Digital Ecosystems and Business Intelligence Institute
Curtin Business School

Quality and Interestingness of Association Rules Derived from Data Mining of Relational and Semi-Structured Data

Izwan Nizal Mohd Shaharanee

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

February 2012

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: .....

Date: 29/02/2012

Acknowledgement

First, I would like to thank my wife Shariza and my two children Asyraf and Aliya for for their constant courage and inspiration. To my parents, who have been instrumental in guiding my life and encouraging me to succeed, I express my deepest thanks. They are the reasons that I am who I am today.

I would like to convey my highest appreciation to Professor Tharam Dillon, my supervisor, and Dr. Fedja Hadzic, my co-supervisor, for their guidance throughout my study period. Without Professor Tharam Dillon's priceless knowledge, especially in the application area of semi-structured data mining, insightful advice, important suggestions related to my research framework, and experienced guidance in writing the thesis; I would have never arrived at this point.

For various knowledge stretches beyond academic context, I am deeply indebted to Dr. Fedja Hadzic. He has generously given his time and intellectuality, always ready for intelligent discussions and to give suggestions where needed, very supportive during the experiment phase, including providing elaborations of his work that related to this research. He has patiently reading many drafts of this thesis as well, to which constructive feedbacks were given.

I am very fortunate to have had the opportunity to work with and learn from them. I genuinely appreciate their kindness and help. Thank you. Your wisdom teaches me to live humbly and sincerely.

I am also grateful to Professor Elizabeth Chang, who was the Director of DEBI Institute during most of my PhD tenure, for providing a wonderful research environment. The seminars throughout the years, together with friendly research fellows, visiting lectures and supporting staff have made my study much easier. I would also like to extend my gratitude to the Malaysian Ministry of Higher Education and Universiti Utara Malaysia for sponsoring my study and stay in Australia. Their financial aid was indispensable for the completion of this thesis. Last, but by no means least, I thank all the friends whom I have met here in Australia, for their friendship and support.

Abstract

Deriving useful and interesting rules from a data mining system are essential and important tasks. Problems such as the discovery of random and coincidental patterns or patterns with no significant values, and the generation of a large volume of rules from a database commonly occur. Works on sustaining the interestingness of rules generated by data mining algorithms are actively and constantly being examined and developed. As the data mining techniques are data-driven, it is beneficial to affirm the rules using a statistical approach. It is important to establish the ways in which the existing statistical measures and constraint parameters can be effectively utilized and the sequence of their usage.

In this thesis, a systematic way to evaluate the association rules discovered from frequent, closed and maximal itemset mining algorithms; and frequent subtree mining algorithm including the rules based on induced, embedded and disconnected subtrees is presented. With reference to the frequent subtree mining, in addition a new direction is explored based on utilizing the *DSM* approach capable of preserving all information from tree-structured database in a flat data format, consequently enabling the direct application of a wider range of data mining analysis/techniques to tree-structured data. Implications of this approach were investigated and it was found that basing rules on disconnected subtrees, can be useful in terms of increasing the accuracy and the coverage rate of the rule set. A strategy that combines data mining and statistical measurement techniques such as sampling, redundancy and contradictive checks, correlation and regression analysis to evaluate the rules is developed. This framework is then applied to real-world datasets that represent diverse characteristics of data/items. Empirical results show that with a proper combination of data mining and statistical analysis, the proposed framework is capable of eliminating a large number of non-significant, redundant and contradictive rules while preserving relatively valuable high accuracy rules. Moreover, the results reveal the important characteristics and differences between mining frequent, closed or maximal itemsets; and mining frequent subtree including the rules based on induced, embedded and disconnected subtrees; as well as the impact of confidence measure for the prediction and classification task.

List of Publications

Shaharanee, I. N. M., Hadzic, F., & Dillon, T. S. (2012). Evaluating Frequent, Closed and Maximal Itemset Rules Using Statistical Approaches. Submitted to *Expert Systems With Applications*.

Shaharanee, I. N. M., Hadzic, F., & Dillon, T. S. (2011). Interestingness measures for association rules based on statistical validity. *Knowledge-Based Systems, 24*, 386-392.

Shaharanee, I. N. M., Hadzic, F., & Dillon, T. (2010). A Statistical Interestingness Measures for XML Based Association Rules. In B.-T. Zhang & M. Orgun (Eds.), *PRICAI 2010: Trends in Artificial Intelligence* (Vol. 6230, pp. 194-205): Springer Berlin / Heidelberg.

Shaharanee, I. N. M., Hadzic, F., & Dillon, T. (2009). Interestingness of Association Rules Using Symmetrical Tau and Logistic Regression. In A. Nicholson & X. Li (Eds.), *AI 2009* (Vol. 5866, pp. 442-431): LNAI.

Shaharanee, I. N. M., Dillon, T. S., & Hadzic, F. (2009). Ascertaining Association Rules Using Statistical Analysis. In P. S. Sandhu (Ed.), *2009 International Symposium on Computing, Communication and Control (ISCCC 2009)* (pp. 180-188). Singapore: IACSIT

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Introduction

Since the explosion of information and data, largely due to the automation of business activities and increase in computing power, humans have faced the great challenge of converting data/information into meaningful and presentable formats. Consequently, this creates a growing space between the data/information generation and data/information understanding (Frawley, Piatetsky-Shapiro, & Matheus, 1992). Thus, the knowledge discovery process by either automatic or semi-automatic means has been introduced in order to discover useful information, hidden patterns or rules from large quantities of data.

Knowledge discovery techniques have been well researched by the data mining community. Such techniques, especially those used for unsupervised learning, generate a large quantity of rules and patterns. While many interesting and useful rules can be generated, there are still situations where many of the rules discovered are not of interest to the application domain as they often reflect previously established patterns (existing domain knowledge), coincidently occurring patterns, and in general, those that amount to worthless knowledge in real-world applications.

Sustaining the interestingness of rules generated by data mining algorithms is an active and important area of data mining research. Different methods have been proposed and have been well examined for discovering interestingness in rules. Even though the interesting rules may be found from the database using these methods, a problem that can still occur is that those rules nevertheless reflect just the database being observed (Webb, 2007).

Therefore, we can still argue the validity of using the rules and patterns for practical problems. Data mining approaches naturally are data-driven. For that reason, in order to make them more reliable in practice, each hypothesis generated from a data mining algorithm can be confirmed by statistical methodology. Therefore, in this research, the quality of data mining rules will be verified by both data mining and statistical measurement techniques. Such a combination is crucial in addressing the

practical problems produced by data overload and an excessive number of patterns/rules discovered from the data.

1.2 Knowledge Discovery and Data Mining

The knowledge discovery process involves several phases including: data pre-processing, data mining applications, patterns evaluation and knowledge presentation. One of the essential phases, considered to be the heart of knowledge discovery, is the data mining phase. In this phase, the focus is mainly on developing and analysing useful algorithms that can be used to extract and reveal the information and patterns emerging from the data. Since the definition of knowledge discovery and data mining are interchangeable (Han & Kamber, 2001), we make a clear distinction between them. We consider knowledge discovery to be the overall process of discovering useful knowledge from data, and data mining is that particular phase in the process which concentrates on method and algorithm application (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). In the following, we define each of the knowledge discovery phases (Han & Kamber, 2001; Roiger & Geatz, 2003) which are related to the thesis.

1.2.1 Data Pre-processing

The main idea of data pre-processing is to ensure that data fed into the data mining phase is clean (high quality of data) and only appropriate data are selected. Generally, there are multiple sources of data involving different data types and metrics sources to be used in the knowledge discovery process. Data may be anomalous, incorrect or missing. Within this phase, the erroneous data may be corrected or removed, whereas records containing missing data may be ignored, or the missing data may be supplied or predicted (often using data mining tools). Additionally, data from different sources are often converted into a common format for processing and some data may be encoded or transformed into more usable formats. Data reduction, data cleaning, data integration, data transformation, data reduction and data discretization can offer multiple ways of handling data pre-processing problem. An early and fairly good understanding of the data is needed since the original data are often raw in nature. Hence, a domain expert may be

needed to help translate the data into another form that is understandable by the data miner in order to model the problem (Han, Kamber, & Pei, 2012).

1.2.2 Data Mining

An intelligent and efficient method is needed to digest and find hidden and useful information from a large volume of data. Generally, the data mining process involves the application of certain methods that are capable of extracting information and reveal hitherto unknown patterns. The most commonly known methods are classification, prediction, characterization, clustering, association rules, sequence analysis, etc. The choice of method is strongly related to "*what to solve*" and "*the nature of data*". In this thesis, although our main focus is on patterns evaluation, the initial phases of data pre-processing and the important selection of suitable data mining applications will be discussed in sufficient detail to support the clarity of the generated patterns to be evaluated.

1.2.3 Pattern Evaluation

The hidden patterns or rules that are obtained from the data mining technique, are considered interesting and useful if the rules are *comprehensible, valid on tests and new data with some degree of certainty, potentially useful, actionable, and novel* (Han & Kamber, 2001). However, problems such as random pattern discovery, coincidental patterns, patterns with no significant value and the generation of a large volume of rules from a database, commonly occur. Thus, the model generated using the selected methods must be evaluated in order to measure its validity. Evaluation methods might include the confusion matrix, expert evaluation and field test, and statistical analysis.

1.2.4 Knowledge Interpretation/Presentation

Certain data mining output is in a format not readily understood by humans, and hence needs further processing in order to be interpreted. The interpretation may require converting such output into an easy-to-understand medium using various visualization and knowledge representation techniques. Additionally, the knowledge

must be consolidated and resolved with previous knowledge, shared, reported, disseminated, and acted upon (Han & Kamber, 2001).

