities to predict the outcome of a future observation [44].

One the basic topic of data mining is Association Rule Mining. Association rule mining finds interesting association or correlation relationships among a large set of data items. With massive amounts of data continuously being collected and stored in databases, many industries are becoming interested in mining association rules from their databases. For example, the discovery of interesting association relationships among huge amounts of business transaction records can help catalog design, cross-marketing, lossl-eader analysis, and other business decision making processes [4].

Classification also basic topic of Data mining, Classification is the processing of finding a set of models (or functions) which describe and distinguish data classes or concepts, for the purposes of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data [44]. In the machine learning literature, classification usually means establishing rules by which we can classify new data into the existing classes that are known in advance. Each data is assumed to belong to a predetermined class, and this class is called a class label [36].

Association rules mining and classification can be viewed within a unifying framework of rule discovery. Associative classification is integrating association rules mining and classification, and has been extensively studied as a new classification approach [29, 32, 52]. It uses association rule mining algorithms, such as Apriori [4] to generate the complete set of association rules. Then it selects a small set of high quality rules and uses this rule set for prediction. It is usually find only rules with high support values (support is the percentage of records in database that contains two items or more together), because reducing the minimum support to be satisfied increases computational cost, and also rules with low support but high confidence (confidence is the percentage of records in database containing x-item that also contain y-item) may contribute to classification.

The problem of previous approaches in associative classification [29,52,13] generates a very large number of association rules, which leads to high processing overhead, and also it takes efforts to select high-quality rules from among them. In this study, generates a much smaller set of high-quality predictive rules with high classification accuracy compare with previous approaches [29,52,13,41] and this could be done by, first pre-process the data to make data easier to mine for knowledge, and then post-analysis include pruning the redundant itemsets, clustering the itemsets into coherent itemsets clusters for a better organization and automatic summarization of each itemsets cluster.

1.2 Data Mining

This chapter will give a brief introduction to data mining. Data Mining is defined as the discovery of interesting information, patterns or trends from a large database or data warehouse [23]. Data mining is a subprocess of Knowledge Discovery in Databases in which the different available data sources are analyzed using various data mining algorithms. Speaking of Data Mining we refer to “a multi-disciplinary field involving machine learning, statistics, databases, artificial intelligence, information retrieval and visualization” [33].

The two high-level goals that data miners want to achieve are prediction and description [16]:

- Prediction is used to find patterns which can be used to project the future.
- Description represents discovered patterns to the user in a human-understandable form.

The two high-level goals in data mining can be defined as prediction and description [15]. Prediction uses some fields or variables in the database to predict future values of other interesting variables. Description sets focus on making the data comprehensible and interpretable. The boundaries between those two goals are not sharp, because some predictive models can be descriptive meaning that they are understandable and vice versa. Nevertheless, the distinction can help understanding the overall mining goal.

Many different data mining techniques have been developed which will be discussed in more detail later. A great number of these techniques descend from classical statistics; nevertheless there are newer approaches that use artificial intelligence approaches. This makes a prediction of trends possible which would not have been feasible with the traditional statistical methods. In addition, the data needed to conduct the data mining is widely available in the age of the Internet and e-commerce.

1.3 Data Mining Techniques

The goals of prediction and description are achieved by using the primary data mining techniques such as Classification, Association Rules, Regression, Clustering, Estimation and Prediction. We investigate more details about these techniques that usually used in data mining:

1.3.1 Classification

Individual items are placed into predefined groups. The main task is to find the function that maps data items correctly into the several classes [4]. Data mining techniques can be classified according to different views, including the kinds of knowledge to be discovered, the kinds of databases to be mined, and the kinds of techniques to be adopted. For example, for the kinds of knowledge to be mined, one may classify data mining techniques into generalization, characterization, association, classification, clustering, pattern matching, etc., or based on the level of concepts to be discovered, into primitive level, high level, multiple-level, etc.

In classification, there is a target categorical variable. For example, in the medical field, suppose that we are interested in classifying the type of drug a patient should be prescribed, based on certain patient characteristics, such as the age of the patient and the patient's sodium/potassium ratio. Figure 1.1 is a scatter plot of patients' sodium/potassium ratio against patients' ages for a sample of 200 patients. The particular drug prescribed is symbolized by the shade of the points. Light gray points indicate drug Y; medium gray points indicate drug A or X; dark gray points indicate drug B or C. This plot was generated using the data mining software suite. In this scatter plot, Na/K (sodium/potassium ratio) is plotted on the *Y* (vertical) axis and age is plotted on the *X* (horizontal) axis.

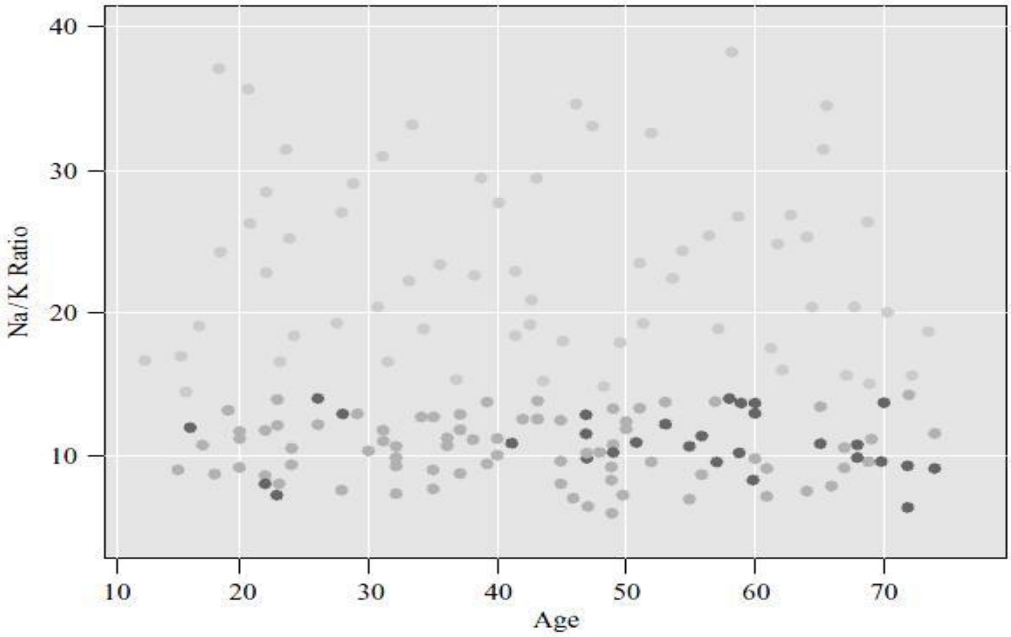


Figure 1.1: classification example [21]

1.3.2 Association Rules

Association rule mining is a fundamental topic in Data mining [12]. Association rule mining finds interesting association or correlation relationships among a large set of data items. Its key idea consists in looking for causal relationships between sets of items, commonly called itemsets, where the presence of some items suggests that others follow from them. A typical example of a successful application of association rules is the market basket analysis, where the discovered rules can lead to important marketing and management strategic decisions. Recently, mining association rules was extended to various pattern classes like sequential patterns, graphs, etc. Nevertheless, the main moan that can be addressed to the contributions related to association rules is their focus on co-occurrences between items [43], probably as a heritage of the market basket analysis framework. Indeed, almost all related works neglect the other kinds of relations, like mutually exclusive occurrences [47] that can also bring information of worth interest for users.

For example, a particular supermarket may find that of the 1000 customers shopping on a Thursday night, 200 bought milk, and of those 200 who bought milk, 50 bought tea. Thus, the association rule would be “If buy milk, then buy tea” with a support of 200/1000 = 20% and a confidence of 50/200 = 25%.

1.3.3 Regression

A dependent variable is mapped to an independent variable. Here, a function should be learned that maps data points to real-valued prediction value. Cohesion between different variables can thereby be discovered. The results are mainly used for prediction, for example predicting customer's buying behavior after a certain amount of promotion expenditures [44].

1.3.4 Clustering

Refers to the grouping of records, observations, or cases into classes of similar objects. A cluster is a collection of records that are similar to one another, and dissimilar to records in other clusters. Clustering differs from classification in that there is no target variable for clustering. The clustering task does not try to classify, estimate, or predict the value of a target variable. Instead, clustering algorithms seek to segment the entire dataset into relatively homogeneous subgroups or clusters, where the similarity of the records within the cluster is maximized and the similarity to records outside the cluster is minimized.

Clustering Unlike the regression, clustering is a descriptive task rather than predictive. The goal is to find a finite number of unknown categories in a dataset. An application is the subdivision of the customers in a database into several homogeneous subcategories to better understand their behavior. “Closely related to clustering is the task of probability density estimation which consists of techniques for estimating from data the joint multivariate probability density function of all of the variables or fields in the database.“ [4].

For example, clustering can be performed on AllElectronics customer data in order to identify homogeneous subpopulations of customers. These clusters may represent individual target groups for marketing. Figure 1.2 shows a 2-D plot of customers with respect to customer locations in a city. Three clusters of data points are evident [21].

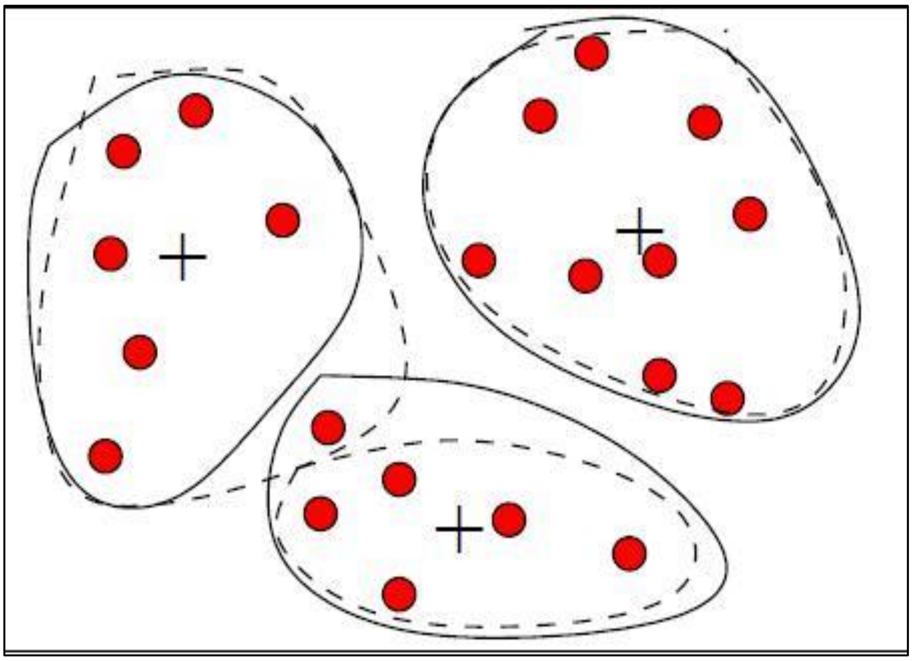


Figure 1.2:clustering example, a 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster `center' is marked with a `+' [21].

1.3.5 Estimation

Estimation is similar to classification except that the target variable is numerical rather than categorical. Models are built using complete records, which provide the value of the target variable as well as the predictors. Then, for new observations, estimates of the value of the target variable are made, based on the values of the predictors [4]. For example, Figure 1.3, where we have a scatter plot of the graduate grade-point averages (GPAs) against the undergraduate GPAs for 1000 students. Simple linear regression allows us to find the line that best approximates the relationship between these two variables, according to the least-squares criterion. The regression line, indicated in black in Figure 1.3, may then be used to estimate the graduate GPA of a student given that student's undergraduate GPA.

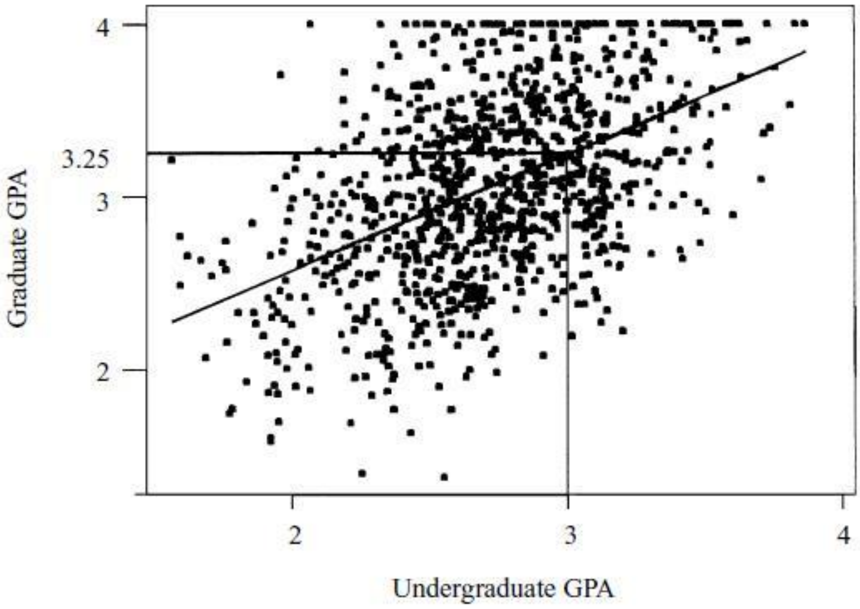


Figure 1.3 Regression estimates [21]

1.3.6 Prediction

Prediction is similar to classification and estimation, except that for prediction, the results lie in the future. Prediction may need to be preceded by relevance analysis which attempts to identify attributes that do not contribute to the prediction process. These attributes can then be excluded. For example Figure 1.4 predicting the price of a stock three months into the future [44].

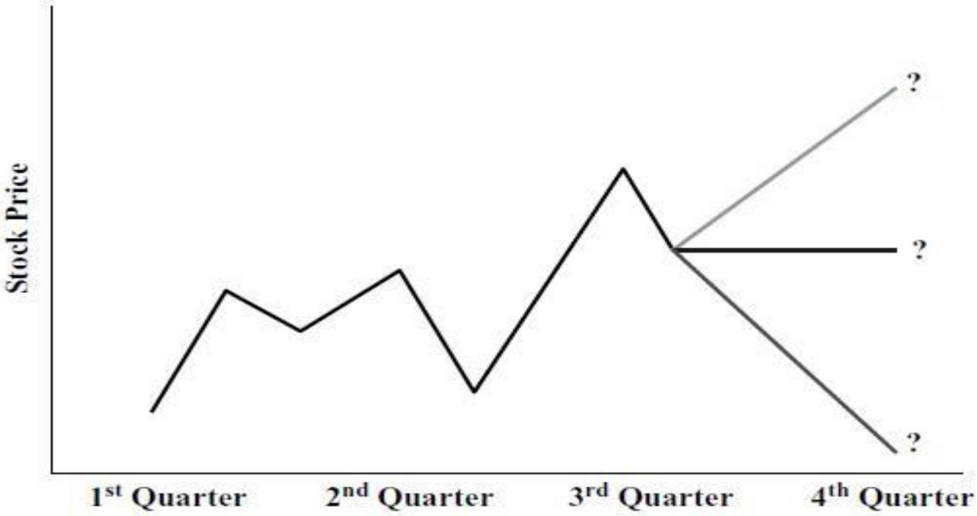


Figure 1.4 predicting the price of a stock three months in the future.

1.4 Main Algorithms of Data Mining

The data mining algorithm is the mechanism that creates a data mining model [4]. To create a model, an algorithm first analyzes a set of data and looks for specific patterns and trends. The algorithm uses the results of this analysis to define the parameters of the mining models. These parameters are the applied across the entire dataset to extract actionable patterns and detailed statistics. The basic data mining algorithms is C4.5 and beyond, *k*-means algorithm, Support vector machines, Apriori algorithm, PageRank, Naive Bayes, CART, in this section we introduce sample details about these algorithms:

1.4.1 C4.5 and beyond

Complex decision trees can be difficult to understand, for instance because information about one class is usually distributed throughout the tree. C4.5 introduced an alternative formalism consisting of a list of rules of the form "if A and B and C and ... then class X", where rules for each class are grouped together. A case is classified by finding the first rule whose conditions are satisfied by the case; if no rule is satisfied, the case is assigned to a default class [40].

C4.5 rulesets are formed from the initial (unpruned) decision tree. Each path from the root of the tree to a leaf becomes a prototype rule whose conditions are the outcomes along the path and whose class is the label of the leaf. This rule is then simplified by determining the effect of discarding each condition in turn. Dropping a condition may increase the number N of cases covered by the rule, and also the number E of cases that do not belong to the class nominated by the rule, and may lower the pessimistic error rate determined as above. A hill-climbing algorithm is used to drop conditions until the lowest pessimistic error rate is found.

To complete the process, a subset of simplified rules is selected for each class in turn. These class subsets are ordered to minimize the error on the training cases and a default class is chosen. The final ruleset usually has far fewer rules than the number of leaves on the pruned decision tree.

1.4.2 K-means algorithm

Clustering is the process of partitioning or grouping a given set of patterns into disjoint clusters [44]. This is done such that patterns in the same cluster are alike and patterns belonging to two different clusters are different. Clustering has been a widely studied problem in a variety of application domains including neural networks, AI, and statistics.

The k-means method has been shown to be effective in producing good clustering results for many practical applications. However, a direct algorithm of k-means method requires time proportional to the product of number of patterns and number of clusters per iteration. This is computationally very expensive especially for large datasets [25].

1.4.3 Support vector machines

A Support Vector Machine (SVM) performs classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories. SVM models are closely related to neural networks. In fact, a SVM model using a sigmoid kernel function is equivalent to a two-layer, perception neural network.

Support Vector Machine (SVM) models are a close cousin to classical multilayer perception neural networks. Using a kernel function, SVM’s are an alternative training method for polynomial, radial basis function and multi-layer perception classifiers in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training[48].

In the parlance of SVM literature, a predictor variable is called an attribute, and a transformed attribute that is used to define the hyperplane is called a feature. The task of choosing the most suitable representation is known as feature selection. A set of features that describes one case (i.e., a row of predictor values) is called a vector. So the goal of SVM modeling is to find the optimal hyperplane that separates clusters of vector in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other size of the plane. The vectors near the hyperplane are the support vectors [44].

1.4.4 Apriori algorithm

One of the most popular data mining approaches is to find frequent itemsets from a transaction dataset and derive association rules. Finding frequent itemsets (itemsets with frequency larger than or equal to a user specified minimum support) is not trivial because of its combinatorial explosion. Once frequent itemsets are obtained, it is straightforward to generate association rules with confidence larger than or equal to a user specified minimum confidence.

Apriori is a seminal algorithm for finding frequent itemsets using candidate generation [44]. It is characterized as a level-wise complete search algorithm using anti-monotonicity of itemsets, “if an itemset is not frequent, any of its superset is never frequent”. By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. Let the set of frequent itemsets of size k be Fk and their candidates be Ck. Apriori first scans the database and searches for frequent itemsets of size 1 by accumulating the count for each item and collecting those that satisfy the minimum support requirement. It then iterates on the following three steps and extracts all the frequent itemsets [4]:

1. Generate $Ck+1$, candidates of frequent itemsets of size $k+1$, from the frequent itemsets of size k.
2. Scan the database and calculate the support of each candidate of frequent itemsets.
3. Add those itemsets that satisfies the minimum support requirement to $Fk+1$.

1.4.5 PageRank

It is a search ranking algorithm using hyperlinks on the Web. Based on the algorithm, they built the search engine Google, which has been a huge success. Now, every search engine has its own hyperlink based ranking method [10]. PageRank produces a static ranking of Web pages in the sense that a PageRank value is computed for each page off-line and it does not depend on search queries. The algorithm relies on the democratic nature of the Web by using its vast link structure as an indicator of an individual page’s quality. In essence, PageRank interprets a hyperlink from page x to page y as a vote, by page x, for page y. However, PageRank looks at more than just the sheer number of votes, or links that a page receives. It also analyzes the page that casts the vote. Votes casted by pages that are themselves “important” weigh more heavily and help to make other pages more “important”. This is exactly the idea of rank prestige in social networks [50].

1.4.6 Naive Bayes

Given a set of objects, each of which belongs to a known class, and each of which has a known vector of variables, our aim is to construct a rule which will allow us to assign future objects to a class, given only the vectors of variables describing the future objects. Problems of this kind, called problems of supervised classification, are ubiquitous, and many methods for constructing such rules have been developed. One very important one is the naive Bayes method—also called idiot's Bayes, simple Bayes, and independence Bayes. This method is important for several reasons. It is very easy to construct, not needing any complicated iterative parameter estimation schemes. This means it may be readily applied to huge datasets. It is easy to interpret, so users unskilled in classifier technology can understand why it is making the classification it makes. And finally, it often does surprisingly well: it may not be the best possible classifier in any particular application, but it can usually be relied on to be robust and to do quite well [14, 22].

1.4.7 Classification and Regression Trees

The Classification and Regression Trees (CART) decision tree is a binary recursive partitioning procedure capable of processing continuous and nominal attributes both as targets and predictors. Data are handled in their raw form; no binning is required or recommended. Trees are grown to a maximal size without the use of a stopping rule and then pruned back (essentially split by split) to the root via cost-complexity pruning. The next split to be pruned is the one contributing least to the overall performance of the tree on training data (and more than one split may be removed at a time). The procedure produces trees that are invariant under any order preserving transformation of the predictor attributes. The CART mechanism is intended to produce not one, but a sequence of nested pruned trees, all of which are candidate optimal trees. The "right sized" or "honest" tree is identified by evaluating the predictive performance of every tree in the pruning sequence. CART offers no internal performance measures for tree selection based on the training data as such measures are deemed suspect. Instead, tree performance is always measured on independent test data (or via cross validation) and tree selection proceeds only after test-data-based evaluation. If no test data exist and cross validation has not been performed, CART will remain agnostic regarding which tree in the sequence is best. This is in sharp contrast to methods such as C4.5 that generate preferred models on the basis of training data measures. The CART mechanism includes automatic (optional) class balancing, automatic missing value handling, and allows for cost-sensitive learning, dynamic feature construction, and probability tree estimation [8].

1.5 The Problem Statement

Associative classification uses association rule mining algorithms to generate the sets of association rules. The classification approaches uses to selects a set of high quality rules. The main problem that makes associative classification algorithms inefficient and difficult for the end users is the large number of rules in association rule mining. This lead to the number of attributes are much more the number of rows data (reduce the speed of data analysis), and the relationships between attributes are often more complex than rows data (need high processing). The problem of the huge number of rules has continually been attracting more and more attentions from both the academia and the industry

The problem of rules number there are so many possible rules. For the product range of a supermarket, for example, which may consist of several thousand different products or even millions of possible association rules. It is obvious that such a vast number of rules cannot be processed by inspecting each one in turn. Efficient algorithms are needed that restrict the search space and check only a subset of all rules, but, if possible, without missing important rules.

The importance of a rule is usually measured by two numbers: Its support and its confidence. To select interesting rules from the set of all possible rules, a minimum support and a minimum confidence are fixed. If the minimum support and minimum confidence are high, only commonsense "knowledge" will be found. However, when minimum support and minimum confidence are set low, a huge number of association rules will usually be generated, a majority of which are redundant or no informative.

Many studies proposed many algorithms to solve this problem in [1, 3], one strategy to improve the efficiency involves the use of random sampling to estimate the support of an itemset [45]. Use of hash-based efficient data structures [42] and mining of vertical (rather than horizontal) database [9] are some other approaches that have been tried to improve the efficiency of the association rule mining algorithms. In this study we develop a system to solve the above problem by, automatically analyze the complex relationship among the rules discovered by rules mining and organize them in a way that human can easily process.

The objective of this study is to reduce the number of the association rules with high classification accuracy to improve the performance of data mining technique in the application. This study deals with the following dimensions of the problem:

1- Complex relationship among itemsets.
2- Pruning of redundant or frequent itemsets.
3- Clustering of discovering itemsets.
4- Summarization of itemsets in each cluster.

1.6 The Study Contribution

Most of the previous approaches of associative classification suffer from the problem of generates a very large number of association rules. In this study we developed a method to solve the problem of large number of association rules by automatically analyzing, organizing and managing the discovered rules, so that the end users can obtain useful information from them. Here the associative classification process involves two stages: data preprocess stage and post-analysis stage.

Data preprocessing techniques can improve the quality of the data, thereby helping to improve the accuracy and efficiency of the subsequent mining process. Data preprocessing is therefore an important step in the knowledge discovery process, since quality decisions must be based on quality data. Detecting data anomalies, rectifying them early, and reducing the data to be analyzed can lead to huge pay-offs for decision making.

The post-analysis stage includes data pruning, data clustering and data summarization. The following list gives more details of the steps of the post-analysis:

1- Pruning of redundant itemsets: pruning of redundant itemsets is highly desirable because it can reduce the complexity of the subsequent analysis. The interrelating nature of itemsets, the notion of redundancy has to be defining by considering how itemsets overlap entangle with each other in the data through the explicit itemsets-data relationship. The notion of redundancy is defined, redundant itemsets can be pruned. In this study, the concept of sample-itemset matching is used to define redundancy between itemsets.

2- Clustering of discovered itemsets and their associated rules: data clustering is an important task for organizing and analyzing the discovered data. In data clustering, the key function of clustering is to bring similar data together. In this study; a clustering process is proposed to try to cluster both itemsets and their associated rules while maintaining an explicit one-to-one itemsets-data relationship.

3- Summarization of itemsets: it is very useful in real word applications for data analysis system to automatically generate a summary of the itemsets. In this study, we develop the ability to generate a concise and informative summary describing a huge set of interrelating and entangling itemsets.

1.7 The objectives

In this study we develop a system to achieve the followings objectives:

1- Design a classifier system with strong classification ability. This includes higher classification accuracy and fewer numbers of association rules.

2- Develop the ability to prune redundant itemsets with controllable tradeoff of information loss and the number of itemsets retained.

3- Develop an effective clustering algorithm to simultaneously cluster both itemsets and their associated rules and maintains an explicit one-to-one relationship between itemsets and rules for subsequent analysis.

4- Develop the ability to generate a concise and informative summary describing a huge set of interrelating and entangling itemsets.

1.8 The Study Organization

In this study we focus on each of the above problems, first describing each problem of previous studies. We then propose solutions to the above problems. The solutions emphasize increasing classification performance and decreasing rules number. In addition to introductory chapter, there are four chapters in this study.

Chapter two divide into two parts, first part consist the definition of classification problem. Some of literature surveys in the area of associative classification with different types of methodologies will present. Traditional classification methods and estimation techniques accuracy also present. Second part; present the definition of association rule mining and the Apriori algorithm and its implementation by using Hash Tree.

Chapter three proposes the system that will solve the study problem. Also this chapter discusses the details of the system step by step and component by component. The last section present the summary explains the effectiveness of each step of proposed strategy on wine dataset.

Chapter four will implement the study problem and will do the experiment needed to test the performance of the classification and how Hybrid Strategy solves the study problem. In this chapter present an implementation of each system building block will be discussed in details. Also a screen shots will be taken to show the Hybrid Strategy performance. Then present comparisons between pervious methods and Hybrid Strategy to see the achievement of Hybrid Strategy. Different datasets taken from UCI repository used in these comparisons.

Chapter five includes the conclusion and future work. This chapter presents the conclusion extracted from each bit and piece discussed in this study. Also it puts the recommendations for future work that will be focused on the area of Hybrid Strategy.

Chapter Two

Associative Classification

2.1 Overview

Associative classification integrates association rule mining and classification into a single system [28,29,49]. Classification rule mining aims to discover a small set of rules in the database to form an accurate classifier [40,8]. Association rule mining finds all rules in the database that satisfies some minimum support and minimum confidence constraints [2]. For association rule mining, the target of mining is not pre-determined, while for classification rule mining there is one and only one pre-determined target, i.e., the class. Both classification rule mining and association rule mining are indispensable to practical data mining applications. If these two relevant jobs can be integrated, great savings and conveniences to the user could result. Hence, considerable efforts have been made to integrate these two techniques into one system. A typical associative classification system is constructed in two stages: first, discovering all the association rules in which the frequency of occurrences is significant according to some tests; second, generating classification rules from the association rules to build a classifier. In the first stage, the learning target is to discover the association rules inherent in a database (also referred to as knowledge discovery). In the second stage, the task is to select a small set of association rules discovered to construct a classifier given the predicting attribute.

This chapter divided into two parts: first part, provides a brief introduction to classification. The definition of the classification problem is given in Section 2.2. In section 2.2.1 we will present some of literature surveys in the area of classification and association rules. Traditional classification methods are introduced in Section 2.2.2, which include statistical classification methods (Section 2.2.2.1) and the decision tree method (Section 2.2.2.2).Estimation techniques of classification accuracy are introduced in Section 2.2.3. Second part, provides a brief introduction to the association rule mining. We provide the definition of association rule mining and its classification scheme in Section 2.3. One of the most important association rule mining algorithms, the Apriori, is explained in Section 2.3.1. Section 2.3.2 presents the implementation of Hash Tree algorithm.

2.2 Definition of Classification

The classification approach based on rules provides a modularized, clearly explained format for a decision, which is compatible with a human being's reasoning procedure. Classification produces a function that maps a data item into one of several predefined classes, by inputting a training dataset and building a model of the class attribute based on the rest of the attributes [35]. Classification is also known as supervised learning, in the sense that the classification rules are established from given data whose class labels have been known. This contrasts with unsupervised learning, such as clustering, in which the class label of each given data, and the number of classes to be learned may not be known in advance.

Classification can be divided into different categories. In the first type of classification, the class label's value is discrete or nominal, such as "Yes/No", "High/Medium/Low", etc. In the second type of classification, the class label's value is continuous or ordered. Different classification methods can be applied to these two types of problems. For instance, a decision tree is the common technique to be used in the first type of classification problem, while regression is used for the second type.

Classification is an important data mining problem [21,4] and can be described as follows. The input data, also called the training set, consists of multiple examples (records), each having multiple attributes or features. Additionally, each example is tagged with a special class label. The objective of classification is to analyze the input data and to develop an accurate description or model for each class using the features present in the data. The class descriptions are used to classify future test data for which the class labels are unknown. They can also be used to develop a better understanding of each class in the data. Applications of classification include credit approval, target marketing, medical diagnosis, treatment effectiveness, store location, etc. The decision tree is the most popular approaches to solve the classification problem [3]. Association rules can also be used for classification [26,32].

2.2.1 Review of Literature

A lot of studies and researches presented and focused on association rules and associative classifiers. Here, we shall look at the main algorithms of these studies such as: Mining Association Rules Between Sets of Items in Large Databases, Fast algorithms for mining association rules and other for association rules and Accurate and Efficient Classification Based on Multiple Class-Association Rules (CMAR), Classification based on Predictive Association Rules (CPAR) and others for associative classifiers.

1- **Rakesh Agrawal, Tomasz Imielinski, Arun Swmi** [1] proposed an approach called (Mining Association Rules between Sets of Items in Large Databases) by introduced the problem of mining association rules between sets of items in large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. Then interested in finding those rules that have, minimum support and minimum confidence. Then proposed an efficient algorithm to solve this problem which has the following features, first, it uses a carefully tuned estimation procedure to determine what itemsets should be measured in a pass. Second, it uses pruning techniques to avoid measuring certain itemsets, while guaranteeing completeness. Third, It incorporates buffer management to handle the fact that all the itemsets that need to be measured in a pass may not fit in memory, even after pruning**.**

2- **Bing Liu, Wynne Hsu, Yiming Ma** [32] proposed an approach of associative classification called (Integrating Classification and Association Rule Mining). It is done by focusing on a special subset of association rules whose right-hand-side is restricted to the classification class attribute. In this study, the association rule mining algorithm [2] is adapted to mine all the class association rules that satisfy the minimum support and minimum confidence constraints. This adaptation is necessary for two main reasons: first, adaptation of the existing association rule mining algorithm to mine only the class association rules is needed so as to reduce the number of rules generated. Second, classification datasets often contain many continuous (or numeric) attributes; adaptation involves discrediting continuous attributes based on the classification predetermined class target.

3- **Tom Brijs, Koen Vanhoof and Geert Wets** [9] proposed an approach called (Reducing redundancy in characteristic rule discovery by using integer programming techniques) that introduce integer programming model to solve the problem of optimally selecting the most promising subset of characteristic rules. Moreover, the proposed technique enables to control a user-defined level of overall quality of the model in combination with a maximum reduction of the redundancy extant in the original ruleset. This study is using real-world data to empirically evaluate the benefits and performance of the proposed technique against the well-known RuleCover heuristic. By using two integer programming models to tackle the problem of redundancy in a set of characteristic rules. The first model searches for an optimal selection of rules that is able to maximally reduce redundancy given the constraint of covering all (positive) instances that are covered by the original ruleset. In the second model, the first model was adapted (by incorporating two parameters α and β) to account for a flexible adjustment of the completeness and discriminate power of the final ruleset.

4- **W. Li, J. Han, and J. Pei.** [29] Proposed associative classification method by called, CMAR (Accurate and Efficient Classification Based on Multiple Class-Association Rules), for accurate and efficient classification and make the following contributions. First, instead of relying on a single rule for classification, CMAR determines the class label by a set of rules. Given a case for prediction, CMAR selects a small set of high confidence, highly related rules and analyzes the correlation among those rules. To avoid bias, develop a technique, called weighted X^2, which derives a good measure on how strong the rule is under both conditional support and class distribution. Second, to improve both accuracy and efficiency, CMAR employs a novel data structure, CR-tree, to compactly store and efficiently retrieve a large number of rules for classification. CR-tree is a prefix tree structure to explore the sharing among rules, which achieves substantial compactness. CR-tree itself is also an index structure for rules and serves rule retrieval efficiently. According to 18 datasets from UCI machine learning repository, the average number of rules is (584.3) and the average number of accuracy is (82.0) [13].

5- **Yin, X. and Han, J.** [52] proposed classification approach called CPAR (Classification based on Predictive Association Rules). Which combines the advantages of both associative classification and traditional rule-based classification. Instead of generating a large number of candidate rules as in associative classification, CPAR adopts a greedy algorithm to generate rules directly from training data. CPAR generates and tests more rules than traditional rule-based classifiers to avoid missing important rules. CPAR employs the following features to further improve its accuracy and efficiency: (1) CPAR uses dynamic programming to avoid repeated calculation in rule generation; and (2) when generating rules, instead of selecting only the best literal, all the close-to-the-best literals are selected so that important rules will not be missed. According to 18 datasets from UCI machine learning repository, the average number of rules is (95) and the average number of accuracy is (81.2) [13].

6- **France Coenen, Paul Leng, Lu Zhang** [13] This study proposed approach called TFPC (Threshold Tuning for Improved Classification Association Rule Mining) that describe a Classification Association Rule Mining algorithm to avoids the need for coverage analysis, and a technique for tuning its threshold parameters to obtain more accurate classification. This can be done by identify three general problems of Classification Association Rule Mining. First, the Association Rule Mining task is inherently costly because of the exponential complexity of the search space. Second, generate a very large number of candidate rules, and so the selection of a suitable subset for classification may also be computationally expensive. Finally, the reliability of the resulting classifier depends in some degree on the rather arbitrary choice of support and confidence thresholds used in the mining process. This study describes an approach to the generation of Classification Association Rules, that significantly reduces the cost of mining the training data by using both support and confidence thresholds in the first stage of mining, to produce a small set of rules without the need for coverage analysis. According to 18 datasets from UCI machine learning repository, the average number of rules is (298) and the average number of accuracy is (80.7) [13].

7- **Girish K. Palshikar, Mandar S. kale, Manoj M.Apte** [19] proposed an approach called (Association Rules Mining Using Heavy Itemsets) that introduce a concept called a heavy itemset. An itemset *A* is heavy (for given support and confidence values) if all possible association rules made up of items only in *A* are present. And prove a simple necessary and sufficient condition for an itemset to be heavy. Then using a formula for the number of possible rules for a given heavy itemset, and show that a heavy itemset compactly represents an exponential number of association rules. The efficient greedy algorithm used to generate a collection of disjoint heavy itemsets in a given transaction database. Then using a modified apriori algorithm that uses a given collection of heavy itemsets and detects more heavy itemsets, not necessarily disjoint with the given ones, and of course the remaining association rules.

8- **Mohammed J. Zaki, Ching-Jui Hsiao** [53] This study proposed CHARM (Closed Association Rule Mining), an algorithm for mining closed frequent itemsets, and CHARM-L, an algorithm for generating the closed itemset lattice. These include the following features: First, they simultaneously explore both the itemset space and transaction space over a novel IT-tree (itemsettidset tree) search space. Second, they use a highly efficient hybrid search method that skips many levels of the IT-tree to quickly identify the frequent closed itemsets, instead of having to enumerate many possible subsets. Third, CHARM uses a fast hash-based approach and CHARM-L uses an intersection-based approach to eliminate nonclosed itemsets during subsumption checking. Forth, CHARM-L explicitly outputs the frequent itemset lattice, which is useful for rule generation and visualization.

9- **Yohji Shidara, Mineichi Kudo, Atsuyoshi Nakamura**[41] This study proposed an approach called (Classification by Bagged Consistent Itemset Rules)**,** which build a classifier composed of almost all consistent (100% confident) rules. In this approach, confidence is given more priority than support. Regardless of their values of support, more rules with a high confidence value can be found efficiently. Note that an individual instance can be regarded as a 100% confident rule itself, while the support value of the corresponding rule is quite low (such a rule might explain only one instance.). They are the most specific rules. Merge some instances by taking intersection (the set of common items among them) to make them more general. This merging is done only when the consistency (explaining instances of one class only) is maintained. As a consequence, then obtain the most general rules, in the set inclusion relation, keeping consistency. In addition, such rules are expected to be highly interpretable because they represent some unique patterns for a class.

2.2.2 Traditional Classification Techniques

We will briefly review two major categories of traditional classification techniques here. The first category is statistical classification methods that including Bayesian classifiers, linear discriminat and the k-nearest neighbor method. The second category is the decision tree method.

2.2.2.1 Statistical Classification Methods

The classification problem has been widely studied in pattern recognition and statistics. Many methods were developed in these two areas.

Bayesian Classifiers

Bayesian classifiers are a classification method based on Bayes Theorem for analysis the classification problems [34]. The formal definition of a Bayesian Classifiers is given in [44]: let a record is denoted as *t*, and all records are assigned to q known class labels. Thus, we have a class label set C= {Ci} (i=1,……,q).

The basic principle of a Bayesian classifier is that it will classify the record as the class Ci with the greatest posterior probability for this record. The posterior probability of a record t toward class label Ci is given as

$P(\mathrm{Cj} \mid t) > P(\mathrm{Ck} \mid t)$ for all (j= 1…..q , k=1 , j $\neq$ k)

From Bayes rules, we know that the posterior probability of a record *t* toward Cj is given as

$$P(C_j \mid t) = \frac{P(t \mid C_j)P(C_j)}{P(t)}$$

Where $P(t \mid \mathrm{Cj})$ is the conditional probability of a given record for a specific class Cj. Simple mathematical manipulation of the Bayes rule shows that an alternative formulation of classifying a record t is to choose the class Cj to satisfy following

$P(\, t \mid \mathrm{Cj})\ P(\mathrm{Cj}) > P(\, t \mid \mathrm{Ck})$ for all j $\neq$ k

While we can use the proportion of each class in the given data to represent the prior probability of class Cj, it is fairly difficult to estimate the true population values of the conditional probability of a given record *t* for a specific class Cj. A lot of statistical classification methods can be viewed as approximations to the Bayes rule with varying assumptions made to estimate the conditional probabilities. The assumption may be a characteristic distribution of the population, or specific format for the decision solution itself.

Linear Discriminant

Linear discriminant analysis is a popular technique in computer vision, machine learning and data mining. It has been successfully applied to various problems, and there are numerous variations of the original approach. Linear discriminant is one of the most common forms of classifier, and it has a quite simple structure [24]. Assume the dataset has *m* attributes, and then the linear discriminant is to use a linear combination of the attributes to separate or discriminate among the classes and assign the class label for a new record. For a dataset with *m* attributes, this means geometrically that the separating surface between the records will be a *m*-1 hyperplane.