1.3 Data Mining Tasks

Descriptive modelling and predictive modelling are two major data mining functionalities as defined by (Han & Kamber, 2001). In general, predictive modelling is defined as the process of making inferences from the current data in order to make predictions or classifications; descriptive modelling is the task of characterising the general properties of the data in a database. The following is a preview of the major tasks in data mining.

1.3.1 Predictive Modelling

The main aim of predictive modelling is to make predictions about values of data using known results or based on other historical data. Example applications include credit card fraud detection, breast cancer early warning system, terrorist acts and tsunami alerts. Several techniques for predictive modelling tasks are classification, prediction and outlier detection.

1.3.1.1 Classification

Classification is the act of classifying a set of input data with unknown class labels according to the correct class label. This involves two steps as defined by (Han & Kamber, 2001). The first is the model building step where the set of data which consist of input data and predefined class label are analyzed. This set of data is also referred to as the training dataset. The training data is used to build a classifier model that can map the input attributes into one of several discrete classes. This common characteristic is recognised as the supervised learning (Roiger & Geatz, 2003). Next, the input data with unknown class labels (the testing dataset) are fed into the classifier to determine their correct group as accurately as possible. The classification task is crucial in order to build a model that is capable of assigning new instances to one of the well-defined classes. For instance, a group of medical practitioners could develop a classification model to identify and classify previously unseen patients into

two groups, one of which will be the group likely to develop the cancer. Hence, this will create an early warning system for the more susceptible patients who then have the option of taking preventative measures.

1.3.1.2 Prediction

(Roiger & Geatz, 2003) agree that the function of the prediction model in the data mining task is to determine the future/new outcome(s) rather than discover current behaviours. Additionally, the outcomes of the prediction model may be either categorical or numerical values as opposed to those of the classification task. For example, we may want to build a model to predict the sales volumes of a new product, or the number of students enrolling in certain courses offered per semester. While, there are slight differences between the classification and prediction tasks, the determination of whether a model is suitable for classification or prediction depends on the nature of the data used (Roiger & Geatz, 2003).

1.3.1.3 Outlier Analysis

Real-world data often varies in terms of complexity which presents further challenges to the data mining and knowledge discovery process. Moreover, such complexities may arise because of the existence of data objects that do not comply with the general behaviour or model of the data (Han & Kamber, 2001) and have been recognised as *outliers*. Usually, the *outliers* may occur because of the noise or because they are truly exceptional cases. Outlier analysis is important in certain domains such fraud detection, network intrusion detection, credit risk assessment, terrorist attack and some financial and marketing applications. These applications usually look for uncommon events and rare behaviours as this can provide extensional knowledge for further analysis.

1.3.2 Descriptive Modelling

The descriptive modeling serves mainly to identify patterns or relationships in data. It provides a way to explore the properties of data examined, not to predict new properties. Generally, no proper group or predefined class label is known in advance. Clustering, sequence discovery and association rules are examples of descriptive

modeling applications. While predictive modelling provides data with predefined class labels for analysis, descriptive modelling is far more challenging as the class label for each object is unknown (Han & Kamber, 2001).

1.3.2.1 Clustering

Clustering is the task of composing a set of data into natural subclasses or groups. Such cluster formations may be based on the collection of patterns that are similar to one another, and dissimilar to patterns in other clusters (Sestito & Dillon, 1994). The aim is to maximize the similarity of data objects within a cluster (intra-cluster similarity) and minimize the similarity of data objects belonging to different clusters (inter-cluster similarity) (Han et al., 2012). Market segmentation and customer profiling are essential applications that greatly benefit from clustering analysis (Collica, 2007).

1.3.2.2 Association Analysis

Association analysis is a popular data mining technique aimed at discovering novel and interesting relationships between the data objects present in a database. Market basket or transaction data analysis is prominent association rules analysis that has attracted great interest from the data mining community. Association rule mining involves two important problems as defined by (Agrawal, Imieliski, & Swami, 1993), namely: frequent patterns discovery and rule construction. Frequent pattern discovery poses the greater challenge. Thus, many algorithms have been proposed for the efficient mining of frequent patterns (Han & Kamber, 2001). A simple yet popular example of this association rule mining task is to find that an association between customers buying a certain product will result in increase of or change in the customers buying certain other products.

1.3.2.3 Sequence Mining

Sequence mining is aimed at discovering frequently occurring ordered events or sub-sequences from a database consisting of ordered items or events, with or without a time stamp (Han et al., 2012). Examples include the analysis of weather prediction,

telecommunication records for customer retention and targeted marketing, web traversal sequence and mining the sequential patterns from protein data included in a DNA database. The discovery and generation of sequence patterns present more difficult problems (Agrawal & Srikant, 1995; Han et al., 2012), because of the high volume of data in databases due to time-volatile and complex applications.

1.4 Types of Data

The data mining system can be classified into various categories such as the types of data, structures and knowledge to discover (Han & Kamber, 2001). In this section, an overview is provided of four most commonly encountered data types in the data mining process, namely relational, sequential, semi-structured and unstructured data.

1.4.1 Relational Data

Initially, many data mining tools required input data to be in a flat file such as relational data format which is presented in two dimensional tables of rows and columns. The schema of the data is fixed and structured. Each column has names called ‘attributes’ and values related to each attribute. There are many efficient and successful developments in mining the relational data (Han & Kamber, 2001). (Han, Cheng, Xin, & Yan, 2007) assert that tremendous progress has been made in methods for discovering useful patterns particularly for relational data, especially with the development of the association rule mining application.

1.4.2 Sequential Data

Another advanced relational data type that emerged due to the progress of database technologies in addressing new and complex applications is the sequential data. It stores sequences of ordered events with which a notion of time may or may not be associated (Han et al., 2012). Time series data as described in (Hand, Mannila, & Smyth, 2001) are a popular instance of sequential data, in which a sequence of events is measured over time, such that each event is indexed by time variable *t*. However, as asserted in (Han et al., 2012), the notion of sequence data might not just be

restricted to the function of time, but can further extend into a sequence of ordered events such as customer shopping sequence and biological sequence.

1.4.3 Semi-structured Data

With the rapid growth in the amount of electronic data such as Web pages and XML data, this offers a new dimension in pattern recognition and rules discovery. These electronic data are a heterogeneous collection of ill-structured data that have no rigid structures, and are often referred to as 'semi-structured data' (Suciu, 1998; Zhang, Ling, Bruckner, Tjoa, & Liu, 2004). Due to the complex structures and semantics representation, the semi-structured data poses more challenges to the data mining process compared to structured and relational data (Feng, Dillon, Weigand, & Chang, 2003).

1.4.4 Unstructured Data

Unstructured data is the most difficult and challenging for data mining purposes. This data type has no schema that can describe the underlying structure of the data compared to relational and semi-structured data. Examples of such data include the data from audio, video and unstructured texts such as the body of email or word processor documents. Due to the lack of structure available to be exploited in the data mining process, the development of a powerful yet computationally efficient data mining algorithm for unstructured data is an important and challenging problem (Madria, Bhowmick, Ng, & Lim, 1999).

1.5 Motivation for the Thesis

Whilst there are many data mining techniques available for discovering hidden patterns and rules, each of the techniques differs in term of objectives, outcomes and representation techniques. In response to this, (McGarry, 2005) claims that the majority of data mining/machine learning type patterns are rule-based in nature with a well-defined structure such as rules derived from decision trees and association rules. (Geng & Hamilton, 2006) agrees and indicates that the most common patterns that can be evaluated by interestingness measures include association rules,

classification rules, and summaries. Furthermore, (McGarry, 2005) states that rule-based patterns are composed of a number of primitive patterns (antecedents) connected by logical connectives that imply, if true, the target class (consequent).

Association rule discovery is a problem that has generated a lot of interest in the data mining community. The techniques are used for discovering interesting associations and correlations between data elements in a diverse range of applications. Obviously, association rule mining has been very successful in discovering useful associations between data (Agrawal et al., 1993; Han et al., 2012).

The main problems in association rule discovery are frequent pattern and rules construction. Frequent pattern discovery is the pre-requisite for and the first step in the generation of association rules. Approaches such as candidate generation, frequent pattern growth (FP-Growth) have been proposed to discover frequent patterns (Han & Kamber, 2001). While there have been many association rule algorithms proposed in the data mining literature and successfully used in many application, there are still situations that these rules fail to capture (Zhang, Balaji, & Alexander, 2004). Rule generation also produces a large number of rules (Hand, 1998) and it is impossible for an expert in the field being mined to sustain the rules (Lenca, Meyer, Vaillant, & Lallich, 2008). The problem in sustaining the interestingness of rules generated by a data mining algorithm is an active and important area of data mining research. Different methods have been proposed for discovering interesting rules from data such as support and confidence (Agrawal et al., 1993), lift/interest (Aggarwal & Yu, 1998; Silverstein, Brin, & Motwani, 1998), chi-squared test (Silverstein et al., 1998), correlation coefficient (Brijs, Vanhoof, & Wets, 2003), log linear analysis (Brijs et al., 2003), leverage (Webb, 2007), and empirical bayes correlation (Brijs et al., 2003).