The major advantage of linear discrimiant is its simple classification structure. The general form of any linear discriminant is given as follows:

$W1AT1 + W2AT2 + \ldots + WmATm - W0$

Where ATj $(j = 1 \ldots m)$ are the *m* attributes of the dataset, Wi $(I = 0 \ldots m)$ are the constant parameters to be estimated. Linear discriminant tends to perform well in practice, though it is true that different classes cannot always be separated by a simple linear combination of attributes. Moreover, more than one plane or line can be used to separate two classes. The major issues of applying linear discrimants is to decide the constant parameters for the linear discriminant. The most common approach is to decide those parameters under certain assumption about the data distribution, such as a normal or Gaussian distribution.

k-nearest Neighbor Method

K-nearest neighbor classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. k-nearest neighbor method is to first find out the k-nearest neighbor of a new record, and then assign the data to the class label that appears most frequently among the k neighbors [44,4]. K is generally an odd number so that tie situation won't happen. This method needs to calculate the distance between a new record and every existing record. Distance metrics such as absolute distance, Euclidean distance, and various normalized distances are used in calculating the distance between a new record and an old record. Generally, the distance is compared attribute by attribute and then added together. For absolute distance, the difference between the values of each attribute is added together. For Euclidean distance, the difference between the values of each attribute is squared and added together for all attributes. The square root of the sum becomes the Euclidean distance. In some cases different attributes may be scaled differently, such as in different units or in different conventions; therefore, it is more appropriate to normalize the distance metric. For example, we can measure the distance in terms of standard deviations from the mean of each attribute.

The major computational effort of the k-nearest neighbor method lies in the classification stage. The new record must be compared with every existing record in the dataset. This increases the computational effort, especially for huge datasets. On the other hand, k-nearest neighbor method does not need an underlining assumption on the data distribution; therefore, it is a non-parametric method. The above two characteristics differentiate this method from parametric methods such as the linear discriminant method.

2.2.2.2 Decision Tree Method

Decision trees are powerful and popular tools for classification and prediction. Decision tree classifies data based on its top-down tree structure. Starting from the root node, each internal node in the tree specifies a test on a certain attribute of the dataset. Each branch from that node corresponds to one of the possible values of this attribute. A record is classified by starting being tested by the root node attribute, and moving down the tree branch corresponding to the value of the attribute in the given record [4]. A well-know decision tree example is given in Figure 2.1.

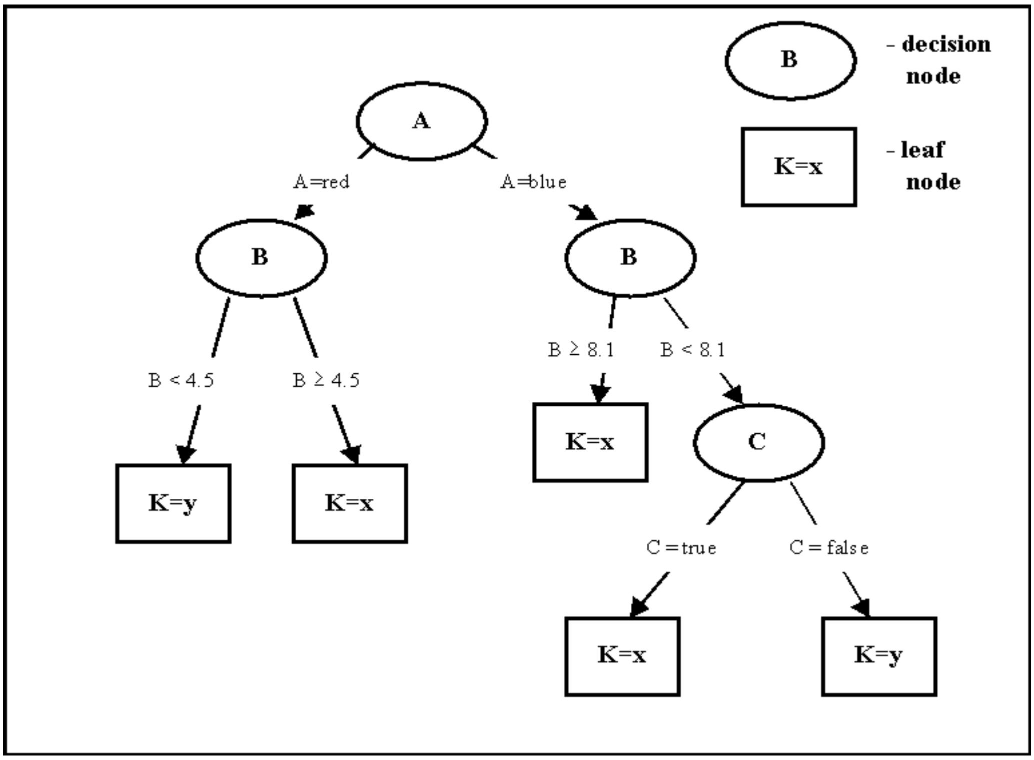


Figure 2.1: An example of a simple decision tree

The decision tree method uses a statistical property, information gain, to measure how well a given attribute separates the training dataset according to the class label. Most decision tree algorithms use this information gain to select among the candidate attributes at each step while growing the tree. The information gain is based on the entropy concept that is commonly used in information theory.

Assume we have a given dataset *T*, and the class label set C={Ci}, (i= 1…. *q*). The entropy of *T* is defined as:

$$Entropy(T) = \sum_{i=1}^{q} -P_i log_2 P_i$$

Where *Pi* is the proportion of *T* belonging to class *Ci*. The information gain of each candidate attribute is actually the expected reduction in entropy resulting from partitioning the records according to this attribute. Starting from the root node with all the records, the decision tree algorithm selects the attribute with the largest information gain. The process of selecting a new attribute and partitioning the records is repeated for each node down the tree. The algorithm stops at a certain leaf node only when all the attributes have been included in the path down to the certain leaf node, or the records associated with that leaf node all belong to the same class.

Some decision tree methods such as ID3 and C4.5 [40] perform a simple to complex search through its search space. It starts from an empty tree, then considering trees with more attributes guided by information gain heuristic. This process is illustrated in Figure 2.2.

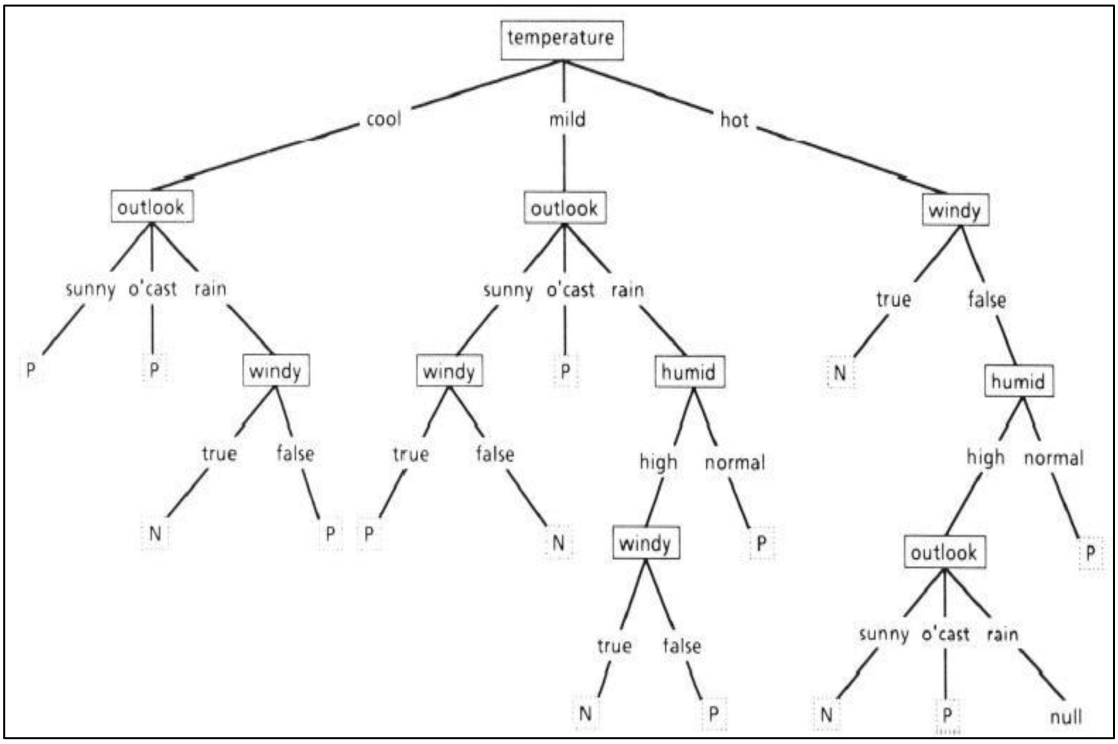


Figure 2.2: Simple to complex grow of decision tree

The search space of the decision tree method is the set of all possible decision trees. At each node, only the single best attribute is chosen to partition the records though other attributes may also be consistent with the records. Once it selects an attribute to test at a particular level of the tree, it does not backtrack to reconsider the alternative attributes at a higher level. Therefore, it loses the capability to examine all possible decision trees that are consistent with the records in the search space.

2.2.3 Classification Accuracy

A classification is a function that maps an unlabelled instance to a label using internal data structures. The purpose of classification is to build a classifier from given data and predict correctly on future data. The most commonly used performance measurement is the classification accuracy. For a given finite number of records, the empirical accuracy is defined as the ratio of the correct classification to the number of given records [21]. If the number of given records approaches infinity, then the empirical accuracy becomes statistically the true accuracy of the classifier under the actual population distribution. However, there is only limited given data in a real situation. Consequently, a technique to estimate the true accuracy from empirical accuracy becomes very important. They are at least as important as studies on the classification method itself. In order to honestly estimate the true classification accuracy, the given data should be a random sample. This means that the given data should not be pre-selected or sifted by any specific human-involved criteria. Without a random sample, the empirical accuracy based on give data will not be a good estimate of the true accuracy.

In this section, we will briefly review some common accuracy evaluation techniques such as the holdout, the k-fold cross-validation and the leaving-one-out method. In the empirical study section of this study, we will mainly use the k-fold cross-validation technique to estimate the true accuracy of the associative classification.

2.2.3.1 The Holdout Method

The holdout method sometimes called train-and-test estimation method. The holdout method partitions the data into two mutually exclusive subsets called a training set and a test set or holdout set. It is designate to 2/ 3 of the data as the training set to derive the classification model and the remaining 1/3 as the test set used for testing the classifier's accuracy. If the holdout method is repeated k times using different partition schemes, it is also referred to as random subsampling, and the accuracy estimate is taken as the average of the accuracies from each iteration. If the size of the given data is small, it is crucial to randomly divide the data into a training and a testing set, and this is often implemented by computer. The accuracy estimate by the holdout method is a bit pessimistic, because not all the data are used in designing the classifier. Besides, a single train-and-test partition may cause bias to the estimate [44].

2.2.3.2 The k-fold Cross Validation

k-fold cross validation belongs to the family of resampling methods. For k-fold cross validation, the cases are randomly divided into k mutually exclusive test partitions of approximately equal size. When we pick one test partition, the rest of the partitions are used for training of the classifier. The average accuracy rate over all k partitions is the cross validation accuracy rate. This method was extensively tested with varying numbers of partitions and 10-fold cross validation seemed to be most appropriate [51,27]. k-fold cross-validation can be extended to stratified cross-validation, in which each partition is stratified so that the class distribution of the data in each partition is approximately the same as that in the original data.

2.2.3.3 Leaving-one-out Method

Leaving-one-out method can be regarded as a special case of k-fold cross validation, where k is set to be the size of the dataset. In each iteration, this method uses one record as the testing data, and the rest will all be used for training of the classifier. The average accuracy rate over all records is used as the estimate of the accuracy. The major advantage of the leaving-one-out technique is that it introduces less bias compared to other estimation techniques. However, the computational cost for applying leaving-one-out is high, especially for a large dataset [26].

2.3 Definition of Association Rule Mining

The discovery of association rules constitutes a very important task in the process of data mining. Association rules are an important class of regularities within data which have been extensively studied by the data mining community. The general objective here is to find frequent co-occurrences of items within a set of transactions. The found co-occurrences are called associations. The idea of discovering such rules is derived from market basket analysis where the goal is to mine patterns describing the customer's purchase behavior [33].

Association rule mining was originally motivated by “market basket analysis” which studies the buying habits of customers [1]. It provides the answer to the following question: “Which groups of items are usually associated so that customers are likely to purchase them together when they shop at the supermarket?” The analysis can be performed on the retail data or customer transactions at the store. The result can be used for store layout, cross-marketing advertisement and direct mailing applications.

Assume customers who buy a computer also tend to buy financial management software. For one layout strategy, those two items can be placed in close proximity in order to further promote the sale of such items together. In an alternative strategy, placing hardware and software at opposite ends of the store may entice customers who purchase such items to pick up other items along the way [2]. Now, association rule mining has been extended to other areas like geographical databases, network security, medical diagnosis, etc.

The formal definition of an association rule is given in [2]:

Let $I = \{i_1, i_2, ...,i_m\}$ be a set of literals, called items. Let D be the set of all transactions, where each transaction t is a set of items such that $t \subseteq I$. Each transaction is associated with a unique identifier, called TID. Let A be a set of items in I. A transaction t is said to contain A if and only if $A \subseteq t$. An association rule is of the form $A \rightarrow B$, where $A \subset I$, $B \subset I$ and $A \cap B = \emptyset$. Within the rule $A \rightarrow B$, A is called the antecedent while B is called the consequent of the rule.

Support and confidence are the two key measures of rule interestingness in association rule mining. A set of items X in I is referred to as an *itemset*. X is called the k-itemset if it contains k items; t_x is the set of transactions that contain the itemset X; $| D |$is the total number of transactions;

Support: The support of the rule $A \longrightarrow B$ is the percentage of transactions in T that contain $A \cap B$. It determines how frequent the rule is applicable to the transaction set T . The support of a rule is represented by the formula

$$Sup\ (A \rightarrow B) = \frac{|\, t_A \cap t_B|}{|D|}$$

Where $|t_A \cap t_B|$ is the number of transactions that contain all the items of the rule and D is the total number of transactions.

The support is a useful measure to determine whether a set of items occurs frequently in a database or not. Rules covering only a few transactions might not be valuable to the business. The above presented formula computes the relative support value, but there also exists an absolute support. It works similarly but simply counts the number of transactions where the tested itemset occurs without dividing it through the number of tuples.

Confidence: The confidence of a rule describes the percentage of transactions containing A which also contain B .

$$Conf = \frac{Sup\ (A \rightarrow B)}{Sup(A)} = \frac{|t_A \cap t_B\ |}{|t_A|}$$

This is a very important measure to determine whether a rule is interesting or not. It looks at all transactions which contain a certain item or itemset defined by the antecedent of the rule. Then, it computes the percentage of the transactions also including all the items contained in the consequent.

From these definitions, we can infer that the values of support and confidence range from 0 to 1.

Given an example in Table 2.1, let's examine the support and confidence of a potential association rule: "Bread → Butter".

$$Sup = \frac{Sup\ (t_{Bread} \cap t_{Butter})}{|D|} = \frac{2}{5} = 40\%$$

$$Conf = \frac{Sup\ (t_{Bread} \cap t_{Butter})}{Sup\ (t_{Bread})} = \frac{2}{4} = 50\%$$

Therefore, the rule "Bread → Butter" has support of 40% and confidence of 50%.

Table 2.1. An example of a transaction database

TID	Items
1	Bread, Salsa
2	Milk, Bread, Eggs, Jam
3	Bread, Butter
4	Bread, Butter, Lettuce, Salsa
5	Milk, Butter

Given user specified minimum support and minimum confidence thresholds, the association rule mining is to find the rules with support and confidence larger than the respective thresholds.

Generally, association rule mining can be described as a two-step process:

- First, find all itemsets whose support is above the predetermined minimum support. These itemsets are called the frequent itemsets or the large itemsets.
- Second, generate interesting association rules from the frequent itemsets.

Market basket analysis is just one form of association rule mining. There are different kinds of association rules besides this. Based on the types of values handled in the rule, it can be classified as either a boolean association rule or a quantitative association rule. Boolean association rules only concern the associations between the presence or absence of items. Market basket analysis belongs to boolean association rule mining. If either the antecedent or consequence of the rule contains quantitative attributes, then it is a quantitative association rule. For instance, age(X, 30…39) AND income(X, 40K…50K), Number of Cars(X, 1…3) is a quantitative association rule [4].

If the items or attributes in an association rule involve only one dimension, it is a single dimension rule, while a multi-dimensional association rule involves two or more dimensions. The above quantitative association rule is a multi-dimensional association rule. Based on the levels of abstraction involved in the rule set, it can be classified as a single-level rule or a multi-level association rule. If the rule can find associations at different levels of abstraction, then it is a multi-level association rule. For instance, we may have “Apple” and “Banana” as items in the transactions, and “Fruit” as a category name for those items at a higher abstraction level. Hence, the rule “Fruit → Juice” is a multi-level association rule.

2.3.1 Apriori algorithm

Apriori was the first scalable algorithm designed for association rule mining algorithm. It has been presented in [2] for the first time. The Apriori algorithm searches for large itemsets during its initial database pass and uses its result as the seed for discovering other large datasets during subsequent passes. Rules having a support level above the minimum are called large or frequent itemsets and those below are called small itemsets [4,3].

The algorithm can be used for both, finding frequent itemset and also deriving association rules from them. To represent the algorithm, the notation in Table 2.2 will be used.

Table 2.2: Notation [2]

k -itemset	An Itemset having *k* items
L_k	Set of large *k* -itemsets (those with minimum support). Each member of this set has two fields: i) itemset and ii) support count.
C_k	Set of candidate *k* -itemsets (potentially large itemsets). Each member of this set has two fields: i) itemset and ii) support count.

The Apriori algorithm is given in Figure 2.3. Firstly, the database is passed over in order to count the occurrences of single elements. If a single element has a support value that is below the defined minimum support, it does not have to be considered anymore because it hence can never be part of a large itemset. A subsequent pass *k* consists of two phases:

1- The discovered large itemsets of pass $k-1$, i.e. the sets L_{k-1} , are used to generate the candidate itemsets, C_k for the current pass.

2- The database is scanned once more in order to determine the support for the candidate itemsets C_k . If the support is above the minimum support, the candidates will be added to the large itemsets. Discovering the right candidates is crucial in order to prevent a long counting duration.

```
$L_1$ = { large 1-itemsets } ;
for ( k = 2; $L_{k-1} \neq 0$ ; k++ ) do begin
     $C_k$ = apriori-gen ($L_{k-1}$) ; / / New candidates
     for all transactions $t \in D$ do begin
       $C_t$ = subset ($C_k$, t) ; / / Candidates contained in t
       for all candidates $c \in C_t$ do
       c.count++ ;
     end
     $L_k = \{c \in C_k \mid c.count \geq minsup\}$
end
Answer = $\cup_k L_k$ ;
```

Figure 2.3: Apriori Algorithm [2].

The apriori-gen function takes the large itemsets of the previous iteration as an input. Then the Apriori can be divided into two major steps: join and prune. Explain this tow steps with more details:

Join Step

In this step, k-itemset are used to form (k+1)-itemset. For example, after one full scan of the dataset, the algorithm first finds the set of frequent 1-itemset L_1 . By joining L_1 with itself, L_2 can be found and so forth.

Denote l_i as the ith itemset in L_{k-1}; $l_i[j]$ as the jth item in l_i. The two itemset l_i and l_j is joinable only if their first (k-2) items are in common, that is: (li[1] = lj[1]) Λ (li[2] = lj[2]) Λ …. Λ (li[k-2] = lj[k-2]) Λ (li[k-1]≠lj[k-1]). Where Λ represents a logical AND. This ensures that no duplicate itemset L_k is generated. The resulting candidate itemset will be li[1] li[2]…. Li[k-1] lj[k-1].

Prune Step

By joining L_{k-1} to itself, we can get candidate set C_k. To find the frequency of itemsets in C_k will involve heavy computation because C_k can become huge. By using the closure property, if any (k-1) subset of an itemset C_k is not in L_{k-1}, then that candidate itemset cannot be frequent either and therefore can be removed from C_k. This subset pruning can be implemented by maintaining a hash tree of all frequent itemsets. The Apriori algorithm terminates when there are no frequent k-itemsets [21].

2.3.2 Hash Tree Implementation

The Apriori algorithm is implemented in the form of a hash tree in [2]. The hash tree includes two types of nodes: leaf nodes and interior nodes. The itemsets are stored in the leaf node. In an interior node, each bucket of the hash table points to another node. The root of the hash tree is defined to be at depth 1, an interior node at depth d points to nodes at depth $d+1$. When a new itemset is inserted into the hash tree, the algorithm traverses from the root level to the leaf level. At depth d, a hash function is applied to the d^{th} item of the itemset. After the number of itemsets stored in the leaf node reaches to the specified limit, the leaf node is converted to an interior node.

If it is at a leaf node, the frequency of the itemsets in all the transactions is counted and recorded. If it is at an interior node, and item i is hashed, then every item that comes after i will be hashed. This procedure will be recursively applied to the node in the corresponding bucket.

Suppose there is a simple transaction database as shown in Figure 2.4. From the figure it has four transactions and each transaction contains a series of items among {Bread, Milk, Jam, Butter, Eggs}. By scanning the database, we form the 1-item candidate set ({Bread}, {Milk}, {Jam},{Butter},{Eggs}) and store them into a 1-item hash tree based on the given hash function. Assume the maximum itemsets that one leaf node can hold is two, then obviously one leaf node cannot hold all the 1-item candidate sets. Further, assume each internal node has three hash buckets. According to the hash function, if the d^{th} element of the transaction is {Bread} or {Milk}, the itemset goes to the left branch; if the d^{th} element of the transaction is {Jam} or {Butter}, then the itemset goes to the middle branch; similarly, {Eggs} goes to the right branch. The value in the parenthesis is the frequency count of each item. Each itemset stored in the hash tree is compared with the transaction to be scanned. If the transaction contains the itemset, then the count value is increased by1. This repeats for each transaction (record) in the dataset.

In Figure 2.4, we assume minimum support value is two. By applying the closure property, we can remove item {Eggs} from the 1-item candidate set because its frequency count is one. Thus, we get the 1-item frequent set ({Bread},{Milk},{Jam},{Butter}). By joining the 1-item frequent set, we can get the 2-item candidate set: ({Bread, Milk}, {Bread, Jam}, {Bread, Butter}, {Milk, Jam}, {Milk, Butter}, {Jam, Butter}). By examining the 2-itemset hash tree and frequency count value of each itemset, we know only {Bread, Milk}, {Milk, Jam}, {Milk, Butter} remains in the 2-item frequent set.

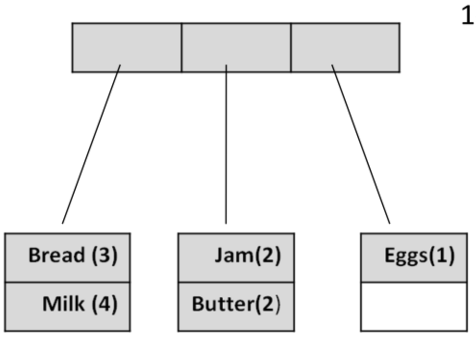
TID	Transaction
1	Bread, Milk
2	Bread, Milk, Jam
3	Bread, Milk, Butter, Eggs
4	Milk, Jam, Butter

Hash function

If dth elements = Bread, Milk	then left branch
If dth elements = Jam, Butter	then middle branch
If dth elements = Eggs	then left branch

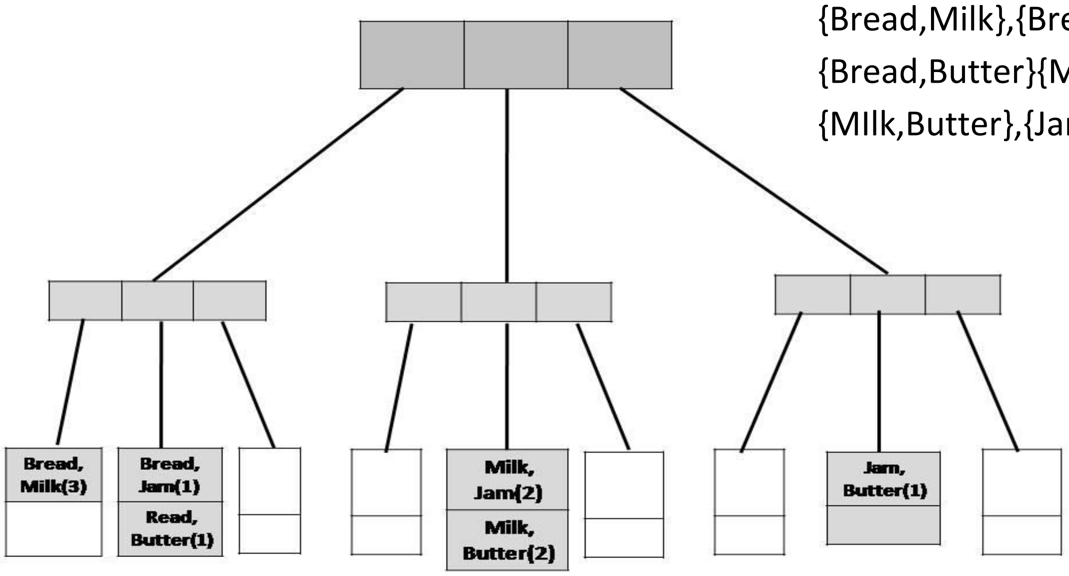
Minimum support = 2

1-itemset hash tree



1-item frequentset:{Bread},{Milk},{Jam},{Butter}

2-itemset hash tree



2-items candidate set:
{Bread,Milk},{Bread,Jam},
{Bread,Butter}{Milk,Jam},
{MIlk,Butter},{Jam,Butter}

Figure 2.4. Hash tree illustration diagram

Chapter Three

The Proposed Strategy

3.1 Overview

The importance of associative classification is to enable the end users to comprehend, interpret, and validate the huge amount of discovered rules. To achieve this, here we developed a method for automatically analyzing, organizing and managing the discovered rules, so that the end users can obtain useful information from them.

In this study the associative classification process involves two stages and it is illustrated in figure 3.1: in first stage, the data is preprocess by using data cleaning, data transformation and data reduction to make data easier to mine for knowledge. In second stage, we develop an effective data post-analysis system to solve the problem of rules number. Post-analysis system include data pruning to prune the redundant itemsets, data clustering to groping the itemsets into coherent itemsets clusters for a better organization and data summarization to summarize each itemsets cluster.

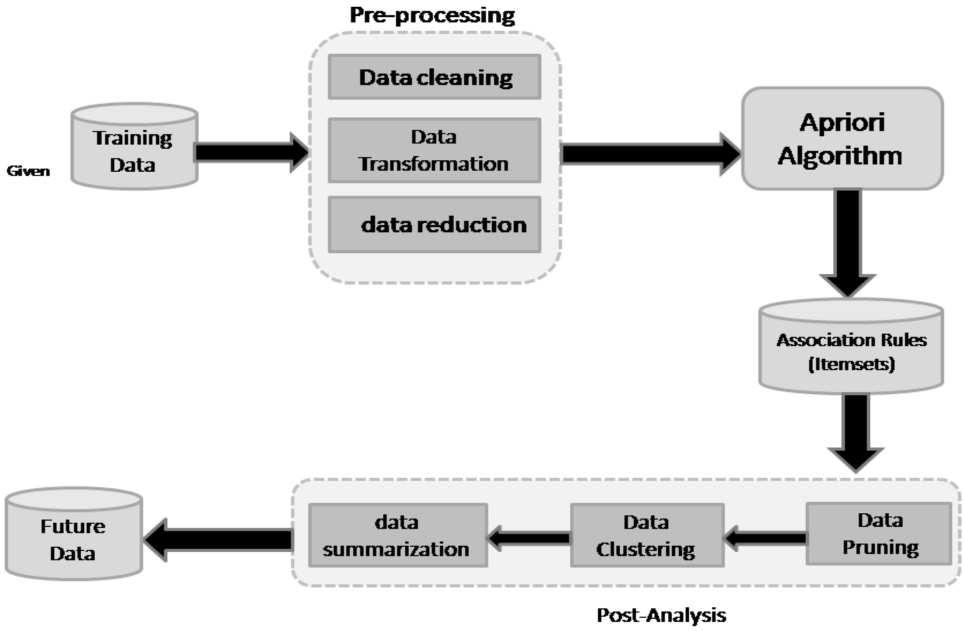


Figure 3.1: the proposed associative classification process

This chapter introduces the developed system that we proposed to solve the problem of large number of rules in association rule mining. The data preprocess procedures are explained in Section 3.2. Section 3.3 presents the post-analysis system that used in this study. The proposed algorithm introduce in section 3.4.

3.2 Data Pre-processing for Classification

In order to improve the classification ability and the efficiency of the classification process, some data preprocessing procedures are required. This includes data cleaning, data transformation and data reduction. We will briefly introduce them here.

3.2.1 Data Cleaning

Data cleaning is a process used to determine inaccurate, incomplete or unreasonable data and then improving the quality through correction of detected errors and omissions. Data cleaning generally refers to removing noise and identifying anomalies of the data. The purpose of the data cleaning is centered around improving the quality of data to make them fit for use by users through reducing errors in the data and improving their documentation and presentation [21]. There are many methods using in data cleaning such as missing values, noisy data and inconsistent data, but in this study we use the missing values only to deal with empty fields, as we shall see.

Missing values

Many tuples have no recorded value for several attributes, How can you go about filling in the missing values for this attributes? Let's look at the following methods [21]:

1- Fill in the missing value manually: In general, this approach is time-consuming and may not be feasible given a large dataset with many missing values. For example in the adualt dataset in the native-country attribute, we fill the missing values by United-States because more than 90% of values are United-States.

2- Use a global constant to fill in the missing value: Replace all missing attribute values by the same constant, such as a label like "Unknown", or "∞" or "?". For example the missing values in the heart dataset replace by "?" because the number of missing values are very large cannot fill manually.

3- Use the attribute mean to fill in the missing value: For example, the adualt dataset in hours-per-week attribute we fill the missing values by the average of hours (40).

3.2.2 Data Transformation

In data transformation, the data are transformed or consolidated into forms appropriate for mining. Data transformation can involve the following [21]:

1- **Normalization**: Normalization is one of the most common techniques for data transformation. It scales all attribute values for a given attribute so that they fall within a small specified range, such as from –1.0 to 1.0. In this study we used normalization to reduce noise and this lead to increase accuracy, for example in the hepatitis dataset in the bilirubin attribute we fall the values in 0.00 to 1.99 ranges.

2- **Generalization:** of the data, where low level or 'primitive' (raw) data are replaced by higher level concepts through the use of concept hierarchies. For example, categorical attributes, like street, can be generalized to higher level concepts, like city or county.

3.2.3 Data reduction

Techniques can be applied to obtain a reduced representation of the dataset that is much smaller in volume, yet closely maintains the integrity of the original data. That is, mining on the reduced dataset should be more efficient yet produce the same (or almost the same) analytical results. In this study we use data compression Strategy for data reduction [17]:

Data compression: where encoding mechanisms are used to reduce the dataset size. For example in the iris dataset in the class attribute we replace the Iris-setosa value by c1, the Iris-versicolor value by c2 and Iris-virginica value by c3.

3.3 Post-Analysis

After the data is pre-process, the large number of association rules extraction by the apriori algorithm. In this study we developed an integrated data post-analysis to solve the number of association rules problem. It consists of three major components data pruning removes an interesting and/or redundant itemsets , data clustering groups similar itemsets into cluster and data summarization builds a representative summary of the itemsets . In section 3.3.1, an effective data pruning method is introduced which allows the users to control the tradeoff between information loss after pruning and the number of itemsets pruned. In section 3.3.2, the data clustering component is introduced. In section 3.3.3, an effective data summarization method is introduced which automatically generates an informative yet concise summary for a data cluster.

3.3.1 Data Pruning

It is common to prune redundant data before conducting other post-analyses such as clustering and summarization. Today, most of the pruning algorithms are based on sample matching. Tow common pruning techniques are closed itemset pruning [38,53,39] and maximal itemset pruning [11,20], both of them are based on sample matching. A frequent itemset X is called maximal if it is not a subset of any other frequent itemsets. A frequent itemset X is called closed if none of its immediate superset has exactly the same support count as X [44].

In Table 3.1, the 3 frequent itemsets B_2, A_1 and A_1B_2 are reduced to one frequent closed itemset A_1B_2. By the same token, the 14 itemsets 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.

Table 3.1. An example of frequent itemsets, closed itemsets and maximal

Frequent itemsets	Closed itemsets	Maximal itemsets
B_2 , A_1 , A_1B_2	A_1B_2	
F_5 , D_4 , F_5D_4	F_5D_4	
C_3 , E_3 , B_2C_3 , A_1C_3 , E_3F_5 , E_3D_4 , $A_1B_2C_3$, $E_3D_4F_5$	$A_1B_2C_3$, $E_3D_4F_5$	$A_1B_2C_3$, $E_3D_4F_5$

Note that all maximal frequent itemsets are closed because none of the maximal frequent itemsets can have the same support count as their immediate supersets. The relationships among frequent, maximal frequent and closed frequent itemsets are shown in figure 3.2 [44].

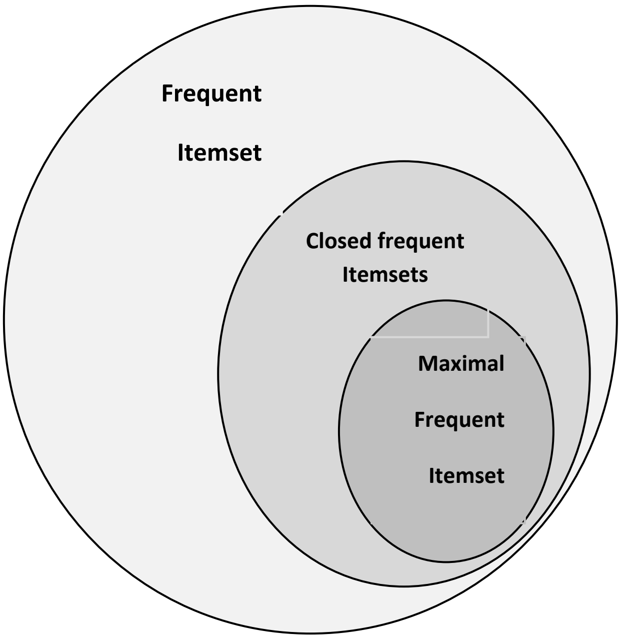


Figure 3.2: Relationships frequent, maximal frequent and closed frequent itemsets[44]

In this study we used the classical maximal itemsets pruning to prune the itemsets. The maximal itemsets pruning allows the maximal amount of information loss under certain conditions to allowing the user to control the tradeoff between information loss and the number of itemsets being pruned. We used "An Efficient Algorithm for Mining Maximal Frequent Item Sets" [54] for maximal itemsets pruning and the next example to explain the work of the algorithm.

Let us consider the support to be 3. In such case, the itemsets I1, I2, I3, I4, and I5 are frequent itemsets (Table 3.2). So, only I1, I2, I3 itemsets will be considered to next level because have items equal or more than support.

Table 3.2: The transactional database D in first level of data pruning

Itemset	Items
I1	m1, m4, m5, m7, m8, m9
I2	m1, m2, m3, m4, m6, m8, m9
I3	m3, m5, m6, m7, m8, m9
I4	m2, m4
I5	m1, m8

Now in the second level, intersection of items of all permutations of the frequent itemsets is taken. In the second level, the itemsets {I1, I2}, {I1, I3}, {I2, I3} are frequent itemsets. Now, total frequent itemsets are {I1, I2, I3, {I1,I2}, {I1,I3}, {I2, I3}}. From these, maximally frequent itemsets are {I1, I2}, {I1, I3}, {I2, I3}. This is shown in Table 3.3.

Table 3.3: The transactional database D in second level of data pruning

Itemset	Items
I1 , I2	m1, m4, m8, m9
I1 , I3	m5, m7, m8, m9
I2 , I3	m3, m6, m8, m9

The third level itemset {I1, I2, I3} has only 2 items {m8, m9}. So, this is not even frequent itemset. So, the frequent itemsets and maximally frequent itemsets obtained from the second level are the final result. It can observed that the number of levels increase if the support is less. If we increase the support, then number of levels decreases and so as the time to find MFI decreases. The algorithm is given in Figure 3.3.

```
Input
   D: a database of transactions
  Min_sup: the minimum threshold support

Output
   M:  MFI (Maximal Frequent Itemsets) in D

Method:
     1.  Count the support of each item in a given set of transactions
     2.  if support(itemset) > = min_sup then
     3.  FI[i] = itemset;
     4.  find the intersection of the transactions of each items in each
         permutation
     5.  count the support of each FI
     6.  MFI = FI – subset of FI
     7.  go to 2
     8.  return MFI
```

Figure 3.3: An Efficient Algorithm for Mining Maximal Frequent ItemSets[54].

3.3.2 Data Clustering

Data clustering is a process of grouping the data into classes or clusters, so that objects within a cluster have high similarity in comparison to one another but are very dissimilar to objects in other clusters [21]. It is similar to the classification analysis regarding the purpose and outcome. The difference is that classification requires labeling of the datasets, which is used to model each group, while cluster analysis processes reversely: it partitions the set of data into groups based on the similarity, and then assigns labels to each group according to its characteristics. Therefore cluster analysis is sometimes referred to as unsupervised classification.

Data clustering is an important process in data mining, because in practice, it is quite common that the class label of each object is not known, or it will be very costly to do the labeling for a large set of data. Clustering is an effective way find distinguishing groups. It can produce meaningful insights from the results alone, and it can also be used as a pre-processing step to reduce the dimensions of datasets by generalizing them into simple classes. Many clustering algorithms have been developed and studied in the literature [6, 7, 5].It is generally accepted that there are broadly two types of clustering approaches: hierarchical methods and partitional methods.

Hierarchical methods build a hierarchy of clusters. They can be either agglomerative or divisive. Agglomerative algorithm is a bottom-up approach, begins at each object to form a separate group, and continuously merges the objects or groups to one another, until all the groups are merged into the top of the hierarchy. Divisive algorithm, on the contrary, starts from the root, by having all the objects in the same cluster, and split it up into smaller clusters. It is also called top-down method [21].

Partitional methods attempt to construct *k* partitions of the data, where each partition represents a cluster. The first step of the partitional method is to decide the best value of *k*–number of clusters. Ways to determine that include: simple guesswork, application requirement, or selecting the best one from number of trials with different *k* based on certain criteria [44].

Hierarchical 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 clustering is not scalable. In contrast, *k*-means clustering is fast and scalable since it only produces a partition rather than the entire hierarchy [18]. Therefore in our study we used the k-means algorithm to clustering the data.

K-means algorithm is one of the most well-known and popular square-error based partitioning methods. The process of K-means algorithm is: The dataset is partitioned into K clusters and the data points are randomly assigned to the clusters resulting in clusters that have roughly the same number of data points. For each data point: first, calculate the distance from the data point to each cluster. Second, if the data point is closest to its own cluster, leave it where it is. If the data point is not closest to its own cluster, move it into the closest cluster. Repeat the previous step until a complete pass through all the data points' results in no data point moving from one cluster to another. At this point the clusters are stable and the clustering process ends the next example to explain the work of the algorithm.

Let objects are {2,4,10,12,3,20,30,11,25}, and let K = 2. Randomly select 5, 2 as centers and by using Euclidean Space we found K1={2,3} and K2= {12,20,30,11,25,4,10}. Then recomputed the clusters centers m1= 2.5, m2= 16, again by using Euclidean Space K1= {2,3,4}, K2= {10,12,20,30,11,25}. Repeat until gets these results:

m1	m2	K1	K2
3	18	{2,3,4,10}	{12,20,30,11,25}
4.75	19.6	{2,3,4,10,11,12}	{20,30,25}
7	25	{2,3,4,10,11,12}	{20,30,25}

Until there is no change in clusters, then K1= {2,3,4,10,11,12} and K2= {20,30,25}. The steps of the algorithm are given in figure 3.4:

```
Initialize $k$, $I(i)$, $i = 1,\dots, n$
Take the first $k$ $I(i)$, $i = 1,\dots, k$ as single-element pattern clusters $c(i)$
For each of the remaining $I(i)$, $i = k+1,\dots,n$ do
    Find the nearest pattern cluster $c(j)$ for $I(i)$ and merge them
End {for}
Repeat
    For $i := 1$ to $n$ do
        Find the nearest pattern cluster $c(j)$ for $I(i)$
        If $I(i)$ is not contained in $c(j)$ then
            Remove $I(i)$ from its current pattern cluster, say $c(k)$
            Merge $I$(i) and $c(j)$
        End {if}
    End {for}
Until no new remove and merge operation occurs
Return $k$ clusters $c(i)$, $i = 1,\dots, k$
```

Figure 3.4: The k-means clustering [21]

K-means is a simple and direct approach. The advantage of k-means algorithm is simplicity and efficiency. It can process large datasets because of its relatively low computational complexity. It has limitation as well. First of all, it requires the input of k–number of clusters, which is sometimes not known in advance. And the step of cluster mean value calculation makes it hard to apply to data with categorical attributes. Another disadvantage of k-means method is that the clustering result depend on the initial random assignments, therefore it often terminate at a local optimum and does not ensure a global one. The k-means method is also sensitive to noise and outliers because they can significantly affect the mean value of the clusters.