While great progress has been made regarding the discovery of association rules within well-structured (relational) data, a number of works are still in the preliminary stages concerning semi-structured data (Chi, Muntz, Nijssen, & Kok, 2005). Since the introduction of the association rule mining problem by (Agrawal et al., 1993), substantial work has gone into various trends, including the development of efficient algorithms to find the associations and measure the interestingness of the association

rules in relational data. As the increase in data captured in semi-structured format such as XML begins to permeate many applications (Mignet, Barbosa, & Veltri, 2003), association rule mining from semi-structured data has become an important research area (Braga, Campi, Ceri, Klemettinen, & Lanzi, 2003; Hadzic, Tan, & Dillon, 2011; Tan, Hadzic, Dillon, & Chang, 2008). Similar to the problem of frequent pattern mining from relational/transactional data, mining the frequent subtrees from semi-structured data comprises of candidate subtrees enumeration and frequency counting. Works such as (Asai et al., 2002; Feng et al., 2003; Tan, Hadzic, Dillon, Chang, & Feng, 2008; Zhang et al., 2004) have developed algorithms to enable efficient and effective association rule mining from semi-structured data.

A rule is said to be interesting if, in addition to meeting certain minimum support and confidence criteria, it also satisfies some measure of interestingness. Despite the fact that interesting association rules may be found in such databases, the possibility remains that they reflect only the database being observed. (Webb, 2007) emphasizes that each assessment of whether a given rule satisfies certain constraints is accompanied by a risk that the rule will satisfy the constraints with respect to the sample data but not with respect to the whole data distribution. This problem arises because some association rules are discovered due to pure coincidence resulting from certain randomness in the particular dataset being analysed. In addition to that, (Hand, 1998, 1999) agrees that the real requirement is to consider how many of the discovered rules are real rather than chance of fluctuations in the database. Therefore, we still can argue the validity of the rules and patterns to be used in practical problems. Since the nature of data mining techniques are data-driven, the patterns generated by these algorithms should be further validated by statistical methodology in order for them to be useful in practice (Goodman, Kamath, & Kumar, 2008). On the other hand, data mining techniques are automated and are scalable and effective for finding associations between large numbers of variables, while statistical techniques can address only a small number of variables. It is therefore imperative to combine the benefits of both approaches and establish the ways in which the existing statistical measures and constraint parameters can be effectively utilized, and the sequence of their usage. This research employs the science and engineering research paradigm to develop a framework to measure and

verify the usefulness of rules from association rule mining techniques using statistical analysis, and redundancy and contradictive assessment methods.

1.6 Scope of the Thesis

The goal of this thesis is to develop a unified framework for evaluating the rules developed in association rules mining of relational and semi-structured data. Note that within the semi-structured data, the focus is on XML documents and tree-structured data in general. Generally, rules can be discovered by many data mining functionalities such as the concept/class description, classification and prediction, cluster analysis, outlier analysis, data evolution analysis and sequential analysis (Han & Kamber, 2001); however, these are outside the scope of this thesis. From the perspectives mentioned above, the main interest is mainly in developing a unified framework to evaluate the association rules involving frequent itemsets for relational data and frequent subtrees for tree-structured data. The objectives of the thesis are outlined in more detail as follows:

Develop a framework to measure and verify the interestingness and statistical significance of rules obtained from association rules mining techniques using the proper sequence of statistical analysis, redundancy and contradictive assessment methods, to reduce the size of the rule set, while preserving/improving the rules coverage and accuracy with the following sub-goals:

- Obtain a compact set of rules that have better accuracy and coverage than the original large set of rules.
- Identify proper sequence of applying appropriate measures for use in association rules mining quality measurement.
- Develop a unified model for evaluating the quality of association rules using statistical analysis, and measures of redundancy and contradiction.
- Evaluate the implications of basing association rules on frequent, maximal and closed itemsets when used for classification tasks in relational data.
- Evaluate the implications of basing the association rules on different subtree types when used for classification tasks in tree-structured data.

The proposed research is significant because it:

- Offers a way to reduce the rules generated from the association rules mining process that are commonly overwhelming and impractical for use.
- Identifies and provides suitable statistical techniques and interestingness measures for accessing the association rules.
- Offers a systematic way to verify the usefulness and statistical validity of rules obtained from association rules mining using statistical analysis and appropriate measures.
- Details the sequence of usage of appropriate measures to arrive at a more reliable and interesting set of rules.
- Provides a framework that can arrive at high quality rules by:
  a) removing rules that were randomly and coincidently generated from the database
  b) removing redundant rules
  c) removing contradictive rules
  d) choosing suitable *confidence* thresholds
- Identifies the implication and the characteristics of rules based on frequent, maximal and closed patterns for relational data, and rules based on different subtree types for tree-structured data, when used for classification tasks.

1.7 Plan of the Thesis

The thesis is organized into 8 chapters.

In Chapter 2, the existing works on association rules mining are discussed beginning with an overview of the theoretical basis of topics related to association rules mining, and interestingness measurement techniques for both relational and semi-structured data. The aim of this chapter is to outline the achievements of the interestingness measures for relational data and semi-structured data. Existing techniques will be examined. Additionally, in one subsection we overview the relationship between feature subset selections with rule interestingness. We then conclude this chapter by highlighting the current difficulties and challenges that need to be addressed.

Chapter 3 focuses on describing the general concepts and definitions; and problem definition that serves as a basis for discussion of subsequent chapters. The chapter starts with general definitions of relational and semi-structured data. The definitions of association rules mining for relational and semi-structured data are elaborated. The problems of evaluating the interestingness of frequent itemset and frequent subtree are formulated. Finally, the chosen methodology used in this thesis in addressing the aforementioned problem is outlined.

Chapter 4 provides an overview of the proposed solution to the problems described in Chapter 3. The rationale for the choice of objective measurement technique is discussed. The overall proposed solution is divided into two parts: 1) association rule mining for relational data; and, 2) association rule mining for tree-structured data/XML documents.

Chapter 5 describes the development of a detailed solution to evaluate the association rules derived from relational data. Here, the details of the pre-processing task, data partitioning and the determination of relevant attributes will be described. A way to evaluate rules from association rule mining and its related measures will be formalized.

Chapter 6 evaluates the proposed framework and presents the experimental findings of significant rules from relational data. Evaluation of the proposed framework is performed using real-world datasets of different complexities obtained from UCI Machine Leaning Repository (Frank & Asuncion, 2010).

Chapter 7 is devoted to the development of a detailed solution to evaluate frequent subtrees generated from the tree-structured data. Here, an overview of tree-structured data and the issues pertaining to tree-structured data related to this thesis will be discusses. Within the same chapter, the evaluation of the proposed framework using three subtree types and datasets with different structural characteristics will be explained.

Chapter 8 recapitulates the research works undertaken in this thesis and provides insights into future works worth further exploration.

References

Aggarwal, C. C., & Yu, P. S. (1998). A new framework for itemset generation. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. Seattle, Washington, United States: ACM.

Agrawal, R., Imieliski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec., 22*, 207-216.

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on* (pp. 3-14).

Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H., & Arikawa, S. (2002). Efficient Substructure Discovery from Large Semi-structured Data. In R. L. Grossman, J. Han, V. Kumar, H. Mannila & R. Motwani (Eds.), *Second SIAM International Conference on Data Mining*. Arlington, VA, USA: SIAM.

Braga, D., Campi, A., Ceri, S., Klemettinen, M., & Lanzi, P. (2003). Discovering interesting information in XML data with association rules. In *Proceedings of the 2003 ACM Symposium on Applied Computing*. Melbourne, Florida: ACM.

Brijs, T., Vanhoof, K., & Wets, G. (2003). Defining interestingness for association rules. *International journal of information theories and applications, 10(4)*, 370-376.

Chi, Y., Muntz, R. R., Nijssen, S., & Kok, J. N. (2005). Frequent Subtree Mining - An Overview. *Fundamenta Informaticae, 66*, 161-198.

Collica, R. S. (2007). *CRM Segmentation and Clustering Using SAS Enterprise Miner*: SAS Press Series.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. In *AI Magazine* (Vol. 17): AAAI.

Feng, L., Dillon, T., Weigand, H., & Chang, E. (2003). An XML-Enabled Association Rule Framework. In *Database and Expert Systems Applications* (pp. 88-97).

Frank, A., & Asuncion, A. (2010). {UCI} Machine Learning Repository. In: University of California, Irvine, School of Information and Computer Sciences.

Frawley, W. J., Piatetsky-Shapiro, G., & Matheus, C. J. (1992). Knowledge Discovery in Databases: An Overview. In *AI Magazine* (Vol. 13): AAAI.

Geng, L., & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Comput. Surv., 38*, 9.

Goodman, A., Kamath, C., & Kumar, V. (2008). Data Analysis in the 21st Century. *Stat. Anal. Data Min., 1*, 1-3.

Hadzic, F., Tan, H., & Dillon, T. S. (2011). *Mining of Data with Complex Structures* (Vol. 333): Springer-Verlag Berlin Heidelberg.

Han, J., Cheng, H., Xin, D., & Yan, X. (2007). Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery, 15*, 55-86.

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

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: concepts and techniques* (3rd ed.). Waltham, Mass.: Elsevier/Morgan Kaufmann.

Hand, Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge, Mass.: MIT Press.

Hand, D. J. (1998). Data Mining: Statistics and More? *The American Statistician, 52*.

Hand, D. J. (1999). Statistics and data mining: intersecting disciplines. *SIGKDD Explor. Newsl., 1*, 16-19.

Lenca, P., Meyer, P., Vaillant, B., & Lallich, S. (2008). On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. *European Journal of Operational Research, 184*, 610-626.

Madria, S. K., Bhowmick, S. S., Ng, W. K., & Lim, E. P. (1999). Research Issues in Web Data Mining. In *DataWarehousing and Knowledge Discovery* (Vol. 1676, pp. 805-805): Springer Berlin / Heidelberg.

McGarry, K. (2005). A survey of interestingness measures for knowledge discovery. *Knowl. Eng. Rev., 20*, 39-61.