3.3.3 Data Summarization

After data pruning and data clustering the remaining data are summarized. The objective of data summarization is to obtain a small subset of rules that are representative to other rules. Data summarization can be considered as a very aggressive pruning method where most rules, except for a few representative rules, are pruned. In practice, rules are not really pruned. Instead, a few representative rules are selected to summarize the other rules [30,31]. The RuleCover method [46] used in this study 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 3.5, the data-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 rules). 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 3.5. 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}.

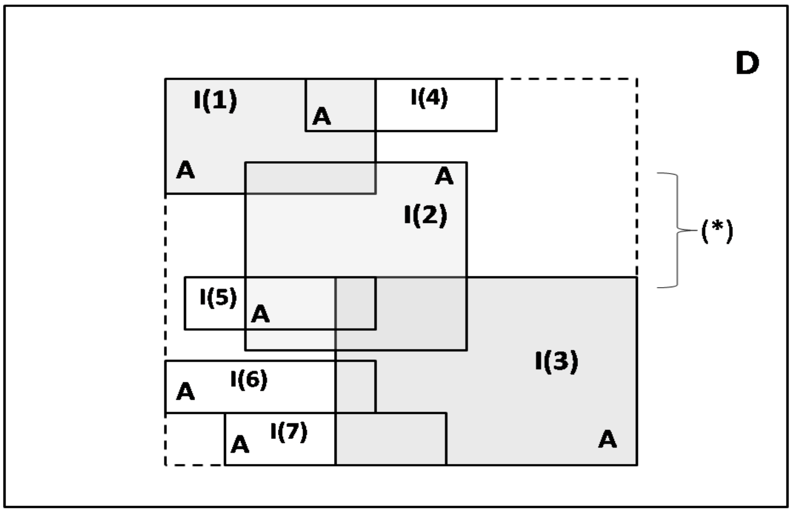


Figure 3.5: RuleCover example

The major steps in the RuleCover algorithm are shown in figure 3.6 , where Γ is the original set of rules, *m*(*i*) is the set of matched samples of the rule X_i and ε is the user-specified constant for specifying the desired portion of rules covered by the rule cover Δ:

```
Initialize ε, $\Gamma := \{ X_i \mid i = 1, \ldots, n\}$, $m(i), i = 1, \ldots, n$
Δ: = ø
$u := \bigcup_{i=1}^{n} m(i)$
$O = | u | \times (1 - \varepsilon)$
For $i$: = 1 to $n$ do
    $u_i := m(i)$
End {for}
Repeat
    Choose $X_i \in \Gamma$ so that $| u_i |$ is largest
    $\Delta : = \Delta \cup Xi$
    $\Gamma : = \Gamma \setminus X_i$
    For all $X_i \in \Gamma$ do
        $u_i := u_i \setminus m(i)$
    End {for}
    $u := u \setminus m(i)$
Until $| u | \leq o$
Return the rule cover Δ
```

Figure 3.6: RuleCove algorithm[46]

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 3.5, ε is 100%, indicating that the rule cover will cover 100% of samples in *I*(1, 2, …, 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 lose ε so that rules covering small uncovered samples are not included in the rule cover. For example, in Figure 3.5, 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 (*).Given a set of rules, RuleCover prunes most rules and only a few rules that cover the largest portions of samples are retained. RuleCover could be a good method for summarizing patterns. For example, in Figure 3.5, 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).

3.4 Hybrid Strategy Algorithm

In this section, we proposed the Hybrid Strategy algorithm to design the associative classification. In this algorithm, the classic classification reasoning methods as motioned in above sections is applied to classify a new rule. The steps of Hybrid Strategy algorithm are given in figure 3.7:

```
Input
   D: a database of transactions
  Min_sup: the minimum threshold support

Method:
   1.  Count the support of each itemset
   2.  If support(itemset) > = min_sup then
   3.  FI[i] = itemset
   4.  Find the intersection of the itemsets
   5.  Count the support of each FI
   6.  MFI = FI – subset of FI
   7.  Go to 2 until return MFI
   8.  Select k - MFI from the dataset to form the initial k clusters
   9.  Assign each MFI to the nearest cluster center
   10. Input cluster of itemsets into $\Gamma$
   11. Itemset cover $\Delta$ initialized to empty set
   12. $S1$ used to store itemsets not matched by itemsets in $\Delta$
   13. Sets $Si$ contain those itemsets in $S1$ that are matched by the rule
       $X \rightarrow Y$
   14. Iteratively the itemsets in $\Gamma$ that matches most of the itemsets in $S1$ is
       moved from $\Gamma$ to $\Delta$
   15. Go to  12 until all itemsets matched by $\Gamma$ matched by $\Delta$
   16. Repeated steps 12-15 until all clusters finished
   17. Return the rule cover $\Delta$
```

Figure 3.7: The Hybrid Strategy Algorithm

Hybrid Strategy algorithm use sample itemset matching to first, prune the frequent and redundant itemset. Then the second step grouping the output itemsets of the previous step into coherent itemsets clusters for abetter organization and the final step is summarization each itemsets cluster. The details of Hybrid Strategy algorithm is:

Line 1-7: compute support value of each itemset, and then removed each itemset that have support vale less than minimum support. Intersection these itemsets have support value equal or more than minimum support. Recomputed support value of each itemset after intersection and compare with minimum support and remove itemset less than minimum support. Repeat above steps until to get the Maximal Frequent Itemsets.

Line 8 and 9: select K-MFI from the output of above steps to initial K clusters. By using sample itemset matched, each MFI assign to the nearest cluster center. If the MFI is closet to its own cluster, leave it where it is. If the MFI is not closet to its own cluster, move it into the closet cluster. Repeat the previous step until a complete pass through all the MFIs results in on MFI moving from one cluster to another.

Line 12-17: The output cluster itemsets from above clustering part input in Γ. Itemset cover Δ is initialized to an empty set. The set *S1* is used to store those samples that are not matched by itemsets in Δ whereas the sets *Si* stores those samples in *S1* that are matched by the rule $X \longrightarrow Y$. Iteratively, the rule in Γ that matches most of the samples in *S1* is moved from the Γ to the itemset cover Δ. The rules matched by this rule are removed from *S1*. This is repeated until the itemsets in Δ cover at least most of the itemsets. Repeat the last part with each cluster.

3.5 Test Data

Most of association rule mining methods suffer from the problem of rendering too many rules. Ironically, the number of rules produced can be much more than the number of data samples. For instance, the Wine dataset from UCI [37] 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 and has 178 samples, is given in appendix A.

The mining of association rules is actually a two steps: first is the association rules extraction by the apriori algorithm; and the second step is the evaluation of the rules quality. In the first step the association rule mining produces huge number of itemsets or association rules, for example the wine dataset has only 178 samples but association rule mining can produces over 20000 itemsets or association rules. Hence, it is very time consuming and expensive to analyze the overwhelming number of discovered rules. Therefore, in post-analysis we used three major components to reduce these itemsets or association rules. First, we used maximal itemset pruning to prune the redundant or frequent itemsets. Where 44% of itemests or association rules are reduces, see appendix B.

However, the number of remaining itemsets or association rules could still be more than 10000. Second, we used the data clustering to groping the remaining itemsets into classes. These itemsets are clustered into 10 clusters by using K-mean clustering, on average; each cluster could contain 1000 itemsets, see appendix C. Third, after data pruning and data clustering, we used RuleCover algorithm to summarize the itemsets in each cluster. RuleCover prunes most association rules and only a few rules that cover the largest portions of samples are retained. RuleCover is most useful when applying to a small group of rules since it prunes aggressively and only a few rules 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. Hence, the RuleCover prunes more than 70% of rules in each group, see appendix D. These three algorithms are use together in Hybrid Strategy algorithm to reduce the number of association rules that results from association rule mining, see appendix E.

Chapter four

Implementation and Experimental Results

4.1 Overview

In the previous chapter, we presented an integrated and flexible frame work for data post-analysis. The proposed strategy including data pruning, data clustering and data summarization to support effective analysis and interpretation of the discovered rules. The hybrid Strategy algorithm used to reduce the number of rules that generates by associative classification with high accuracy classification. In order to evaluate the classification performance of the approach proposed in chapter three, in this chapter, we will present the implementation of this algorithm first, then present the experimental results of this algorithm and compare these results with the result of the previous algorithms.

4.2 Implementation

The proposed algorithm implementation has been done using java language. In this section, we have presented the implementation of our algorithm. Then hybrid strategy implementation will be discussed and a few screen shots will be shown. The discussion will only highlight the key differences and areas that are unique to the java implementation and the hybrid strategy demo application.

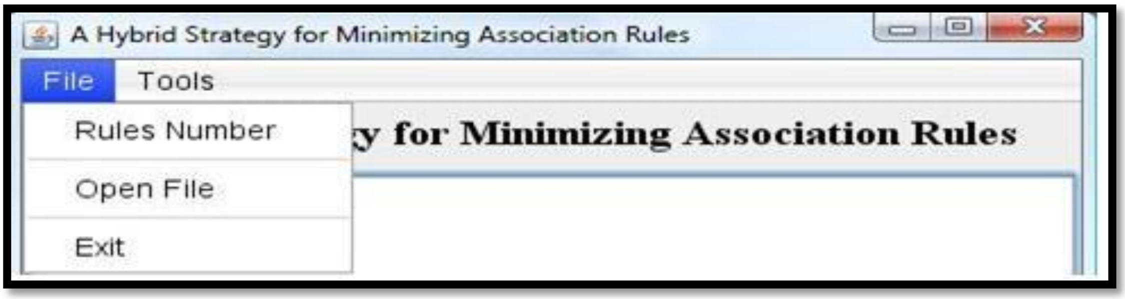
4.2.1 The User Interface

The user interface of the system has been designed on the description of the proposed algorithm. Building the user interface in java is difficult since everything need programming. The basic user interface component is the **form**. It contains all the controls that form the shape of each screen. During the development phase, built in controls (that provided by the programming language itself) have been used to form up the final shape of the system screens. Those include text boxes, buttons, labels, panels and tapped pane. The user interface parts are menu bar and the result tap.

4.2.1.1 Menu Bar

A menu bar is a horizontal strip that consist lists of available menus for a certain program. Menu bar is a part of the user interface and most menu bar have the standard File, Edit View menus listed first. In our implementation the menu bar is consist the following menus:

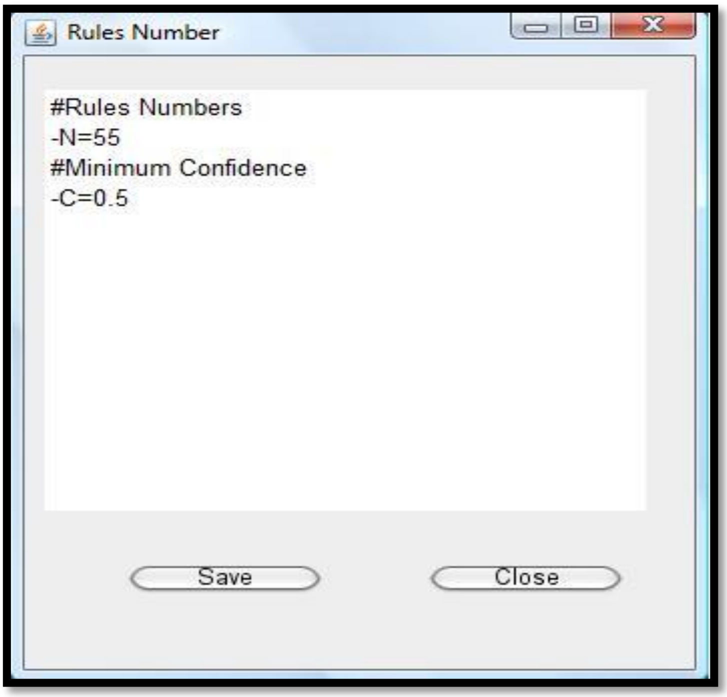
1- File menu: it consist rules number, open file and exit, as shown in figure 4.1.



A Hybrid Strategy for Minimizing Association Rules
File
Tools
Rules Number
Open File
Exit
y for Minimizing Association Rules

Figure 4.1: The file menu

- **Rules Number:** rules number window consist of two variables N (Rules Number) is the number of best rules are found and C (Minimum Confidence) is the percentage of records in database containing x-item that also contain y-item, see figure 4.2.



```
#Rules Numbers
-N=55
#Minimum Confidence
-C=0.5
```

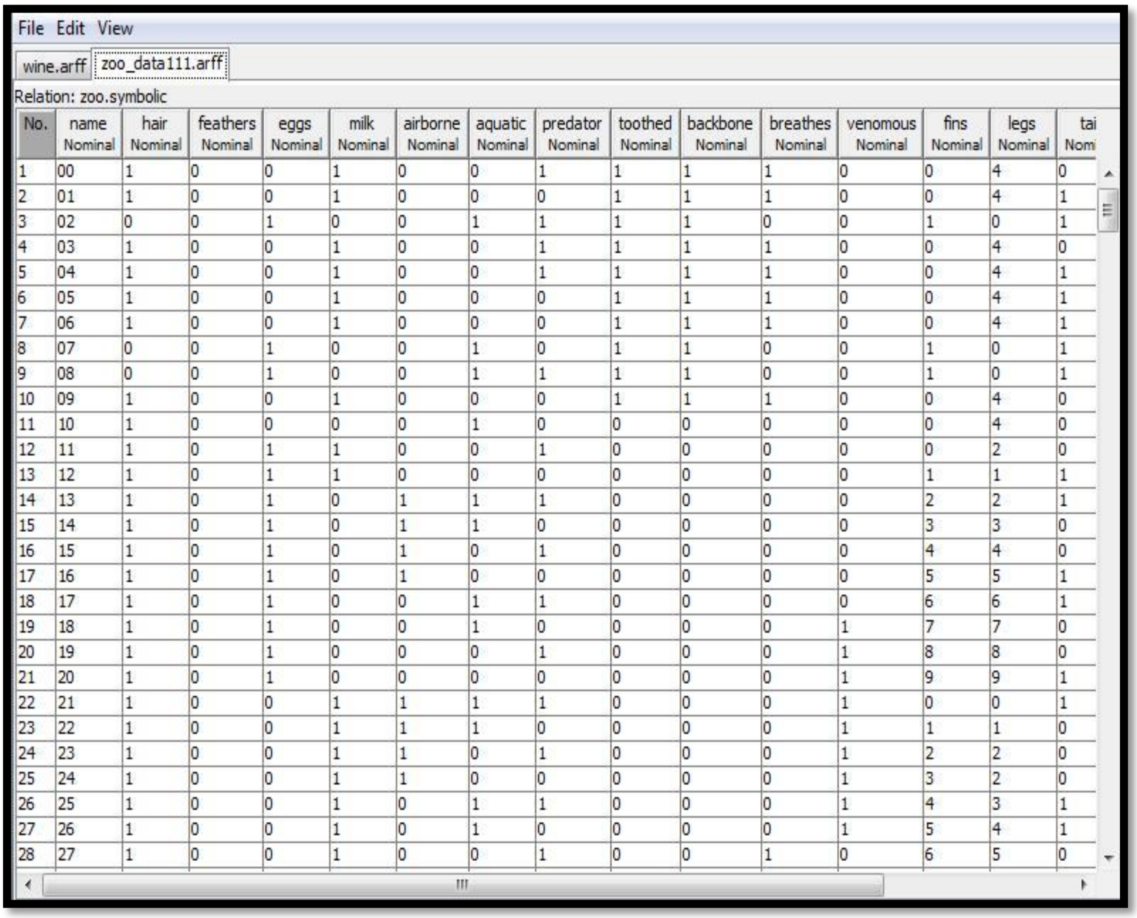
Figure 4.2: the rules number window

- **Open File:** in java, the user must supply the path names to the source folder along with the libraries used by java application. The user can browse to these folders (via brows buttons) or type the path names into the provided edit fields. Open file button open files in a variety of ARFF format. ARFF files typically have a .arff extension.

- **Exit** close the program

2- Tools menu: it consist open file viewer and accuracy**.**

- **Open File Viewer:** is a tool for viewing datasets files in a tabular format. The advantage of this kind of display over the file representation is that attribute name, type and data are directly associated in columns and not separated in definition and data part. But the viewer is not only limited to viewing multiple files at once, but also provides simple editing functionality, like sorting and deleting, as shown in figure 4.3.

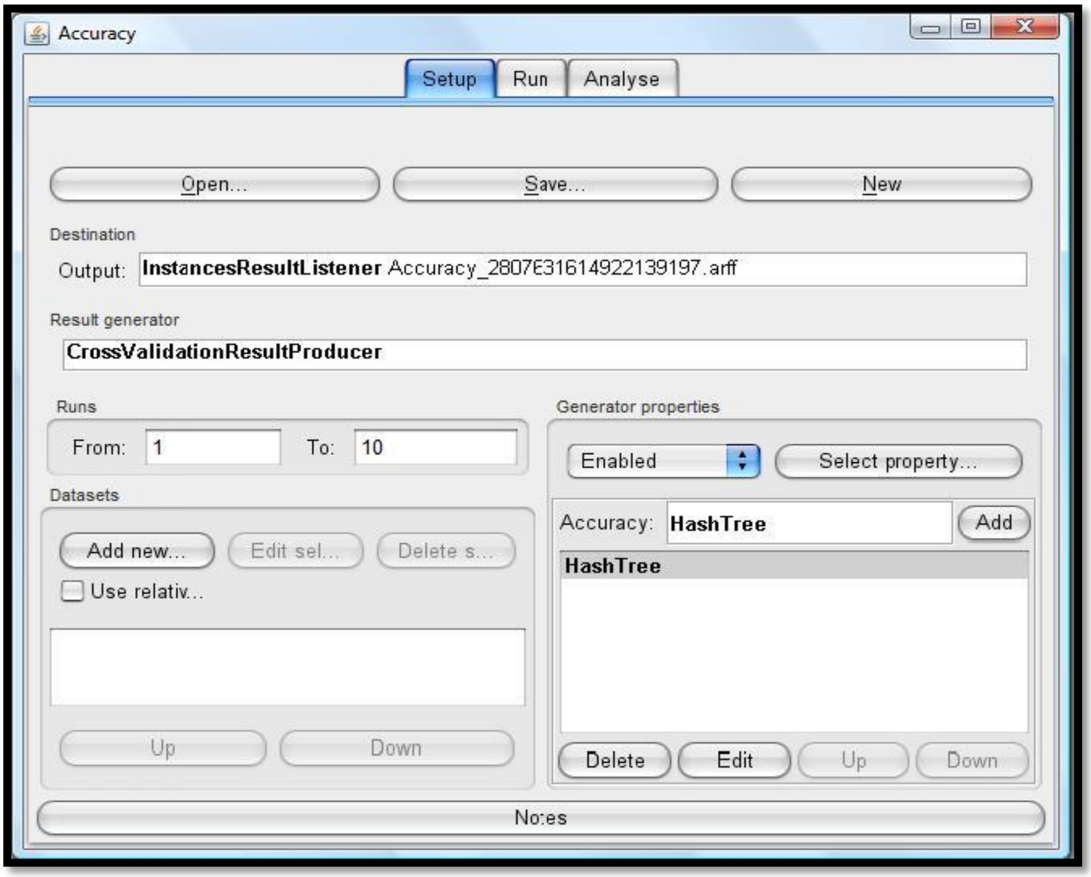


Relation: zoo.symbolic

No.	name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tai
	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nomi
1	00	1	0	0	1	0	0	1	1	1	1	0	0	4	0
2	01	1	0	0	1	0	0	0	1	1	1	0	0	4	1
3	02	0	0	1	0	0	1	1	1	1	0	0	1	0	1
4	03	1	0	0	1	0	0	1	1	1	1	0	0	4	0
5	04	1	0	0	1	0	0	1	1	1	1	0	0	4	1
6	05	1	0	0	1	0	0	0	1	1	1	0	0	4	1
7	06	1	0	0	1	0	0	0	1	1	1	0	0	4	1
8	07	0	0	1	0	0	1	0	1	1	0	0	1	0	1
9	08	0	0	1	0	0	1	1	1	1	0	0	1	0	1
10	09	1	0	0	1	0	0	0	1	1	1	0	0	4	0
11	10	1	0	0	0	0	1	0	0	0	0	0	0	4	0
12	11	1	0	1	1	0	0	1	0	0	0	0	0	2	0
13	12	1	0	1	1	0	0	0	0	0	0	0	1	1	1
14	13	1	0	1	0	1	1	1	0	0	0	0	2	2	1
15	14	1	0	1	0	1	1	0	0	0	0	0	3	3	0
16	15	1	0	1	0	1	0	1	0	0	0	0	4	4	0
17	16	1	0	1	0	1	0	0	0	0	0	0	5	5	1
18	17	1	0	1	0	0	1	1	0	0	0	0	6	6	1
19	18	1	0	1	0	0	1	0	0	0	0	1	7	7	0
20	19	1	0	1	0	0	0	1	0	0	0	1	8	8	0
21	20	1	0	1	0	0	0	0	0	0	0	1	9	9	1
22	21	1	0	0	1	1	1	1	0	0	0	1	0	0	1
23	22	1	0	0	1	1	1	0	0	0	0	1	1	1	0
24	23	1	0	0	1	1	0	1	0	0	0	1	2	2	0
25	24	1	0	0	1	1	0	0	0	0	0	1	3	2	0
26	25	1	0	0	1	0	1	1	0	0	0	1	4	3	1
27	26	1	0	0	1	0	1	0	0	0	0	1	5	4	1
28	27	1	0	0	1	0	0	1	0	0	1	0	6	5	0

Figure 4.3: dataset viewer

- **Accuracy:** The accuracy environment enables the user to create, run, modify, and analyze experiments in a more convenient manner than is possible when processing the schemes individually. For example, the user can create an experiment that runs several schemes against a series of datasets and then analyze the results to determine if one of the schemes is (statistically) better than the other schemes. When the accuracy is started, the Setup tab is displayed. Chose New to initialize an experiment. This causes default parameters to be defined for the experiment, see figure 4.4.



Destination

Output: InstancesResultListener Accuracy_2807E31614922139197.arff

Result generator

CrossValidationResultProducer

Runs

From: 1 To: 10

Datasets

Generator properties

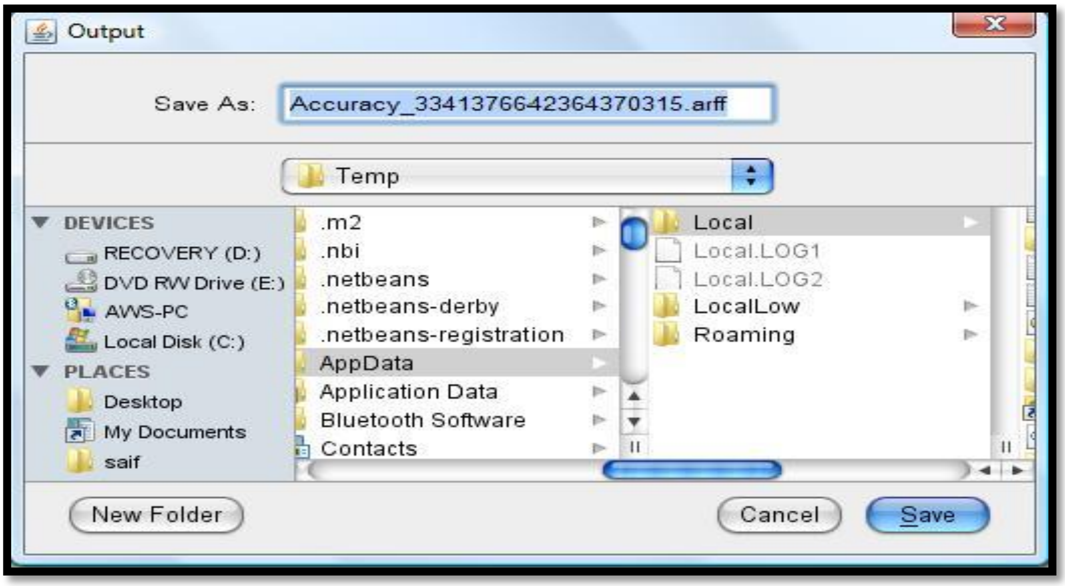
Enabled

Accuracy: HashTree

HashTree

Figure 4.4: The setup tap

To define the dataset to be processed by a scheme, first select Add new in the Datasets panel of the Setup tab to open a dialog window. To saving the results of the accuracy, must identify a dataset to which the results are to be sent. Click on the output entry to display a file selection window, as shown in figure 4.5.

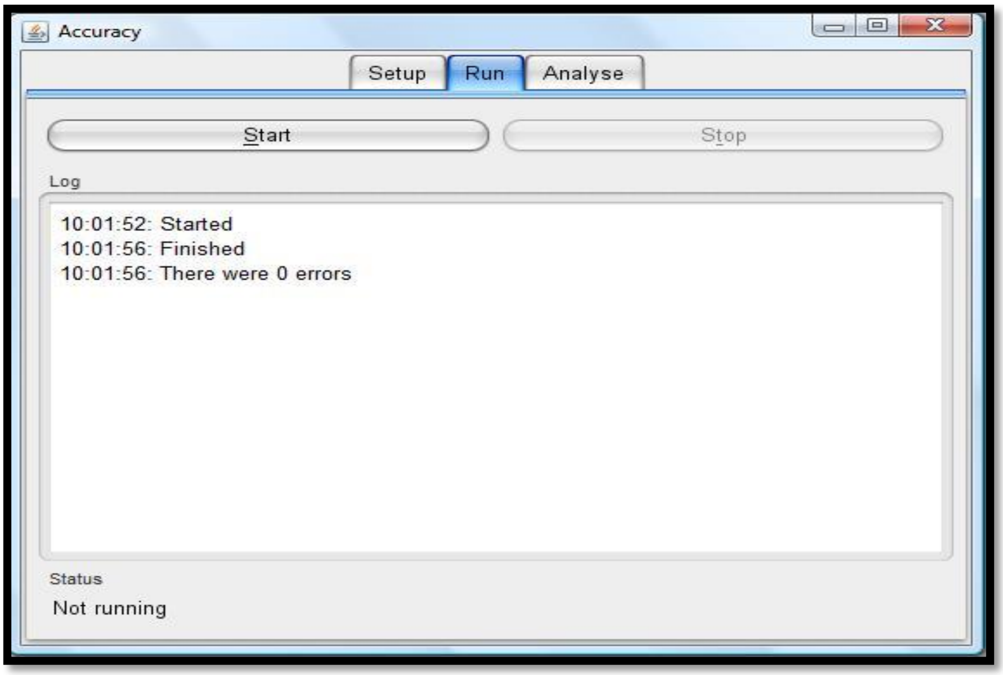


Output

Save As: Accuracy_3341376642364370315.arff

Figure 4.5: the selection window

To run the current experiment, click the Run tab at the top of the accuracy window. The current experiment performs 10 randomized train and test runs on the dataset. If the experiment was defined correctly, the 3 messages will be displayed in the Log panel. The results of the experiment are saved to the dataset accuracy.arff, see figure 4.6.

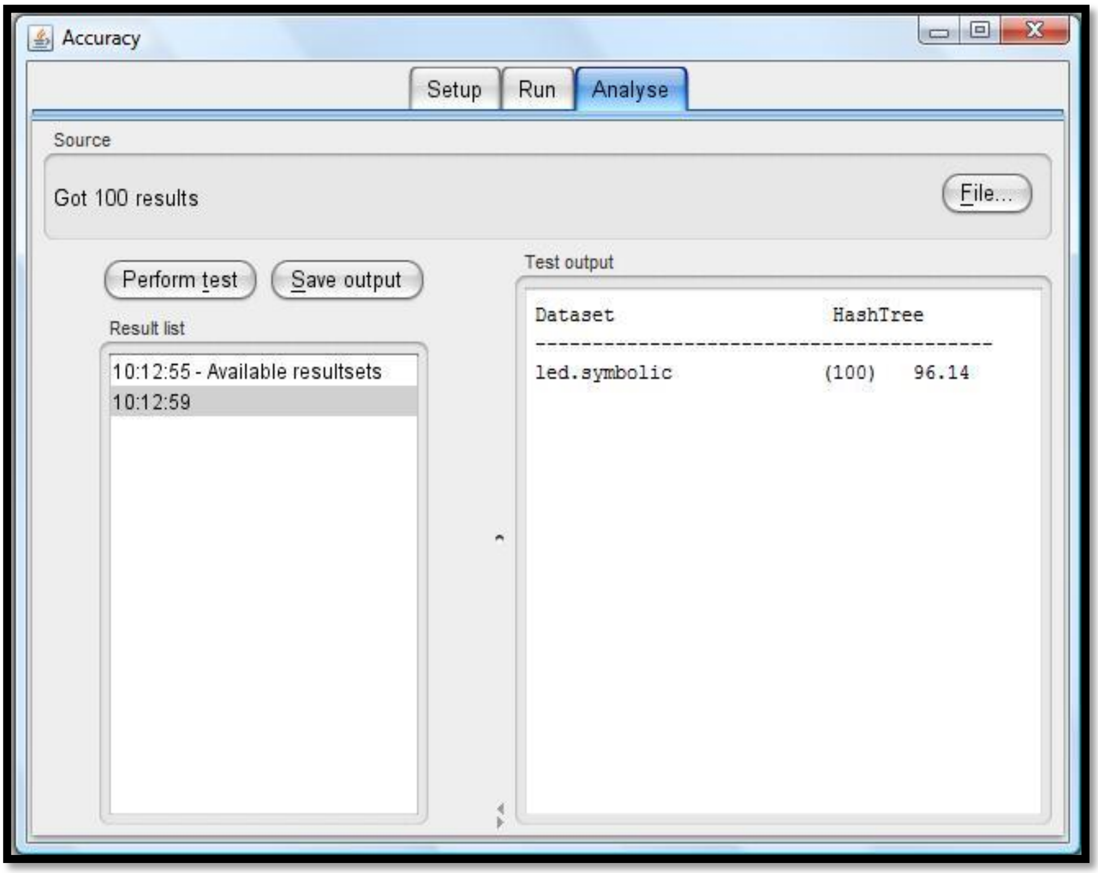


```
10:01:52: Started
10:01:56: Finished
10:01:56: There were 0 errors
```

Status
Not running

Figure 4.6: The run tab

After the experiment setup is complete, run the experiment. Then, to analyze the results, select the Analyse tab at the top of the accuracy window. The output results are run 10 items with 10-fold-cross-validation, the dataset name and accuracy percentage will be displayed in the Log panel see figure 4.7.



Got 100 results

10:12:55 - Available resultsets
10:12:59

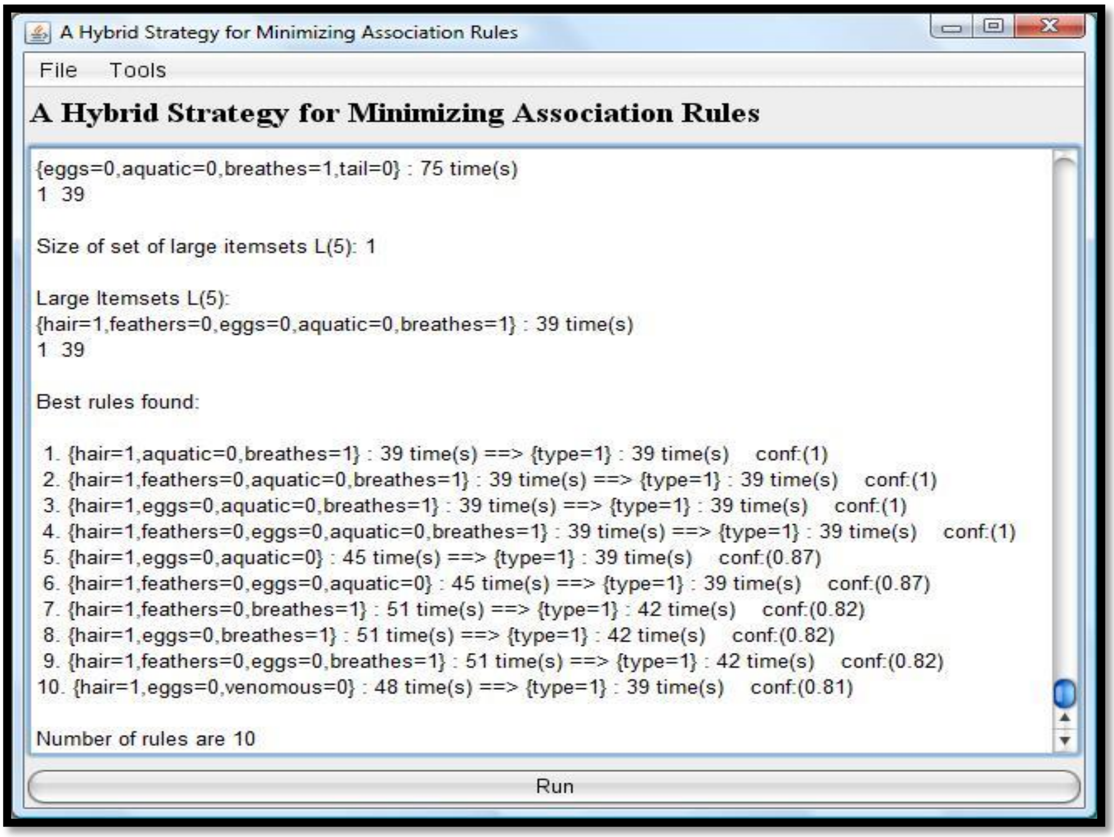
```
Dataset                    HashTree
----------------------------------------
led.symbolic               (100)   96.14
```

Figure 4.7: the analyse tap.

4.2.1.2 Result Tap

This panel displays the following:

- **Minimum support value**
- **Minimum confidence value**
- **Number of cycles performed**: is the number of iteration through the transaction. The result set contains one candidate set to begin with. Each iteration will clear and add new candidate sets to this set. When there no new candidate sets in the result set, the algorithm proceeds to the next step.
- **Association rules**: association rules display as the antecedent (is a subset of items found in sets of data), and the consequent (is an item that is in combination with the antecedent), see figure 4.8.


A Hybrid Strategy for Minimizing Association Rules
File
Tools
A Hybrid Strategy for Minimizing Association Rules
Run

```
{eggs=0,aquatic=0,breathes=1,tail=0} : 75 time(s)
1  39

Size of set of large itemsets L(5): 1

Large Itemsets L(5):
{hair=1,feathers=0,eggs=0,aquatic=0,breathes=1} : 39 time(s)
1  39

Best rules found:

 1. {hair=1,aquatic=0,breathes=1} : 39 time(s) ==> {type=1} : 39 time(s)    conf:(1)
 2. {hair=1,feathers=0,aquatic=0,breathes=1} : 39 time(s) ==> {type=1} : 39 time(s)    conf:(1)
 3. {hair=1,eggs=0,aquatic=0,breathes=1} : 39 time(s) ==> {type=1} : 39 time(s)    conf:(1)
 4. {hair=1,feathers=0,eggs=0,aquatic=0,breathes=1} : 39 time(s) ==> {type=1} : 39 time(s)    conf:(1)
 5. {hair=1,eggs=0,aquatic=0} : 45 time(s) ==> {type=1} : 39 time(s)    conf:(0.87)
 6. {hair=1,feathers=0,eggs=0,aquatic=0} : 45 time(s) ==> {type=1} : 39 time(s)    conf:(0.87)
 7. {hair=1,feathers=0,breathes=1} : 51 time(s) ==> {type=1} : 42 time(s)    conf:(0.82)
 8. {hair=1,eggs=0,breathes=1} : 51 time(s) ==> {type=1} : 42 time(s)    conf:(0.82)
 9. {hair=1,feathers=0,eggs=0,breathes=1} : 51 time(s) ==> {type=1} : 42 time(s)    conf:(0.82)
10. {hair=1,eggs=0,venomous=0} : 48 time(s) ==> {type=1} : 39 time(s)    conf:(0.81)

Number of rules are 10
```

Figure 4.8: the result tap

4.3 Experimental Results

In order to evaluate the classification performance of Hybrid Strategy approach proposed in Chapter three, we used several well-known datasets from the UCI machine learning repository [37]. We mainly use 10-fold cross validation to evaluate the classification performance. Based on this 10-fold cross validation technique, we compared our approach with other associative classification approaches such as [29,52,13,41]. Section 4.3.1 introduces relationships between the number of association rules and the minimum support value when using different datasets. Section 4.3.2 introduces relationships between the number of association rules and the minimum support values when using different minimum confidence values. Results comparison introduce in section 5.3.3 and summary of results introduce in section 5.3.4

4.3.1 Relationships Between Association Rules and Support Using DataSets

We examined the relationships between the number of association rules and the minimum support value when using different dataset. Three types of dataset, adualt dataset, heart dataset and ionosphere dataset are used. Among them, adualt dataset, heart dataset and ionosphere dataset have been commonly used in association rule mining.

Figure 4.9 shows the relationship between the number of rules and minimum support values when different datasets are used. The value of minimum support varies from 0.01 to 0.02 with a minimum confidence set at 50%. In all cases, it can be seen that the number of association rules decreases with an increase in the minimum support value. The adualt dataset generates more rules than the heart dataset and ionosphere dataset within the specified minimum support range. This is because adualt dataset has a larger number of records (more than 47000 records) that affects the number of frequent k-itemsets selected to generate the rules. The heart dataset generates more rules than the ionosphere dataset within the specified minimum support range and has the same number of records (nearly 300 records). This is because heart dataset has a small number of attributes (14 attributes) less than ionosphere dataset (34 attributes), that affects the number of frequent k-itemsets selected to generate the rules.

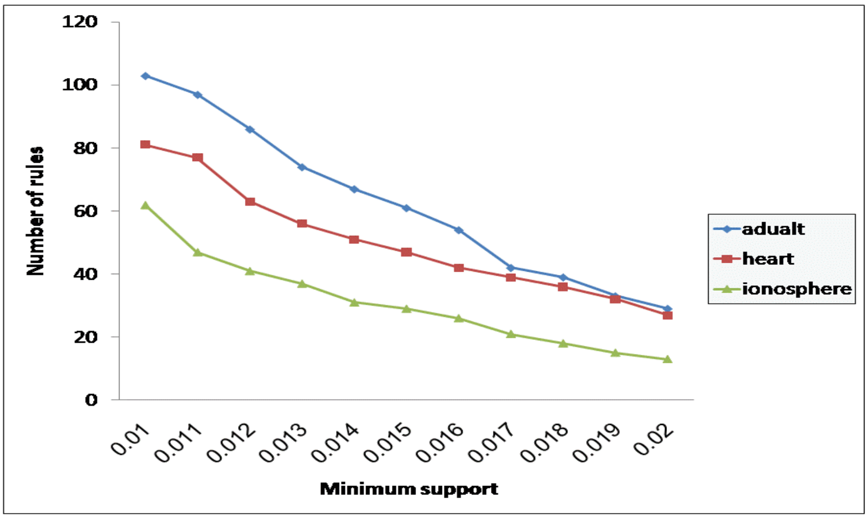


Figure 4.9.Number of rules vs. minimum support

4.3.2 Relationships Between Association Rules and Support Using Different Confidence Values

We use the hepatitus dataset to examine the relationship between the number of association rules and the minimum support values under different minimum confidence values. The results are shown in Figure 4.10. We can see that for the same minimum support value the number of association rules decreases as the minimum confidence value increases because higher confidence value reduces the number of records qualifying for evaluation. Examining Figure 4.10, we note that as the minimum support value increases the number of rules produced decreases and converges. In the figure the number of rules is essentially the same, regardless of the minimum confidence level, for a minimum support value greater than 0.004. This means that the minimum confidence value has a larger effect on the number of association rules when a smaller minimum support value is used.