Mignet, L., Barbosa, D., & Veltri, P. (2003). The XML web: a first study. In *Proceedings of the 12th international conference on World Wide Web*. Budapest, Hungary: ACM.

Roiger, R., & Geatz, M. (2003). *Data mining: a tutorial-based primer*. Boston: Addison Wesley.

Sestito, S., & Dillon, T. S. (1994). *Automated knowledge acquisition*. New York: Prentice Hall.

Silverstein, C., Brin, S., & Motwani, R. (1998). Beyond Market Baskets: Generalizing Association Rules to Dependence Rules. *Data Min. Knowl. Discov., 2*, 39-68.

Suciu, D. (1998). Semistructured Data and XML. In K. Tanaka & S. Ghandeharizadeh (Eds.), *The 5th International Conference on Foundations of Data Organization (FODO'98)* (pp. 1-12). Kobe, Japan.

Tan, H., Hadzic, F., Dillon, T. S., & Chang, E. (2008). State of the art of data mining of tree structured information. *International Journal of Computer Systems Science and Engineering, 23*.

Tan, H., Hadzic, F., Dillon, T. S., Chang, E., & Feng, L. (2008). Tree model guided candidate generation for mining frequent subtrees from XML documents. *ACM Trans. Knowl. Discov. Data, 2*, 1-43.

Webb, G. I. (2007). Discovering Significant Patterns. *Machine Learning, Springer*, 1-33.

Zhang, H., Balaji, P., & Alexander, T. (2004). On the discovery of significant statistical quantitative rules. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Seattle, WA, USA: ACM.

Zhang, J., Ling, T. W., Bruckner, R. M., Tjoa, A. M., & Liu, H. (2004). On Efficient and Effective Association Rule Mining from XML Data. In *Database and Expert Systems Applications* (pp. 497-507).

CHAPTER 2: LITERATURE REVIEWS

2.1 Introduction

This chapter discusses previous work on the interestingness and validity of rules/patterns from association rule mining. Thus, the main goal of this chapter is to present the current state-of-the-art of the existing research literature in the areas of rules interestingness measures with association rule mining as the case point in both relational and semi-structured data.

The chapter outline is as follows. General work in association rule mining will be discussed in the next section. Then, in Sections 2.3 and 2.4 the discussion focuses on mining frequent itemsets from relational data and mining frequent subtrees from semi-structured data. This chapter narrows down the focus of the interestingness measures for association rule mining in Section 2.5. Section 2.6 focuses on the classification of interestingness measures with special attention given to the objective-based measures. Section 2.7 discusses the existing works on interestingness measures for semi-structured data. Additionally, the discussion on the relationship between feature subset selection and rules interestingness is presented in Section 2.8. The chapter concludes with a summary of the open issues.

2.2 Association Rules Mining (General)

Promotional pricing, shelf space plans and product placements are several application benefits derived from the development of association rule mining analysis. Association rules mining has been widely used and successfully implemented for discovering useful associations between data in a large database (Aggarwal & Yu, 1998; Agrawal, Imieliski, & Swami, 1993; Agrawal & Srikant, 1994; Han & Kamber, 2001; Toivonen, 1996).

Association rule mining is capable of finding useful information from a large transactional database. An example is the market basket analysis, which as defined by (Han & Kamber, 2001), is a process of gaining insightful information about customer buying behaviours in order to discover interesting and useful buying patterns.

This can be done by carefully examining customer buying behaviour and then placing each item(s) correctly as this will trigger costumer interest in buying additional item(s) rather than buying a single item (Han & Kamber, 2001). In this context, the aim is to capture the association relationship within costumer transactions. That is, if after deciding to buy a certain item(s), the customer is then more or less likely to buy another item(s). This can be done by placing frequently associated items close together as they are most likely to be bought together. This will increase sales and benefit the company.

2.2.1 Basic Concepts

In general, the association rule mining searches for interesting relationships among items in a given data set under minimum support and minimum confidence conditions.

The problem of finding association rules $x \to y$ was first introduced in (Agrawal et al., 1993) as a data mining task of finding frequently co-occurring items in large databases. (Agrawal et al., 1993) developed a two-phase approach to the association rules problem. The first step is to find all frequently occurring items, typically referred to as frequent itemsets (Lenca, Meyer, Vaillant, & Lallich, 2008). Each of the itemsets will occur at least as frequently as a predetermined minimum *support* count. The second step is to generate strong association rules from the frequent itemsets. These strong rules must satisfy the minimum *support* and minimum *confidence*.

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of items. Let *D*, be a transactions database for which each transaction T is a set of items, such that $T \subseteq I$. An association rule is a condition of the form of $x \to y$ where $x \subseteq I$ and $y \subseteq I$ and $x \cap y = \phi$. The *support* of a rule $x \to y$ is the number of transactions that contain both x and y. Let the *support* (or *support* ratio) of rule $x \to y$ (denoted as $\sigma(x \to y)$) be $s\%$. This implies that there are $s\%$ transactions in *D* that contain items (itemsets) x and y. In other words, the probability $P(x \cup y) = s\%$. Sometimes, it is expressed as *support count* or *frequency*, that is, it reflects the actual frequency count of the number of transactions in *D* that contain the items that are in the rules. An itemset is *frequent* if it satisfies the user-

specified *minimum support* threshold. The *confidence* of a rule $x \to y$ is the conditional probability of a transaction containing the *consequent* (y) if the transaction contains the *antecedent* (x). Hence, the *confidence* of a rule $x \to y$ is calculated as $\sigma(x \to y) \ / \ \sigma(x)$.

(Aggarwal & Yu, 1998) assert that the idea of the association rule is to develop a systematic method by which user can figure out how to infer the presence of some sets of items, given the presence of other items in a transaction. Such information is useful in making decisions such as shopper targeting, shelf spacing and sales promotions. Since the introduction of the association rule mining problem by (Agrawal et al., 1993), substantial work has gone into various trends, including the development of efficient and scalable algorithms for finding the association rules (Aggarwal & Yu, 1998; Agrawal & Srikant, 1994; Mannila, Toivonen, & Verkamo, 1994; Toivonen, 1996) and measuring the interestingness of the association rules in relational data (Bayardo, Agrawal, & Gunopulos, 2000; Jaroszewicz & Simovici, 2001; Lavrač, Flach, & Zupan, 1999; Lenca et al., 2008; Tan, Kumar, & Srivastava, 2002; Webb, 2007; Yun, Ha, Hwang, & Ryu, 2003).

2.2.2 Types of Association Rules

There are various types of association rules, and in (Han & Kamber, 2001) these association rule types were classified according to four major criteria.

The first and second criteria are based on the types of data handled by the rules and dimensions of data involved in the rule set. (Fukuda, Morimoto, Morishita, & Tokuyama, 1996; Ke, Cheng, & Ng, 2008; Moreno, Segrera, Lopez, & Polo, 2006; Piatetsky-Shapiro, 1991; Srikant & Agrawal, 1996) are prominent works that discuss the association rule based on types of data namely, the Booleen Association Rules and the Quantitative Association Rules (QAR). Both the Boolean and the Quantitative Association Rules may refer to either the single attribute or the multiple attributes in both the antecedent and the consequent. (Han & Kamber, 2001) extensively studied the multi-dimensional association rules. These are the rules that imply more than one dimension or predicate. In general, the discretization techniques such as the

clustering, *a set of base interval*, *concept hierarchy* and *equal-depth bucket* are employed in order to handle the quantitative attributes and the multi-dimensional rules. Such actions can reduce the large number of distinct values of quantitative attributes; however, at the same time, there will be loss of information due to the discretization processes.

The third type of association rule is based on the levels of abstraction involved in the rule set. The discovery of multi-level abstraction rules have been successfully implemented by (Han & Fu, 1995; Srikant & Agrawal, 1995). Also, this multi-level approach is capable of finding redundant rules as defined in (Han & Kamber, 2001)

The maximal pattern and closed pattern are various examples that extend the basic form of association rule mining approaches. Their capabilities in reducing a large volume of frequent patterns to a smaller set of rules while preserving the analytical power, has attracted extensive research on the area of maximal and closed patterns. Detailed discussion of maximal and closed itemset mining is given in Section 2.3.2.

2.3 Mining Frequent Itemsets (Relational Data)

2.3.1 Apriori Algorithm

Mining frequent itemsets in a large transaction dataset is considered as a difficult task. Such conditions occur due to the enumeration of all distinct and single items in the database. (Agrawal & Srikant, 1994) introduced the *Apriori* algorithm to overcome this problem whereby an iterative approach known as a *level-wise* search is employed to generate the frequent itemsets. This approach first scans the *k*-itemsets from the database, then uses the *k*-itemsets to generate candidates for the (*k+1*)-itemsets, and checks the database again to obtain the frequent (*k+1*)-itemsets. The scanning and the generation process iterates until there are no more *k*-itemsets to be found (Han & Kamber, 2001).

An *Apriori* property called *downward closure* is utilized to improve the level wise frequent itemsets generation. The *downward closure* property (Agrawal & Srikant, 1994), defines the frequent *k*–itemsets as follows; A *k*-itemset is frequent only if all of

its sub-itemsets are frequent". An example derived from (Han & Kamber, 2001) is: if an itemsets *I* does not satisfy the minimum support threshold, *min_support*, then *I* is not frequent, that is $P(I) < min\_support$. If an item *A* added to the itemsets *I*, then the resulting itemset (i.e., $I \cup A$) cannot occur more frequently than *I*. Therefore, $I \cup A$ is not frequent either, that is $P(I \cup A) < min\_support$.