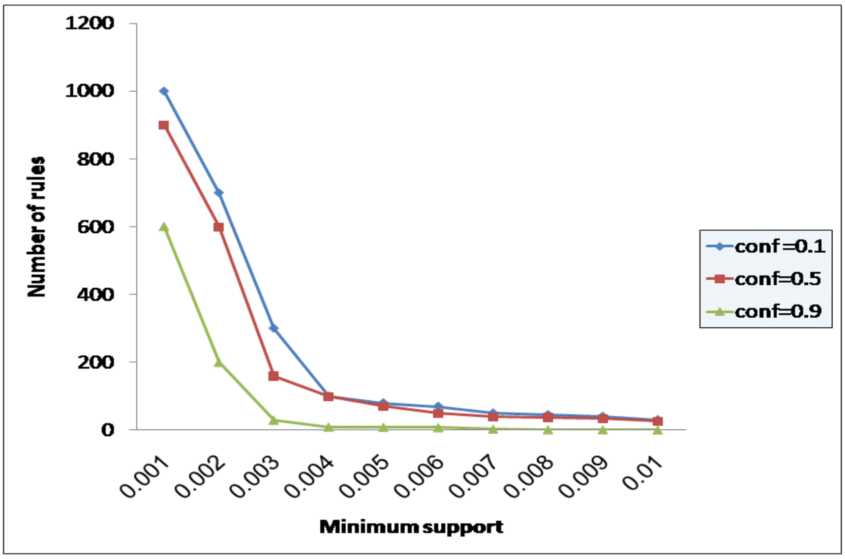


Figure 4.10.Number of rules vs. minimum support on different confidence values.

5.3.3 Results Comparison

To investigate the performance of this study, we compare the classifiers produced by Hybrid Strategy with those produced by CMAR [29], CPAR [52], TFPC [13], and Classification by Bagged Consistent Itemset Rules [41]. We used 18 datasets from

UCI Repository [37] for the purpose. The implementation of Hybrid Strategy was as a Java program, and for comparison purposes we have used our own (Java) implementations of the published algorithms for [29, 52, 13, 41]. In our experiments, minimum confidence is set to 50%. For minimum support, it is more complex. Minimum support has a strong effect on the quality of the classifier produced. If minimum support is set too high, those possible rules that cannot satisfy minimum support but with high confidences will not be included, and also the classification association rules may fail to cover all the training cases. Thus, the accuracy of the classifier suffers. In our experiments, we set minimum support to 1%. The rules number results of these experiments are shown in table 5.1.

Table 5.1. The results of rules number

DataSet	CMAR	CPAR	TFPC	HS
adult	٣٠٦٣	١٨٣	٨٢	١٠٣
anneal	٣١٩	٣٤	٦١	٤٣
breast	١٩١	١٦	٤٣	٢٠
connect	٨٢١	٨١٦	١٠٦	١٦٤
heart	٣٠٥	٥٣	١٩٣	٨١
hepatitus	٩٩	١٤	٥٨	٢٧
horse	٣٨٧	١٨	٩٩	٢٣
ionosphere	٢١٤	٢٦	٣٨٠	٤٢
iris	٦٩	١١	٢٠	١٤
led	٢٢٩	٣١	٢٩	٢٧
mushroom	١٧١	٣١	١١٦	٣٤
nursery	٤٨٧	٨٤	٣٠	١٧
Page blocks	٢٤٣	٥٦	١٩	٣١
Pen digits	٢٠٥٢	١٦٧	٣٠١٥	١٢٤
Pima indians	٥٤٠	٢٣	٣٤	١٨
waveform	١١٨٦	١١٤	٦٦٢	٨٩
wine	١١٠	١٨	١٦٤	١٧
zoo	٣١	١٩	٢٥٠	٣٣
average	**584.3**	٩٥	٢٩٨	**50.3**

CPAR produce fewer rules than other approaches in 8 cases. But the last row in the table 5.1 show the average number of association rules included in final classifiers generated, for the four methods. The HS produce fewer averages of rules than other approaches. The advantage and effectiveness of HS shows in all the cases where the data included more than 10000 records (adult, connect, nursery, pendigits), HS approach produces rules in this cases much smaller than other four approaches because the effectiveness of clustering step increase in a big dataset. The minimizing in the rules number lead to increase the speed of data analysis and reduce the processing cost.

Table 5.2 tabulates the accuracy of the classification obtained by the four methods. In all cases, the table shown is the average obtained from a 10-fold cross-validation using the full dataset. The accuracy obtained using the HS method is in most cases best than that obtained from the other methods investigated. Although the average accuracy of the HS method was more than the others, in 14 cases HS gave accuracy as high as or higher than other three methods, and only in two cases (ionosphere and wine) was it markedly worse than three methods.

Table 5.2. The results of accuracy

DataSet	CMAR	CPAR	TFPC	HS
adult	72.2	76.7	76.1	99.1
anneal	83.5	90.2	84.6	89.9
breast	85.2	94.8	95.9	89.9
connect	66.9	54.3	65.9	68.7
heart	56.8	51.1	57.1	91.9
hepatitus	76.3	76.5	77.3	76.7
horse	74.5	82.3	78.7	97.7
ionosphere	96	92.9	83.8	78.5
iris	94.7	94.7	94.7	95.2
led	73.7	71.2	69	96
mushroom	99.1	98.8	96.1	99.3
nursery	91.4	78.5	77.9	98
Page blocks	89.8	76.2	89.8	98.6
Pen digits	79.1	83	79.7	99.4
Pima indians	77.4	75.6	74.9	97.7
waveform	71.2	75.4	71.7	99.5
wine	97.1	92.5	86.3	83.6
zoo	92	96	93	96.1
Average	**82.0**	**81.2**	**80.7**	**91.9**

Table 5.3 show the accuracy of the classification obtained by the two methods, HS and Classification by Bagged Consistent Itemset Rules [41] by using other datasets. In all cases, the table shown is the average obtained from a 10-fold cross-validation using the full dataset. The accuracy obtained using the HS method is in 10 cases from 16 cases best than that obtained from the other method. Although the average accuracy of the HS method was more than the other method and only in five cases was it markedly worse than Classification by Bagged Consistent Itemset Rules.

Table 5.3. The results of accuracy

DataSet	eCCICbag	HS
Anneal	84.6	89.9
Austral	87.4	99.3
Auto	79.2	80.5
Breast	95.9	89.9
Glass	69.2	93.9
Heart	83.7	91.9
Hepatitis	83.3	76.7
Horse	86.1	97.7
Ionosphere	92.6	78.5
Iris	93.3	95.2
Led	73.2	96
Pima Indians	74.6	97.7
Tic-tac	97.5	94.1
Waveform	83.9	99.5
Wine	97.2	83.6
Zoo	96.1	96.1
Average	**87.2**	**91.2**

5.3.4 Summary of Results

Based on our empirical study, we demonstrated that our Hybrid Strategy approach works well on associative classification problems such as the rules number and classification accuracy. Generally speaking, the quality of the set of Hybrid Strategy generated depends on first; data preprocess to improve the quality of the data, thereby helping to improve the accuracy and efficiency of the mining process. Second; we used data post-analysis stage to reduce the number of itemsets that generates by association rule mining. Data post-analysis include data pruning to prune the redundant itemsets (where 50% of itemsets are prune), then used data clustering to grouping the remaining itemsets into classes, at the end we used data summarization to summarize the itemsets in each cluster where more than 70% of itemsets are removed.

The minimum support and minimum confidence threshold values affects on the results. When the minimum support value is low, the association rule mining algorithm tends to choose rules that cover many records from multiple classes. On the other hand, when the minimum confidence is high, the association rule mining algorithm tends to choose rules that cover only a small number of records from the same class. Consequently, only setting a low minimum support value without an appropriate minimum confidence value will generate rules that may misclassify some unseen records. Only setting high a minimum confidence value without an appropriate minimum support value will generate rules that are overfitted to the training data. It is important to find a good combination of these two threshold values. In our empirical study on 18 datasets, the minimum support is lowered to around 1%, and the minimum confidence value is 50%.

The Hybrid Strategy method was compared with some well-known associative classification methods such as CMAR, CPAR and TFPC. The Hybrid Strategy method has less average of rules (50.3) compare with others approaches CMAR (584.3), CPAR (٩٥) and TFPC (٢٩٨), and best average classification (91.9) compare with others approaches CMAR (82.0), CPAR (81.2) and TFPC (80.7). By using other 16 datasets, we camper Hybrid Strategy with Classification by Bagged Consistent Itemset Rules approach; Hybrid Strategy has best average classification (91.2) than other approach (87.2).

Chapter Five

Conclusions and Future Work

In this chapter, the work in this study is concluded in Section 5.1, and some directions for future research is presented in Section 5.2

5.1 Conclusions

Associative classification is an integrating association rules mining and classification. As the number of itemsets increases, the number of complex relationships among itemsets will grow exponentially. Therefore, it is difficult to apply associative classification on large datasets with many attributes. In order to solve the rules number problem. This study, first pre-process the data to improve the classification ability and the efficiency of the classification process then used an integrated and flexible framework for data post-analysis. It proposes a hybrid system of data post-analysis including data pruning, data clustering and data summarization to support effective analysis and interpretation of the discovered rules. By using the concept of sample itemset matching, redundant itemsets is first pruned by using maximal itemset pruning method. This allows the users to control the tradeoff between the number of itemsets being pruned and the amount of information loss after pruning. For data clustering, using *k*-mean clustering method which able to cluster itemsets as well as their associated data. Finally, the concept of sample-itemset matching is used in data summarization method. Data summarization using RuleCover methods to prune itemsets in each cluster those results from data clustering.

The selection of the minimum threshold value of minimum support and minimum confidence plays an important role in the Hybrid Strategy approach. Through empirical study we observed that only setting the minimum support value low without an appropriate minimum confidence value will generate rules that may misclassify some unseen record. While only setting a high minimum confidence value without an appropriate minimum support value will generate rules that overfit the training data. Consequently, it is important to find a good combination of these two threshold values. In our empirical study on the datasets, when minimum support is lowered to around 1% and the minimum confidence value is 50%.

By using datasets from UCI Repository, we compared the Hybrid Strategy approach with some well-known associative classification algorithms such as CMAR, CPAR, TFPC and Classification by Bagged Consistent Itemset Rules. The results that achieved by Hybrid Strategy not only give better classification accuracy but also have better results of rules number.

5.2 Future Work

This section offers some suggestions for future work. In addition to these recommendations they will encourage the researchers to conduct further studies in the research filed.

1- In this study, we used all the attributes in the dataset to construct the associative classification. However, attribute selection (feature selection) is also an active area in the field of machine learning and data mining. Another potential research direction is to study the impact of attribute selection on the classification performance of the Hybrid Strategy approach.

2- Clustering of unseen samples to existing itemset clusters. Itemset clusters are formed, each of them is well described by its summary itemsets. If the itemset 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 itemset clusters to investigate the properties of the new samples.

3- More efficient implementation of the data pruning. It is possible to integrate the data post-analysis system with the rule mining system so that redundant computation can be eliminated. For example, the counting of the number of samples containing a rule (i.e. supports) is usually available in the rule 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 itemset value pairs and their intersections or unions.

References

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

[2] Agrawal R., and Srikant R., 1994. Fast algorithms for mining association rules. *VLDB-94*, 1994.

[3] Agrawal R., Imielinski T., Swami A., “Database Mining: A Performance Perspective”, *IEEE Transactions on Knowledge and Data Engineering*, Vol.5, No. 6, pp.914-925. 1993b

[4] Al-hamami. Alaa, "Data Mining:- Concepts, Techniques and Applications", Ithraa Publishing and Distribution, Amman, Jordan, 2008.

[5] 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.

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

[7] 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.

[8] Breiman, L, Friedman, J., Olshen, R. and Stone C.. *Classification and regression trees*. Belmont: Wadsworth. 1984

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

[10] Brin S, Page L The anatomy of a large-scale hypertextualWeb Search Sngine. Comput Networks 30(1–7):107–117.1998

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

[12] Ceglar, A., Roddick, J.F.: Association mining. ACM Computing Surveys, volume 38(2) (2006).

[13] Coenen F., Leng P., Zhang L., "Threshold Tuning for Improved Classification Association Rule Mining", Springer-Verlag Berlin Heidelberg, pp. 216-225, 2005.

[14] Domingos P, Pazzani M On the optimality of the simple Bayesian classifier under zero-one loss. Mach Learn 29:103–130. (1997)

[15] Fayyad, Piatetsky-Shapiro, Smyth: From Data Mining to Knowledge Discovery in Databases. AI MAgazine, 1996.

[16] Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic: Knowledge Discovery and Data Mining: Towards a Unifying Framework. In Proceeding of The Second Int. Conference on Knowledge Discovery and Data Mining, pages 82--88, 1996.

[17] G. H. John. Enhancements to the Data Mining Process. Ph.D. Thesis, Computer Science Dept., Stanford Univeristy, 1997.

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

[19] Girish K., Mandar S., Manoj M.,” Association Rules Mining Using Heavy Itemsets”, Jayant Haritsa, T.M. Vijayaraman ,pp.148-155, 2005.

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

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

[22] Hand DJ, Yu K. "Idiot’s Bayes—not so stupid after all?." Int Stat Rev 69:385–398. (2001)

[23] Hansen, Hans Robert; Neumann, Gustaf: Wirtschaftsinformatik I. Lucius & Lucius, 2001.

[24] Hastie,T., Tibshirani,R., Friedman,J., , The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer. 2001

[25] Jin R, Goswami A, Agrawal G. "Fast and exact out-of-core and distributed *k*-means clustering". Knowl Inf Syst 10(1):17–40. (2006)

[26] Kearns M. and Ron. D. Algorithmic stability and sanity-check bounds for leave-one-out crossvalidation. *Neural Computation*, 11(6):1427–1453, 1996.

[27] Kohavi R.. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1137–1143, 1995.

[28] Li J., Dong G., Ramamohanarao K., and Wong L.. DeEPs: a new instance-based lazy discovery and classification system. *Machine Learning*, 54(2):99–124, 2004.

[29] Li W., Han J., and Pei J. CMAR: accurate and efficient classification based on multipleclass-association rules. In *Proceedings of IEEE International Conference on Data Mining (ICDM2001)*, pages 369–376, 2001.

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

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

[32] Liu B., Hsu W., Ma Y.. Integrating classification and association rule mining. *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, USA, New York 1998, pp.80-86.

[33] Liu, Bing: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, 2007.

[34] Madigan,D., Ridgeway,G., 2003, “Bayesian Data Analysis”. The Handbook of Data Mining, LEA Publishers, pp.279-304, 2003.

[35] Michalski,R.S., Bratko,I., Kubat,M., , Machine Learning and Data Mining, Methods and Applications, John Wiley & Sons Ltd. 1998

[36] Michie,D., Spiegelhalter,D.J., Taylor,C.C.,. Machine Learning, Neural and Statistical Classification, Ellis Horwood. 1994

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

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

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

[40] Quinlan JR. "C4.5: Programs for machine learning. Morgan Kaufmann Publishers, San Mateo. (1993)

[41] Shidara Y., Kudo M., A. Nakamura, "Classification by Bagged Consistent Itemset Rules". IEEE.2008.

[42] Soo.P.J.,C. Ming-Syan,P.S.Yu,” Using a Hash-Based Method with Transaction Trimming for Mining Association Rules”,*IEEE Trans.Knowledge and Data Engg.*,Vol.9,No.5,Sep./Oct.1997,pp.813-825.

[43] Steinbach, M., Kumar, V.: Generalizing the notion of confidence. Knowledge and Information Systems, volume 12(3) (2007) 279–299.

[44] Tan P., Steinbach M., Kumar V., "Introduction to Data Mining", Pearson Education,Inc.,2006.

[45] Tiovonen H.,”Sampling Large Databasees ForAssociation Rules”,proc.VLDB 2000 .

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

[47] Tzanis, G., Berberidis, C.: Mining for mutually exclusive items in transaction databases. International Journal of Data Warehousing and Mining, volume 3(3) (2007) 45–59.

[48] Vapnik V (1995) The nature of statistical learning theory. Springer, New York.

[49] Wang Y. and A. K. C. Wong. From association to classification: Inference using weight of evidence. *IEEE Trans. On Knowledge and Data Engineering*, 15(3):764–767, 2003.

[50] Wasserman S, Raust K (1994) Social network analysis. Cambridge University Press, Cambridge.

[51] Weiss,S.M., Kapouleas,I., “An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods, Readings in Machine Learning, edited by J.W. Shavlik, T.G. Dietterich, Morgan Kaufmann Publishers Inc., pp.177-183. 1990.

[52] Yin X. and Han J. CPAR: Classification based on predictive association rules. In *3rd SIAM International Conference on Data Mining (SDM’03)*, 2003.

[53] Zaki M. J., Hsiao C. J., “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**.**

[54] Zubair Rahman A.M.J. Md. and Balasubramanie P. " An Efficient Algorithm for Mining Maximal Frequent Item Sets" Journal of Computer Science 4 (8): 638-645, 2008.

الملخص

التصنيف الترابطي هو دمج تنقيب قواعد البيانات مع التصنيف . وهو يعمل على كشف الأنماط للحصول على معلومات مفيدة وهامة في مجال العلوم والهندسة والأعمال التجارية. في هذه الايام طرق التصنيف الترابطي الفعالة تم تطويرها واستخدامها على نطاق واسع في مختلف التطبيقات التجارية والصناعية. هذه الطرق تحاول تقديم معلومات هامة ومفيدة لصانعي القرار . ومن المفارقات، التصنيف الترابطي يولد اعداد كبيرة من القواعد التي تشكل مشكلة لإدارة المعرفة: للتعامل مع الآلاف أو أكثر من القواعد المكتشفة في مجموعة البيانات . من اجل حل مشكلة عدد القواعد، في هذه الدراسة تم معالجة التصنيف الترابطي على مرحلتين أولا المعالجة الأولية لتحسين قدرة التصنيف والكفاءة في عملية التصنيف. ثانيا من خلال استخدام مطابقة الأنماط تم اقتراح نظام هجين لتحليل البيانات ،هذا النظام الهجين يتضمن تشذيب البيانات، عنقدة البيانات وتلخيص البيانات لدعم التحليل والتفسير الفعال للقواعد المكتشفة. في تشذيب البيانات يتم تشذيب الأنماط المتكررة والزائدة عن الحاجة اما في عنقدة البيانات يتم تجميع الأنماط الناتجة من تشذيب البيانات في مجاميع مماثلة . وأخيرا،تلخيص البيانات يستخدم لتشذيب الأنماط في كل عنقود ناتج من عنقدة البيانات . في الطريقة الإستراتيجية الهجينة، نحن نستخدم معايير مركبة هي الحد الأدنى للدعم والحد الأدنى من الثقة كوزن للقاعدة للإشارة إلى أهمية القاعدة. ومن خلال دراستنا ،فاننا نجد انه من المهم إيجاد توليفه جيدة بين هاتين القيمتين .

باستخدام قواعد البيانات من UCI repository، قارنا نتائج الإستراتيجية الهجينة مع بعض خوارزميات التصنيف الترابطي المعروفة مثل CMAR، CPAR، TFPC و classification by Bagged consistent Itemset Rules. النتائج التي حققتها الإستراتيجية الهجينة ليست فقط اكثر دقة ولكن أفضل من حيث عدد القواعد .الإستراتيجية الهجينة تولد مجموعة اقل بكثير من القواعد التنبؤية العالية الجودة مع دقة عالية مقارنة مع هذه الخوارزميات.