The *Apriori*-based algorithms have been favourable and are popular techniques used in pattern mining. They perform well on sparse data in discovering frequent patterns. The majority of the association rule algorithm evolution is related to and centred on the *Apriori* approach. The hash-based technique, transaction reduction, partitioning, sampling and dynamic itemset counting are variations of association rule mining techniques that attempt to improve the efficiency of the *Apriori* algorithm (Han & Kamber, 2001).

The *Apriori*-based algorithm is one of the most recognized frequent itemset mining algorithms. However, there are other algorithms such as the *frequent-pattern growth-based* algorithm and algorithms that use the vertical format which are also efficient and scalable for mining frequent itemsets (Han, Kamber, & Pei, 2012). Two prominent methods for each of the aforementioned algorithms are FP-Growth (Han, Pei, & Yin, 2000) and ECLAT (Zaki, 2000b). The FP-growth method extracts the frequent itemsets without candidate generation and this can be done by constructing a compact form of the database utilizing the FP-tree structure (Han & Kamber, 2001). The ECLAT, as described in (Han & Kamber, 2001) manipulates a given dataset of transactions in the horizontal data format into the vertical data format.

2.3.2 Maximal and Closed Frequent Itemset Mining

Mining a complete set of rules using the *Apriori* algorithm will ensure the enumeration of all frequent items that satisfy certain thresholds. However, this will create problems such as the generation of a large volume of patterns. Maximal frequent itemsets mining and Closed frequent itemsets mining were proposed and successfully developed as one of the ways to counter this problem (Han, Cheng, Xin, & Yan, 2007). A frequent itemset is called a maximal frequent itemset if it is not a subset of any other frequent itemset. A closed frequent itemset is a frequent itemset of

which no proper superset has the same support count (frequency). All frequent itemsets can be generated from a set of maximal or closed patters, and in addition for closed patterns the exact support information for each itemset can be worked out.

For an illustrative example, a transactional database from (Zaki & Hsiao, 2002) is reproduced, readapted and named it as the 'Sport Shoes' transaction database as shown in Figure 2.1. There are seven different items, $I = \{A, B, C, D, E, F, G\}$ and eight transactions T_{db} (n=8). Table C shows that all 17 frequent itemsets are present in at least four transactions; i.e., *minsup* = 50%. Figure 2.2 shows the 17 frequent itemsets organized as a subset lattice; with their corresponding *tidsets* shown. The 8 closed sets are obtained by collapsing all the itemsets that have the same *tidset*, shown in the figure with the circled regions. Hence, referring to Figures 2.1 and Figure 2.2, ABC, ACD, ACE, AF, AB, AC, AE and A are identified as *Closed* itemsets. As indicated on the top of the *Closed* itemset lattice, there are 4 *maximal* frequent itemsets (marked with a circle), ABC, ACD, ACE and AF (as these 4 *maximal* itemsets are not a subset of any other frequent itemsets)

Sport Shoes Transactions Database Items (Table A)

Adidaz	Brookr	Crocz	Dunlap	Esics	Fuma	Gizuno
A	B	C	D	E	F	G

Database (Table B)

**Trans.	Items
1	ABCDEF
2	CDEAB
3	DCBA
4	ABC
5	BEFA
6	BFAG
7	CDEA
8	CAEFG

**Transaction

Table C

*Support	Itemsets
100%	A
75%	AB,B,C,AC
62.50%	AE,E,
50%	ACD,AF,ACE,D, BC,F,AD,ABC,CE, CD

* Minimum support =50%

Figure 2.1: Example Sport Shoes Transactions Database

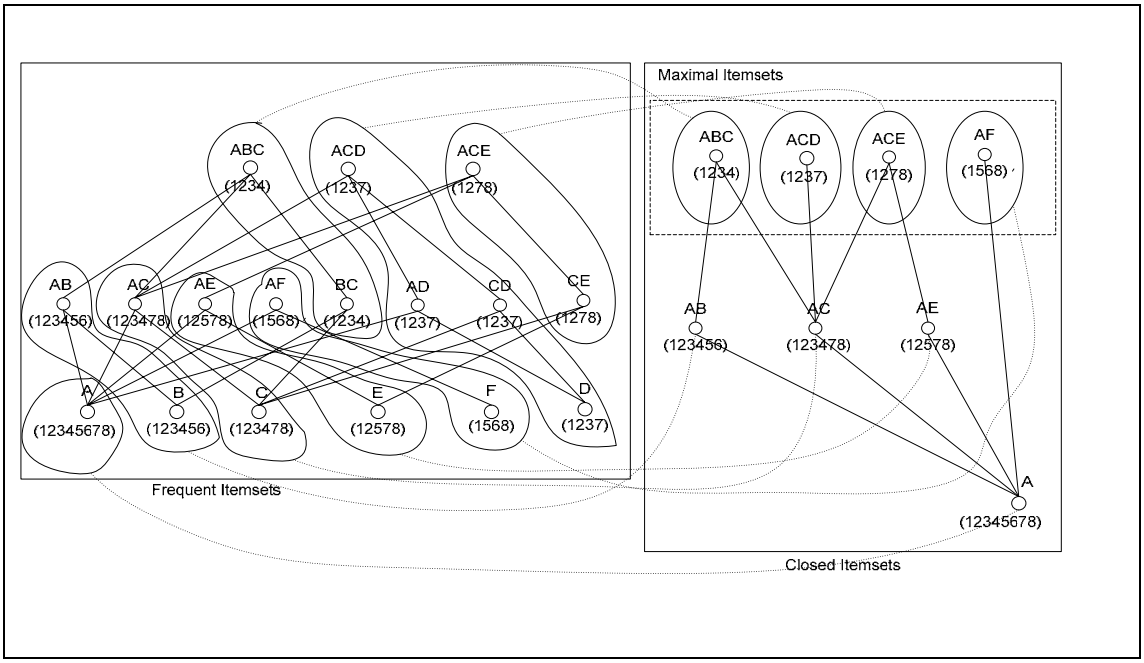


Figure 2.2: Frequent, Closed and Maximal Itemsets at support = 50 %

(Bayardo, 1998) successfully developed an efficient way of mining long patterns from a database by extracting only the maximal frequent itemsets. The *MaxMiner* (Bayardo, 1998) algorithm reduces the itemsets through superset frequency-based pruning. The main characteristic of *MaxMiner* is its capability to identify long frequent itemsets at an early stage of the searching process. *GenMax* (Gouda & Zaki, 2001) is another notable method for mining maximal frequent itemsets. The *GenMax* approach employs backtracking search for efficiently enumerating all maximal patterns.

Additionally, *MAFIA* method proposed in (Burdick, Calimlim, Flannick, Gehrke, & Yiu, 2005) is claimed to outperform the *GenMax* method in terms of performance when mining long itemsets and dense data. In the *MAFIA* method, the vertical bitmap representation and compression mechanism is utilized for counting and pruning in order to search the itemset lattice.

A-Close, an *Apriori*-based algorithm used to discover the closed frequent itemsets, was successfully developed by (Pasquier, Bastide, Taouil, & Lakhal, 1999). As the *A-Close* framework focuses only on the discovery aspect of closed itemsets, (Zaki, 2000a; Zaki & Hsiao, 2002) proposed a notable algorithm, namely the *Charm* algorithm. This algorithm is claimed to outperform the *A-Close* (Pasquier et al., 1999)

in terms of their rules presentable to user and time optimization. The *Closet* algorithm of (Pei, Han, & Mao, 2000) is another algorithm for mining closed itemsets which involves three techniques: firstly, apply the frequent pattern tree FR-tree structure for mining closed itemsets without candidate generation; secondly, develop a single prefix compression and finally, explore a partition-based projection mechanism. These three steps offer an efficient and scalable capacity for the *Closet* algorithm compared to the *A-Close* algorithm and the *Charm* algorithm.

Overall, the capacity to preserve the same analytical power as mining the whole set of frequent itemsets, proves to be an advantage of the closed algorithm compared to the maximal; while closed has an advantage over the *Apriori* algorithm when dense and complex datasets are being considered (Zaki, 2000a; Zaki & Hsiao, 2002).

On examining a number of algorithms for mining itemsets from relational data in Section 2.2 to Section 2.3, it was found that the main strength of the aforementioned algorithms is their capability of finding comprehensive cases and patterns. While one could argue that the discovery of important comprehensive rules might be unrealistic and ineffective (Han & Kamber, 2001), a pre-determined threshold plays an important role in alleviating this issue. In addition, the association rule algorithms have to scale-up in many applications such as dense and sparse databases. The introduction of algorithms such as Frequent Pattern-Growth (FP-Growth), multi-level association rules, and *Maximal* and *Closed* itemsets offers a broad area to utilize the association rule mining algorithms.

While association rule mining techniques have been successfully utilized, in many cases certain aspects of domain knowledge will not be completely captured by the extracted rules (Zhang, Balaji, & Alexander, 2004). Another problem is that the rule sets are often too large and complex, thereby making it impractical or impossible for a domain expert to analyze them in an efficient manner (Hilderman & Hamilton, 2001; Lenca et al., 2008).

Additionally, the main weakness as discussed in the literature is the existence of many uninteresting rules, even when the algorithms were supplied with certain support and confidence thresholds. The fact is that they are unable to discriminate between the

significant, interesting rules, and uninteresting or misleading rules. Purely coincidental and random association rules may still exist due to these problems. One way to resolve this is by developing rules interestingness measures (Han & Kamber, 2001). The importance of interestingness measures in discovering the association rules is discussed in Section 2.5 and Section 2.6.