Appendix A

Sample of wine data set

```
١,١٤.٢٣,١.٧١,٢.٤٣,١٥.٦,١٢٧,٢.٨,٣.٠٦,٢٨,٢.٢٩,٥.٦٤,١.٠٤,٣.٩٢,١٠٦٥
١,١٣.٢,١.٧٨,٢.١٤,١١.٢,١٠٠,٢.٦٥,٢.٧٦,٢٦,١.٢٨,٤.٣٨,١.٠٥,٣.٤,١٠٥٠
١,١٣.١٦,٢.٣٦,٢.٦٧,١٨.٦,١٠١,٢.٨,٣.٢٤,٣,٢.٨١,٥.٦٨,١.٠٣,٣.١٧,١١٨٥
١,١٤.٣٧,١.٩٥,٢.٥,١٦.٨,١١٣,٣.٨٥,٣.٤٩ ،٢٤,٢.١٨,٧.٨,٨٦,٣.٤٥,١٤٨٠
١,١٣.٢٤,٢.٥٩,٢.٨٧,٢١,١١٨,٢.٨,٢.٦٩,٣٩,١.٨٢,٤.٣٢,١.٠٤,٢.٩٣,٧٣٥
١,١٤.٢,١.٧٦,٢.٤٥,١٥.٢,١١٢,٣.٢٧,٣.٣٩,٣٤,١.٩٧,٦.٧٥,١.٠٥,٢.٨٥,١٤٥٠
١,١٤.٣٩,١.٨٧,٢.٤٥,١٤.٦,٩٦,٢.٥,٢.٥٢,٣,١.٩٨,٥.٢٥,١.٠٢,٣.٥٨,١٢٩٠
١,١٤.٠٦,٢.١٥,٢.٦١,١٧.٦,١٢١,٢.٦,٢.٥١,٣١,١.٢٥,٥.٠٥,١.٠٦,٣.٥٨,١٢٩٥
١,١٤.٨٣,١.٦٤,٢.١٧,١٤,٩٧,٢.٨,٢.٩٨,٢٩,١.٩٨,٥.٢,١.٠٨,٢.٨٥,١٠٤٥
١,١٣.٨٦,١.٣٥,٢.٢٧,١٦,٩٨,٢.٩٨,٣.١٥,٢٢,١.٨٥,٧.٢٢,١.٠١,٣.٥٥,١٠٤٥
١,١٤.١,٢.١٦,٢.٣,١٨,١٠٥,٢.٩٥,٣.٣٢,٢٢,٢.٣٨,٥.٧٥,١.٢٥,٣.١٧,١٥١٠
١,١٤.١٢,١.٤٨,٢.٣٢,١٦.٨,٩٥,٢.٢,٢.٤٣,٢٦,١.٥٧,٥,١.١٧,٢.٨٢,١٢٨٠
١,١٣.٧٥,١.٧٣,٢.٤١,١٦,٨٩,٢.٦,٢.٧٦,٢٩,١.٨١,٥.٦,١.١٥,٢.٩,١٣٢٠
١,١٤.٧٥,١.٧٣,٢.٣٩,١١.٤,٩١,٣.١,٣.٦٩,٤٣,٢.٨١,٥.٤,١.٢٥,٢.٧٣,١١٥٠
١,١٤.٣٨,١.٨٧,٢.٣٨,١٢,١٠٢,٣.٣,٣.٦٤,٢٩,٢.٩٦,٧.٥,١.٢,٣,١٥٤٧
١,١٣.٦٣,١.٨١,٢.٧,١٧.٢,١١٢,٢.٨٥,٢.٩١,٣,١.٤٦,٧.٣,١.٢٨,٢.٨٨,١٣١٠
١,١٤.٣,١.٩٢,٢.٧٢,٢٠,١٢٠,٢.٨,٣.١٤,٣٣,١.٩٧,٦.٢,١.٠٧,٢.٦٥,١٢٨٠
١,١٣.٨٣,١.٥٧,٢.٦٢,٢٠,١١٥,٢.٩٥,٣.٤,٤,١.٧٢,٦.٦,١.١٣,٢.٥٧,١١٣٠
١,١٤.١٩,١.٥٩,٢.٤٨,١٦.٥,١٠٨,٣.٣,٣.٩٣,٣٢,١.٨٦,٨.٧,١.٢٣,٢.٨٢,١٦٨٠
١,١٣.٦٤,٣.١,٢.٥٦,١٥.٢,١١٦,٢.٧,٣.٠٣ ،١٧,١.٦٦,٥.١,٩٦,٣.٣٦,٨٤٥
١,١٤.٠٦,١.٦٣,٢.٢٨,١٦,١٢٦,٣,٣.١٧,٢٤,٢.١,٥.٦٥,١.٠٩,٣.٧١,٧٨٠
١,١٢.٩٣,٣.٨,٢.٦٥,١٨.٦,١٠٢,٢.٤١,٢.٤١,٢٥,١.٩٨,٤.٥,١.٠٣,٣.٥٢,٧٧٠
١,١٣.٧١,١.٨٦,٢.٣٦,١٦.٦,١٠١,٢.٦١,٢.٨٨,٢٧,١.٦٩,٣.٨,١.١١,٤,١٠٣٥
١,١٢.٨٥,١.٦,٢.٥٢,١٧.٨,٩٥,٢.٤٨,٢.٣٧,٢٦,١.٤٦,٣.٩٣,١.٠٩,٣.٦٣,١٠١٥
١,١٣.٥,١.٨١,٢.٦١,٢٠,٩٦,٢.٥٣,٢.٦١,٢٨,١.٦٦,٣.٥٢,١.١٢,٣.٨٢,٨٤٥
١,١٣.٠٥,٢.٠٥,٣.٢٢,٢٥,١٢٤,٢.٦٣,٢.٦٨,٤٧,١.٩٢,٣.٥٨,١.١٣,٣.٢,٨٣٠
١,١٣.٣٩,١.٧٧,٢.٦٢,١٦.١,٩٣,٢.٨٥,٢.٩٤ ،٣٤,١.٤٥,٤.٨,٩٢,٣.٢٢,١١٩٥
١,١٣.٣,١.٧٢,٢.١٤,١٧,٩٤,٢.٤,٢.١٩,٢٧,١.٣٥,٣.٩٥,١.٠٢,٢.٧٧,١٢٨٥
١,١٣.٨٧,١.٩,٢.٨,١٩.٤,١٠٧,٢.٩٥,٢.٩٧,٣٧,١.٧٦,٤.٥,١.٢٥,٣.٤,٩١٥
١,١٤.٠٢,١.٦٨,٢.٢١,١٦,٩٦,٢.٦٥,٢.٣٣,٢٦,١.٩٨,٤.٧,١.٠٤,٣.٥٩,١٠٣٥
١,١٣.٧٣,١.٥,٢.٧,٢٢.٥,١٠١,٣,٣.٢٥,٢٩,٢.٣٨,٥.٧,١.١٩,٢.٧١,١٢٨٥
١,١٣.٥٨,١.٦٦,٢.٣٦,١٩.١,١٠٦,٢.٨٦,٣.١٩,٢٢,١.٩٥,٦.٩,١.٠٩,٢.٨٨,١٥١٥
١,١٣.٦٨,١.٨٣,٢.٣٦,١٧.٢,١٠٤,٢.٤٢,٢.٦٩,٤٢,١.٩٧,٣.٨٤,١.٢٣,٢.٨٧,٩٩٠
١,١٣.٧٦,١.٥٣,٢.٧,١٩.٥,١٣٢,٢.٩٥,٢.٧٤,٥,١.٣٥,٥.٤,١.٢٥,٣,١٢٣٥
١,١٣.٥١,١.٨,٢.٦٥,١٩,١١٠,٢.٣٥,٢.٥٣,٢٩,١.٥٤,٤.٢,١.١,٢.٨٧,١٠٩٥
١,١٣.٤٨,١.٨١,٢.٤١,٢٠.٥,١٠٠,٢.٧,٢.٩٨,٢٦,١.٨٦,٥.١,١.٠٤,٣.٤٧,٩٢٠
```

Appendix B

Sample of wine data set after pruning

{Alcohol=1,Malic=1.16,Ash=2.36,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,diluted-wines=1.3} : 40 time

{Alcohol=1,Malic=1.16,Ash=2.36,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,Proline=3.17} : 40 time

{Alcohol=1,Malic=1.16,Ash=2.36,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Color-intensity=2.81,diluted wines=1.3,Proline=3.17} : 40 time

{Alcohol=1,Malic=1.16,Ash=2.36,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Hue=68,dilutedwines=1.3,Proline=3.17} : 40 time

{Alcohol=1,Malic=1.16,Ash=2.36,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Colorintensity=2.81,Hue=68,dilutedwines=1.3,Proline=3.17} : 40 time

{Alcohol=1,Malic=1.16,Ash=2.36,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,dilutedwines=1.3,Proline=3.17} : 40 time

{Alcohol=1,Malic=1.16,Ash=2.36,Alcalinity=2.67,Magnesium=18,phenols=101,Nonflavanoid=3.24,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,dilutedwines=1.3,Proline=3.17} : 40 time

{Alcohol=1,Malic=1.16,Ash=2.36,Alcalinity=2.67,Magnesium=18,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,dilutedwines=1.3,Proline=3.17} : 40 time

{Alcohol=1,Malic=1.16,Ash=2.36,Alcalinity=2.67,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,dilutedwines=1.3,Proline=3.17} : 40 time

}Alcohol=1,Malic=1.16,Ash=2.36,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,dilutedwines=1.3,Proline=3.17} : 40 time

}Alcohol=1,Malic=1.16,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,dilutedwines=1.3,Proline=3.17} : 40 time

{Alcohol=1,Ash=2.36,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,dilutedwines=1.3,Proline=3.17} : 40 time

{Malic=1.16,Ash=2.36,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,dilutedwines=1.3,Proline=3.17} : 40 time

{Alcohol=1,Malic=1.16,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Colorintensity=2.81,Hue=68,dilutedwines=1.3} : 40 time

{Alcohol=1,Malic=1.16,Alcalinity=2.67,Magnesium=18,phenols=101,Flavanoids=2.8,Nonflavanoid=3.24,Proanthocyanins=.3,Color-
(intensity=2.81,Hue=68,Proline=3.17} : 40 time(s

Appendix C

Sample of wine data set after clustering

Itemset	Clusters
1-1.16-2.67-18-101-2.8-3.24-.3-.81-68-1.3-3.17	1 1 1 1 1 1 19 19 1 1
1-1.16-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17	1 1 1 1 1 1 10 1 1 1
1-1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1-1.16-2.36-2.67-18-2.8-3.24-.3-2.81-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1-1.16-2.36-2.67-18-101-3.24-.3-2.81-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1-1.16-2.36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1-1.16-2.36-2.67-18-101-2.8-3.24-2.81-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1.16-2.36-2.67-18-101-2.8-3.24-.3-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1-1.16-2.36-2.67-18-101-2.8-3.24-.3-2.81-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1-1.16-2.36-2.67-18-101-2.8-3.24-.3-2.81-68-3.17	1 1 1 1 1 1 1 1 1 1
1-1.16-2.36-2.67-18-101-2.8-3.24-.3-2.81-68-1.3	1 1 1 1 1 1 1 1 1 1
1-2.36-2.67-18-101-2.8-3.24-.3-2.81-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1.16-2.67-18-101-2.8-3.24-.3-2.81-68-1.3-3.17	10 1 1 1 1 1 1 1 1 1
1.16-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17	1 1 1 19 1 19 1 1 10 1
1.16-2.36-2.67-18-2.8-3.24-.3-2.81-68-1.3-3.17	1 1 1 1 10 1 1 1 1 1
1.16-2.36-2.67-18-101-3.24-.3-2.81-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1.16-2.36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1.16-2.36-2.67-18-101-2.8-3.24-2.81-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1
1.16-2.36-2.67-18-101-2.8-3.24-.3-68-1.3-3.17	1 1 1 1 1 1 1 1 1 1

```
1.16-2.36-2.67-18-101-2.8-3.24-.3-2.81-1.3-3.17     1  1  1  1  1  1  1  1  1  1

1.16-2.36-2.67-18-101-2.8-3.24-.3-2.81-68-3.17      1  1  1  1  1  1  1  1  1  1

1.16-2.36-2.67-18-101-2.8-3.24-.3-2.81-68-1.3       1  1  1  1  1  1  1  1  1  1

1-2.67-18-101-2.8-3.24-.3-2.81-68-1.3-3.17          1  1  1  1  1  1  1  1  1  1

1-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17          1  1  1  1  1  1  1  1  1  1

1-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17        1  1  1  1  1  1  1  1  1  1

1-2.36-2.67-18-2.8-3.24-.3-2.81-68-1.3-3.17         1  1  1  1  1  1  1  1  1  1

1-2.36-2.67-18-101-3.24-.3-2.81-68-1.3-3.17         10  1  1  1  1  1  1  1  1  1

1-2-36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17          1 28 10  1  1  1  1  1  1  1

1-2.36-2.67-18-101-2.8-.24-2.81-68-1.3-3.17         1 10  1  1  1  1  1  1  1  1

1-2.36-2.67-18-101-2.8-3.24-.3-68-1.3-3.17          1 10  1  1  1  1  1  1  1  1

1-2.36-2.67-18-101-2.8-3.24-.3-2.81-1.3-3.17        1  1  1  1  1  1  1  1  1  1

1-2.36-2.67-18-101-2.8-3.24-.3-2.81-68-3.17         1  1  1  1  1  1  1  1  1  1

1-2.36-2.67-18-101-2.8-3.24-.3-2.81-68-1.3          1  1  1  1  1  1  1  1  1  1

1-1.16-18-101-2.8-3.24-.3-2.81-68-1.3-3.17          1  1  1  1  1  1  1  1  1  1

1-1-16-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17        1  1  1  1  1  1  1  1  1  1

1-1.16-2.67-18-2.8-3.24-.3-2.81-68-1.3-3.17         1  1  1  1  1  1  1  1  1  1

1-1.16-2.67-18-101-3.24-.3-2.81-68-1.3-3.17         1  1  1  1  1  1  1  1  1  1

1-1.16-2.67-18-101-2.8-.3-2.81-68-1.3-3.17          1  1  1  1  1  1  1  1  1  1

1-1.16-2.67-18-101-2.8-3.24-2.81-68-1.3-3.17        1  1  1  1  1  1  1  1  1  1

1-1.16-2.67-18-101-2.8-3.24-.3-68-1.3-3.17          1  1  1  1  1  1  1  1  1  1

1-1.16-2.67-18-101-2.8-3.24-.3-2.81-1.3-3.17        1  1  1  1  1  1  1  1  1 10
```

C-٢

Appendix D

Sample of wine data set after summarization

1-2.36-2.67-18-101-2.8-.24-2.81-68-1.3-3.17 27 ==> 1-2-36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17 27

1-1.16-2.67-18-101-2.8-3.24-.3-.81-68-1.3-3.17 18 ==> 1-1.16-2.67-18-101-2.8-3.24-.3-.81-68-1.3-3.17 18

1-1.16-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> 1-1.16-2.67-18-101-2.8-3.24-.3-.81-68-1.3-3.17 18

1-1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> 1-1.16-2.67-18-101-2.8-3.24-.3-.81-68-1.3-3.17 18

1-1.16-2.36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17 18 ==> 1-1.16-2.67-18-101-2.8-3.24-.3-.81-68-1.3-3.17 18

1-1.16-2.36-2.67-18-101-2.8-3.24-2.81-68-1.3-3.17 18 ==> 1-1.16-2.67-18-101-2.8-3.24-.3-.81-68-1.3-3.17 18

1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> 1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1.16-2.36-2.67-18-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> 1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1.16-2.36-2.67-18-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> 1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1.16-2.36-2.67-18-101-3.24-.3-2.81-68-1.3-3.17 18 ==> 1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1.16-2.36-2.67-18-101-3.24-.3-2.81-68-1.3-3.17 18 ==> 1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1.16-2.36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17 18 ==> 1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1.16-2.36-2.67-18-101-2.8-3.24-2.81-68-1.3-3.17 18 ==> 1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1.16-2.36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17 18 ==> 1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1.16-2.36-2.67-18-101-2.8-3.24-.3-68-1.3-3.17 18 ==>1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1.16-2.36-2.67-18-101-2.8-3.24-2.81-68-1.3-3.17 18 ==> 1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1.16-2.36-2.67-18-101-2.8-3.24-.3-2.81-68-3.17 18 ==> 1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1-2.36-2.67-18-101-2.8-3.24-.3-68-1.3-3.17 18 ==> 1-2-36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17 18

1-2.36-2.67-18-101-2.8-3.24-.3-2.81-1.3-3.17 18 ==> 1-2-36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17 18

1-2.36-2.67-18-101-2.8-3.24-.3-2.81-68-1.3 18 ==> 1-2-36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17 18

1-1-16-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> 1-2-36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17 18

1-1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> 1-1.16-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1-1.16-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==>1-1.16-2.36-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1-1.16-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> 1-1.16-2.36-2.67-18-2.8-3.24-.3-2.81-68-1.3-3.17 18

1-1.16-2.36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17 18 ==> 1-1.16-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1-1.16-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> 1-1.16-2.36-2.67-18-101-2.8-.3-2.81-68-1.3-3.17 18

1-1.16-2.36-2.67-18-101-2.8-3.24-2.81-68-1.3-3.17 18 ==> 1-1.16-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17 18

1-1.16-2.36-18-101-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> G=g1-1.16-2.36-2.67-18-101-2.8-3.24-2.81-68-1.3-3.17 18

1-2.36-2.67-18-101-2.8-3.24-.3-68-1.3-3.17 18 ==> B=b1-2.36-2.67-18-101-2.8-.24-2.81-68-1.3-3.17 18

1-2.36-2.67-18-101-2.8-3.24-.3-2.81-1.3-3.17 18 ==> B=b1-2.36-2.67-18-101-2.8-.24-2.81-68-1.3-3.17 18

1-2.36-2.67-18-101-2.8-3.24-.3-2.81-68-1.3 18 ==> 1-2.36-2.67-18-101-2.8-.24-2.81-68-1.3-3.17 18

1-1-16-2.67-101-2.8-3.24-.3-2.81-68-1.3-3.17 18 ==> 1-2.36-2.67-18-101-2.8-.24-2.81-68-1.3-3.17 18

Appendix E

Sample of wine data set after running Hybrid Strategy

{Magnesium=18,Flavanoids=2.8} : 40 time(s) ==> {Proline2=11} : 40 time(s)   conf:(1)

{Magnesium=18,Nonflavanoid=3.24} : 40 time(s) ==> {Proline2=11} : 40 time(s) conf:(1)

{Magnesium=18,Color-intensity=2.81} : 40 time(s) ==> {Proline2=11} : 40 time(s) conf:(1)

{Magnesium=18,Hue=68} : 40 time(s) ==> {Proline2=11} : 40 time(s)   conf:(1)

{phenols=101,Flavanoids=2.8} : 40 time(s) ==> {Proline2=11} : 40 time(s)   conf:(1)

{phenols=101,Nonflavanoid=3.24} : 40 time(s) ==> {Proline2=11} : 40 time(s) conf:(1)

{phenols=101,Proanthocyanins=.3} : 40 time(s) ==> {Proline2=11} : 40 time(s) conf:(1)

{phenols=101,Color-intensity=2.81} : 40 time(s) ==> {Proline2=11} : 40 time(s) conf:(1)

{phenols=101,Hue=68} : 40 time(s) ==> {Proline2=11} : 40 time(s)   conf:(1)

{phenols=101,diluted-wines=1.3} : 40 time(s) ==> {Proline2=11} : 40 time(s) conf:(1)

{phenols=101,Proline=3.17} : 40 time(s) ==> {Proline2=11} : 40 time(s)   conf:(1)

{Flavanoids=2.8,Nonflavanoid=3.24} : 40 time(s) ==> {Proline2=11} : 40 time(s) conf:(1)

{Flavanoids=2.8,Proanthocyanins=.3} : 40 time(s) ==> {Proline2=11} : 40 time(s) conf:(1)

{Flavanoids=2.8,Color-intensity=2.81} : 40 time(s) ==> {Proline2=11} : 40 time(s) conf:(1)

{Flavanoids=2.8,Hue=68} : 40 time(s) ==> {Proline2=11} : 40 time(s)   conf:(1)

{Flavanoids=2.8,diluted-wines=1.3} : 40 time(s) ==> {Proline2=11} : 40 time(s) conf:(1)

{Flavanoids=2.8,Proline=3.17} : 40 time(s) ==> {Proline2=11} : 40 time(s)   conf:(1)

```
{Nonflavanoid=3.24,Proanthocyanins=.3} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Nonflavanoid=3.24,Color-intensity=2.81} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Nonflavanoid=3.24,Hue=68} : 40 time(s) ==> {Proline2=11} : 40 time(s)    conf:(1)

{Nonflavanoid=3.24,diluted-wines=1.3} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Nonflavanoid=3.24,Proline=3.17} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Proanthocyanins=.3,Color-intensity=2.81} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Proanthocyanins=.3,Hue=68} : 40 time(s) ==> {Proline2=11} : 40 time(s)    conf:(1)

{Proanthocyanins=.3,diluted-wines=1.3} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Proanthocyanins=.3,Proline=3.17} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Color-intensity=2.81,Hue=68} : 40 time(s) ==> {Proline2=11} : 40 time(s)    conf:(1)

{Color-intensity=2.81,diluted-wines=1.3} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Color-intensity=2.81,Proline=3.17} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Hue=68,diluted-wines=1.3} : 40 time(s) ==> {Proline2=11} : 40 time(s)    conf:(1)

{Hue=68,Proline=3.17} : 40 time(s) ==> {Proline2=11} : 40 time(s)    conf:(1)

{diluted-wines=1.3,Proline=3.17} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Alcohol=1,Malic=1.16,Ash=2.36} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Alcohol=1,Malic=1.16,Alcalinity=2.67} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)

{Alcohol=1,Malic=1.16,Magnesium=18} : 40 time(s) ==> {Proline2=11} : 40 time(s)
conf:(1)
```