2.4 Mining Frequent Subtrees (Semi-structured Data)

With the fast growth in the amount of electronic data in the form as Web pages and XML data, this offers a new dimension in pattern recognition and rules discovery. These electronic data are heterogeneous collections of ill-structured data that have no rigid structures, and are often referred to as semi-structured data (Suciu, 1998; Zhang, Ling, Bruckner, Tjoa, & Liu, 2004). As the increase in data captured in semi-structured format such as XML begins to flood many applications, association rule mining from the semi-structured data has become a new and interesting research area (Braga, Campi, Ceri, Klemettinen, & Lanzi, 2003; Chen, Bhowmick, & Chia, 2004; Hadzic, Tan, & Dillon, 2011; Zaki, 2005).

Driven by the vast amount of XML applications being exploited in domains such as finance, banking, bioinformatics, biomedical, information technology and sciences and the successful mining of frequent itemsets, many algorithms have been developed to mine XML documents. (Suciu, 1998) assert that, in order to fully utilize and analyse the XML data, the right tools and data format are needed. Thus, the traditional mining of associations among simple-structured items of atomic values had to be extended to detect associations among XML fragments the underlying structure of which is a tree (Braga et al., 2003; Feng, Dillon, Weigand, & Chang, 2003). A detailed explanation of this problem and the parallelism between XML and tree-structured is provided in Chapter 3. As the XML documents can be effectively modelled as a rooted ordered labelled tree (as illustrated later in Chapter 3), the development of frequent subtree mining algorithms has been the main focus to enable association rule discovery from XML. A formal definition of the frequent subtree mining problem will be provided in Chapter 3. Next, a brief overview of some existing algorithms is provided.

2.4.1 Frequent Subtree Mining

Increasing complexity of data in the form of XML, has posed a greater challenge for frequent pattern mining tasks. Frequent subtree mining as discussed by (Chi, Muntz, Nijssen, & Kok, 2005) is relatively more complicated compared to frequent itemset mining, due to the structural aspects that need to be taken into account. Additionally, much progress has been made in approaches for discovering association rules within the well-structured data. As a consequence, many frequent subtree mining techniques have been developed based on frequent itemset mining techniques.

The general problems of association rule mining include the extraction of all the frequent itemsets from which association rules are formed. As described earlier in Chapter 1, a rule is said to be interesting if, in addition to meeting certain minimum support and confidence criteria, it also satisfies the measure of interestingness. The same holds for mining frequent subtrees in semi-structured data which requires candidate subtree enumeration and frequency counting.

Different algorithms have been proposed for mining different subtree types using different constraints and support definitions. An overview of the current state-of-the-art methods in this field of study can be found in (Chi et al., 2005; Hadzic et al., 2011; Tan, Hadzic, Dillon, & Chang, 2008). The popularity of string-like representation was recognized by (Chi et al., 2005; Tan et al., 2008; Zaki, 2005), as an effective way of capturing the hierarchical information of trees. As such, it provides a better way of data manipulation and space efficiency.

In general, the ordered induced subtree preserves the order of sibling nodes in the original subtree and the relationships between nodes positioned vertically in the subtree are limited to the parent-child relationship. On the other hand, ordered embedded subtree types extend the vertical node relationship in the induced subtree to be ancestor-descendant. A detailed explanation of induced and embedded subtrees is provided in Chapter 3. (Chi et al., 2005; Hadzic et al., 2011; Zaki, 2005) have summarized several algorithms which utilized the induced and embedded data types. FREQT(Asai et al., 2002), AMIOT (Hido & Kawano, 2005), IMB3-Miner (Tan, Dillon, Hadzic, Chang, & Feng, 2006) and TreeMiner (Zaki, 2005) are several

approached proposed to mine induced and embedded subtrees. Moreover, other subtree types to be mined are unordered induced and embedded. Mining unordered subtrees is important when the order of sibling nodes in the original tree is considered as irrelevant or is not known. Hence, in an unordered subtree, the order of sibling nodes can be exchanged and it is still considered as the same candidate subtree (Chi et al., 2005; Hadzic et al., 2011). This increases the complexity of the subtree mining process, as each enumerated ordered subtree candidate needs to first be ordered into a standard representative form so that all variations of the subtree with respect to the sibling node order are correctly considered as a single entity. To date, several approaches have been developed that mine unordered subtrees such as Unot (Asai et al., 2002), RootedTreeMiner (Yun, Yirong, & Richard, 2005), HybridMiner (Yun, Yirong, & Richard, 2004), TreeFinder (Termier, Rousset, & Sebag, 2002) and SLEUTH (Zaki, 2004).

As seen in this section, a number of algorithms have been successfully developed to mine different subtree types. While the aforementioned frequent subtree mining techniques may discover interesting associations from a given XML dataset, the problem that remains and that was inherited from traditional association rules mining is that they might reflect aspects only of the database being observed**.** Some subtrees are discovered due to pure coincidence resulting from certain randomness in the particular dataset being analyzed and the algorithms developed being unable to recognize and discriminate between interesting and significant subtrees and purely random association. Therefore, a proper appropriate framework needs to be developed for evaluating the interestingness of subtrees. The importance of interestingness measures for subtree rules will be discussed in Section 2.7, although the focus of the work in this thesis is limited to the interestingness measures for ordered subtrees.

2.5 Interestingness and Validity of Rules

Association mining is a useful technique for discovering interesting rules and patterns from large quantities of data. However, the association rule mining algorithms often tend to generate a large volume of rules. This can cause great difficulties in the analysis and interpretation of results, and it becomes rather impractical for a domain expert to utilize the rules for decision support purposes (Hilderman & Hamilton,

2001; Lenca et al., 2008). Thus, determining which of these patterns are useful can be very challenging. Figure 2.3, (McGarry, 2005) illustrates three main techniques for pattern assessment.

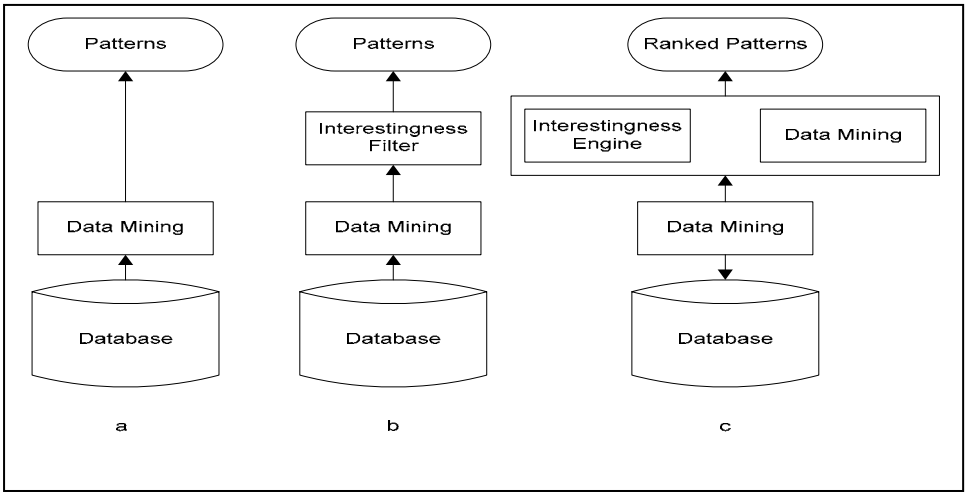


Figure 2.3: Techniques for Knowledge Discovery. Figure 2.3(a) shows that all patterns produced by the data mining process are passed to the user. Figure 2.3(b) shows how the search for interesting patterns occurs as a post-processing effort. Figure 2.3(c) shows the method to integrate the search for interesting patterns within the data mining algorithm reproduced from (McGarry, 2005)

Different criteria have been used to limit the nature of rules extracted, such as support and confidence (Agrawal et al., 1993), collective strength (Aggarwal & Yu, 1998), lift/interest (Silverstein, Brin, & Motwani, 1998), chi-squared test (Silverstein et al., 1998), correlation coefficient (Brijs, Vanhoof, & Wets, 2003), three alternative interest measures any-confidence, all confidence, and bond (Omiecinski, 2003), log linear analysis (Brijs et al., 2003), leverage (Piatetsky-Shapiro, 1991; Webb, 2007) and empirical bayes correlation (Brijs et al., 2003).

Although there are various criteria for determining the usefulness of rules (Geng & Hamilton, 2006; Han & Kamber, 2001; Hilderman & Hamilton, 1999; Lavrač et al., 1999), the rules reflect only the database being captured. One never knows whether the rules produced are useful in practice or are valid for a real-world problem. Applying a data mining algorithm to practical problems is not sufficient because one needs to ensure that the results have a sound statistical basis. Therefore, in this research, the quality of data mining rules will be verified by statistical analysis, and

redundancy and contradictive assessment methods. Such unification is sorely needed to overcome the data overload in practical problems (Goodman, Kamath, & Kumar, 2008). Table 2.1 summarizes the upside and downside of the data mining and statistic approaches as described in (Goodman et al., 2008). Thus, additional measures based on statistical independence and correlation analysis are needed to ensure that the results have a sound statistical basis and are not purely random coincidence.

Table 2.1: Data mining vs. Statistic reproduced from (Goodman et al., 2008)

Data mining	Statistic
Upside	**Upside**
Capable of exploiting results for a massive volume of data	Capable of make sense out of data
Downside	**Downside**
Results may need to be validate through statistic analysis	Analysis must be rigorously up to the level of data volume

2.6 Classification of Interestingness Measures

Measuring the interestingness of discovered patterns is an active and important area of data mining research. Although much work have been done in this area, so far there is no well-known agreement on a formal definition of interestingness in this context (Geng & Hamilton, 2006). This assertion is related to (Tan et al., 2002), who state that there is no measure that is consistently better than others in all cases. Yet, several researchers (Geng & Hamilton, 2006; Han & Kamber, 2001; Lavrač et al., 1999) agree that *conciseness, generality, reliability, peculiarity, diversity, novelty, surprisingness, utility, actionability, coverage* and *accuracy* are the eleven criteria used to determine whether or not a pattern is interesting. (Geng & Hamilton, 2006) also highlight, that the interestingness criteria are sometimes correlated, rather than independent of one another. Table 2.2 summarizes the interestingness criteria as described in (Geng & Hamilton, 2006; Lavrač et al., 1999)

Table 2.2: Summary of Interestingness Criteria (Geng & Hamilton, 2006; Lavrač et al., 1999)

Criteria	**Description**
Conciseness	A pattern is concise if it contains relatively few attribute-value pairs. It is relatively easy to understand and remember.
Generality	A pattern is general if it covers a relatively large subset of a dataset.
Reliability	A pattern is reliable if the relationship described by the pattern occurs in a high percentage of applicable cases.
Peculiarity	A pattern is peculiar if it is far away from other discovered patterns according to some distance measurement.
Diversity	A pattern is diverse if its elements differ significantly from each other.
Novelty	A pattern is novel to a person if he or she does not know it and is not able to infer it from other known patterns
Surprisingness	A pattern is surprising if it contradicts a person's existing knowledge or expectation
Utility	A pattern is of utility if it is used by a person to reach a goal.
Actionability	A pattern is actionable on some domain if it enables decision making about future actions in this domain.
Coverage	Measures the fraction of instances covered by the body of a rule.
Accuracy (Error Rate)	Measures the fraction of predicted positives that are true positives in the case of binary classification problems:

(Han & Kamber, 2001; McGarry, 2005) divide interestingness into two classes: *Objective* and *Subjective*. An *objective* measurement is based on the structure of the discovered patterns and the statistics underlying them. Conversely, *subjective* measurement is based on user beliefs in the data. While (Geng & Hamilton, 2006), add another class of interestingness which is *Semantic*. It is a measure of the pattern's explanatory power.

2.6.1 Objective Measures

Objective interestingness measures are based on probability theory, statistics and information theory. With these measures, additional knowledge from the user or the application is not required and the measure is based on the raw data available (Geng & Hamilton, 2006). Various objective interestingness criteria have been used to limit the nature of rules extracted, as explained in (Geng & Hamilton, 2006). A number of

researchers have anticipated an assessment of pattern discovery by applying a statistical significance test as discussed in (Hämäläinen & Nykänen, 2008; Kirsch et al., 2009; Lallich, Teytaud, & Prudhomme, 2007; Webb, 2003, 2007; Weiß, 2008). In what follows, several prominent objective measures are discussed that are based on *support and confidence* framework, statistical measures and information theory, which are related to the research undertaken in this thesis.

2.6.1.1 Support and Confidence

For a classical association framework, a rule is considered interesting if its *support* and *confidence* exceed some user-defined thresholds. Moreover, these two measures can, to a certain extent, demonstrate the usefulness and certainty of discovered rules, respectively (Han et al., 2012). (Agrawal & Srikant, 1994) developed the *Apriori* algorithm in two steps within the support and confidence framework in which the user needs to fix the minimum support and confidence threshold.

Once the frequent itemsets from a transactional database have been found, it is a straightforward matter to generate strong association rules from them. These s*upport* and *confidence* measures were the original interestingness measures proposed for association rules (Agrawal & Srikant, 1994). Strong rules satisfy the minimum support and confidence thresholds. Yet, the strong rules are not essentially interesting either from a statistical or expert's point of view (Lenca et al., 2008). The high confidence should not be confused with high correlation, or with the causality between the antecedent and the consequent of the rule (Brijs et al., 2003; Brin, Motwani, Ullman, & Tsur, 1997; Han & Kamber, 2001).

2.6.1.2 Statistical Measurements

(Han & Kamber, 2001) argue that, "*strong association rules are not necessarily interesting"*. Even with the application of the minimum support and the minimum confidence thresholds, users are swamped with the generation of uninteresting and misleading rules. Alternative measures based on correlation analysis have been utilized by (Brin, Motwani, & Silverstein, 1997; Han & Kamber, 2001) in finding interesting relationships for strong association. This statistical-based test offers a way

to determine the closeness of two probability distributions and is capable of accessing the statistical significance level of dependence between the antecedent and consequent in association rules (Alvarez, 2003).

Statistics have previously addressed the issue of how to separate out the random effects to determine whether the measured association (or difference in other areas) is significant (Agresti, 2007; Hosmer & Lemeshow, 1989). However, the statistical significance assessment is still poorly understood and remains one of the challenging data mining problems to be solved (Kirsch et al., 2009). The three Rule-Interest (RI) functions proposed by (Piatetsky-Shapiro, 1991) effectively measure the correlation between the antecedents and the consequent of a rule, and is one of the prominent works that deal with the statistical independence of rules in data mining. To date, works proposed by (Hämäläinen & Nykänen, 2008; Kirsch et al., 2009; Lallich et al., 2007; Webb, 2003, 2007; Weiß, 2008; Zaki, 2000a) recognize the need for a statistically significant pattern.

(Webb, 2003, 2007) demonstrate the capabilities of their approach by performing two techniques namely, the holdout and direct adjustment, to check for a productive and significant rule. This approach was intended to extend the works done by (Bay & Pazzani, 2001; Meggido & Srikant, 1998) to strictly control the false discovery.

Presenting uninteresting and misleading patterns from a false discovery action is an important issue that can be addressed with statistical-based measures. False discovery is defined as the error of rejecting a null hypothesis, thereby falsely accepting a pattern. The initial work on avoiding false discovery was successfully undertaken by (Meggido & Srikant, 1998). They built a system which can determine the *p-values* for each association rule being generated. The *p-values* are then used to remove any association rules with sufficiently low *p*. However, this method is valid only when applied to sparse data transactions. The problems of controlling false discovery were then solved by Bay and Pazzani (Bay & Pazzani, 2001) with the development of the STUCCO (Search and Testing for Understandable Consistent Contrasts) algorithm which they claim is capable of pruning rules which allow efficient mining at low *support* and guaranteed control over false discovery. This involves finding a

significant contrast set using the chi-squared test and controlling the search error with the Bonferroni corrections.

Other variants of the statistical approach, such as the bootstrap-based method (Lallich, et al., 2007), are capable of controlling the multiple risks and avoiding the risk of false discovery. Hämäläinen and Nykänen (Hämäläinen & Nykänen, 2008) have successfully implemented the *StatApriori* algorithm which searches statistically significant and non-redundant rules. This algorithm was developed to control the existence of false negatives and false positives discovered in association rule mining. The method of selecting a meaningful *support* threshold by Kirsch et al. (Kirsch et al., 2009) is another novel approach to control the false discovery rate in pattern discovery. They define the significant itemsets with small false discovery rate as those itemsets that deviate substantially from the expected random dataset. Additionally, other works based on the minimum *support* threshold, such as the method proposed by Wei et al. (Wei, Yi, & Wang, 2006), are also able to mine valuable rules while pruning the low relation rules.

In the datasets where there is a predefined class label (i.e. classification tasks), frequent pattern mining can contribute to discovering strong associations between occurring attribute and class values (Li, Shen, & Topor, 2002). A combination of a frequent-pattern-based framework with a feature selection algorithm, namely the Maximal Marginal Relevance Feature Selection (MMRFS) proposed by (Cheng, Yan, Han, & Hsu, 2007), is a discriminative frequent-pattern-based classification that is capable of overcoming the overfitting in a classification problem and has proven to be scalable and highly accurate. The aforementioned framework involves a two-step process, firstly to mine the frequent pattern, and secondly, to perform feature selection or rule ranking. An improvement of this work that is capable of directly mining the discriminative pattern is proposed by (Cheng, Yan, Han, & Yu, 2008). This approach, namely the Direct Discriminative Pattern Mining (DDPMine), is capable of discovering classification rules by incorporating the feature selection method into the mining framework by directly mining the most discriminative patterns, and then incrementally eliminating the training instances which are covered by those patterns. With respect to the application of *closed* itemsets for prediction and classification problems, some work has already been initiated in (Garriga, Kralj, & Lavrač, 2008).

The covering properties of *closed* sets have successfully increased the accuracy of rule-based classifiers, reduced the number of rules from *emerging patterns* and decreased the number of rules to those essential for the classification task. However, in terms of usefulness, no comparison has been made between *closed* sets and *maximal* sets and frequent itemset sets for classification tasks, with respect to their classification accuracy, generalization capability and coverage rate, and the application of interestingness measures to such patterns.

(Novak, Lavrač, & Webb, 2009) proposed unifying supervised descriptive rule discovery framework which combines three data mining areas namely the contrast set mining (CSM), emerging pattern mining (EPM) and subgroup discovery (SD). The unifying framework provides a proper comparison and commonalities in terms of the terminology, definitions, goals, algorithms, heuristics and rules visualization of CSM, EPM and SD. It shows that the heuristics used in each of the areas aims at optimizing the common trade-off between rule coverage and accuracy. Although the descriptive rule discovery offers a systematic comparison and commonalities between each area, an appropriate sequence of applying the different heuristics to arrive at an optimal rule set has not been discussed.

2.6.1.3 Information Theory

Another type of objective measures for interesting patterns is based on the information theory procedure. Information theory refers to the interpretation of information from patterns. A rule is considered interesting when the antecedent provides a great deal of information about the consequent (Blanchard, Guillet, Gras, & Briand, 2005).

Listed here are several measures that evaluate the rules based on the information theory approach. The mutual information as characterized by (Geng & Hamilton, 2006; Jaroszewicz & Simovici, 2001; Ke et al., 2008) is a measure that describes how much information one random variable imparts about another one. (Blanchard et al., 2005; Geng & Hamilton, 2006) described the Shannon conditional entropy as an information theory that calculates the average amount of information of the consequent given that the antecedent is true. The J-measure as proposed by Smyth &

Goodman (Smyth & Goodman, 1992) is an information measure capable of quantifying the information content of a rule or a hypothesis. The Gini index as reported in (Blanchard et al., 2005; Jaroszewicz & Simovici, 2001; Bayardo & Rakesh, 1999) is a measure based on distribution divergence. Moreover, (Blanchard, et al., 2005) presented the Directed Information Ratio (DIR), a new rule interestingness measure which is based on information theory. DIR is specially designed for association rules, and in particular it differentiates two opposite rules $a \to b$ and $a \to \bar{b}$. This measure, as asserted by (Blanchard et al., 2005), is capable of rejecting both independence and equilibrium; that is, it discards both the rules whose antecedent and consequent are negatively correlated, and those rules which have more counter-examples than supporting examples.

2.6.2 Subjective Measures

The data mining model may generate rules that satisfy certain requirements, but the question remains whether the rules are correct. (Roiger & Geatz, 2003) argue that this question is not easily answered, since the rules generated from any data mining model are being developed by a data mining specialist; thus, additional information is needed as this will verify the usefulness of the discovered knowledge and the applicability of the rules.

The additional information may vary based on the knowledge background of the user, the user interest and the evolution of the user's knowledge. Rules' measuring which involves both user knowledge and data is identified as the subjective measure (Geng & Hamilton, 2006). Since the representation of user knowledge may be in various forms, it is hard to formulate the subjective measures into simple mathematical formulas such as those utilized in the objective measures. Additionally, unexpectedness and novelty are the two major criteria used in subjective measures for determining the interestingness of rules (Geng & Hamilton, 2006).

Unexpectedness or the novelty of rules can be determined based on three distinguished roles. In the first roles, a system will pick unexpected rules to be presented to the user based on specific format defined by the user (Liu, Hsu, Mun, &

Lee, 1999; Silberschatz & Tuzhilin, 1996). Interactive feedback between the user and the system is capable of identifying the unexpectedness of the subjective rules. With this approach, the system will remove uninteresting rules based on interaction with the user. The final task in determining the unexpectedness is by reducing the data mining search spaces which will provide fewer results. This can be done by the forming user's specifications as constraints (Padmanabhan & Tuzhilin, 1998).

An example of a user specification constraint is the belief system (Silberschatz & Tuzhilin, 1996). This refers to a subjective measure that can quantify the user belief based on either *hard* or *soft* belief. The *hard* belief refers to some constraint that is fixed and cannot be changed, while *soft* belief refers to the willingness of the user to make some adjustment if new rules are discovered.

Interaction between the user and the system has been proposed by (Sahar, 1999). With this approach, there are no predefined interestingness measures. Patterns' interestingness is measured in three steps: selection of best candidate by the system, candidate presentation to the user (involving selection process by the user) and finally, retention by the system of the true interesting set of rules.

The reduction of the mining spaces based on user specification is the final task stated by (Geng & Hamilton, 2006) in determining the novelty of a rule. Work developed by (Padmanabhan & Tuzhilin, 1998) is based on the belief system approach. However, rather than mining rules that are in agreement with user beliefs, only contradictive rules are mined. This approach is preferred for the discovering of any rules that conflict with user knowledge.

Subjective measures are considered to be more useful and valuable for the more experienced and with active participation from the users (Geng & Hamilton, 2006). However, in most cases, the availability of subjective knowledge/domain knowledge cannot always be assumed. Moreover, these measures also exploit information gathered from different users' background knowledge, which consequently may lead to different measures and ways to determine rules interestingness (Geng & Hamilton, 2006). Hence, the focus of the work in this thesis is on ascertaining rules based on interestingness measures that do not depend on the availability of domain knowledge.

2.6.3 Semantics Measures

Another rules measurement technique is the semantic measure. This is a complementary interestingness measure for evaluating the association rules (Maddouri & Gammoudi, 2007). A semantic measure considers the semantics and explanations of the patterns. Rules semantics are measured based on their utility and actionability (Geng & Hamilton, 2006).

The utility criteria in this semantic measure are a combination of the statistical aspect of the data and the utility of the mined rules. A *weighted association rule mining* is one example of a utility-based measure as proposed by (Geng & Hamilton, 2006). With this measure, each item is allocated a certain weight representing its level of importance. This utility-based measure is an extension of the support measure from a standard *Apriori* algorithm. (Geng & Hamilton, 2006), summarized 11 utility-based measures; however, both Geng and Hamilton agree that each application needs a different utility measures. Actionability criteria in semantic measurement refer to how the user can take advantage of the patterns. Moreover, patterns that are found to be actionable will assist the user to take real actions, thereby providing additional information for the decision-making process (Cao, 2010). Additionally, (Maddouri & Gammoudi, 2007) proposed 12 semantic properties by carefully examining the behaviour of sixty interestingness measures. Based on these 12 semantic properties, (Maddouri & Gammoudi, 2007) claim that the Zhang measurement technique is an important measure that satisfies the majority of the 12 proposed semantic properties.

However, since the semantic measure requires the user to have background knowledge, this will lead to similar difficulties as those that apply to the subjective measure. (Geng & Hamilton, 2006) agree that the presentation of rules that can reflect human interests will remain an active research issue.

2.6.4 Summary of the Main Deficiency in Interestingness Evaluation of Association Rules for Relational Data

As mentioned in our discussion in Section 2.5 and Section 2.6, each of these interestingness measures have their own strengths and weaknesses (Hilderman & Hamilton, 2001). While (Webb, 2007) has examined the latest developments in the

significance of rules discovery. Some areas worth further exploration involve: issues concerning the optimal split between the subset of data used for learning and evaluation, selection of a suitable statistical test and assessment of the rules with more than one itemset in the consequent.

Although there are various criteria for determining the usefulness of rules as overviewed in the aforementioned section, the measures usually reflect the usefulness of rules only with respect to the specific database being observed (Webb, 2007). It is hard to determine whether the rules produced are useful in practice or are valid for real-world problems. Applying a data mining algorithm to practical problems may not be sufficient because one needs to ensure that the results have a sound statistical basis. To conclude, the evaluation of the interestingness of rules is essential in many applications. While a substantial number of interestingness and constraint-based measures have been proposed and successfully applied, there is still a need to understand the roles that these parameters play and the way in which they should be utilized. An understanding of the various implications of applying each parameter and providing a systematic, sequential procedure will ensure that one will arrive at a more reliable and interesting set of rules. Furthermore, with the frequent itemsets, closed and maximal mining being an important approach to arrive at the initial set of rules, there is still a need to understand the difference and advantages/disadvantages of using each of these as a basis for classification tasks with respect to accuracy, rule coverage and generalization power.

2.7 Interestingness and Validity of Rules for Semi-Structured Data

To date, limited work has been done on the rule evaluation phase of semi-structured rules. Many of the well developed rule interestingness measures are in relational data and they have had great success in evaluating rule interestingness as discussed in (Tan et al., 2002). Several works on the evaluation of discovered patterns based on statistical significance are those of (Aumann & Lindell, 2003; Meggido & Srikant, 1998; Webb, 2003, 2007), but these are limited to relational data. The existence of vast well-developed measuring techniques to evaluate interestingness of rules from relational data, offers great opportunities for adapting these techniques for verifying significant subtrees from semi-structured data. The applicability of these

interestingness measures needs to be explored in the context of frequent subtree mining, where necessary adjustments and extensions need to be made to ascertain the validity of the methods given the more complex structured aspects in the data, which often need to be preserved in the rules.

One line of work focusing on more interesting subtree patterns aims to reduce the patterns and the application of plausible constraints techniques. The problem of mining mutually dependent ordered subtrees has been addressed in (Ozaki & Ohkawa, 2009). The proposed algorithm utilizes the hyperclique method (Xiong, Tan, & Kumar, 2006) in the tree mining context so that all the components of a subtree are highly correlated together. These hyperclique subtree patterns are discovered using an *h-confidence* measure which is the minimum probability of an item from a pattern in one transaction implying the presence of all other items in the same transaction. Hence, the extracted hyperclique subtree patterns will satisfy the minimum *h-confidence* threshold. The work done in (Bathoorn, Koopman, & Siebes, 2006) uses the method proposed for database compression in regards to item set mining in (Siebes, Vreeken, & Leeuwen, 2006) to demonstrate how the same minimum description length principle can yield good results for sequential and tree-structured data. Another notable work presented in (Nakamura & Kudo, 2005) extends the idea of the item constraint (Srikant, Vu, & Agrawal, 1997) to that of a node-inclusion constraint in subtrees. Furthermore, (Knijf & Feelders, 2005) proposed the use of monotone constraints in frequent subtree mining, namely monotone, anti-monotone, convertible and succinct constraints. Using these constraints, the frequent subtrees are mined using an opportunistic pruning strategy, and the set of frequent subtrees are reduced to only those satisfying the specific user pre-defined constraints. An approach to mining of frequent subtrees where the distance between the nodes is used as additional grouping criterion has been presented in (Hadzic, Tan, & Dillon, 2008). However, the usefulness of this distance constraint for generating interesting rules has not been explored for the classification task.

Besides the aforementioned constraint-based techniques, to the best of our knowledge, there are limited works on verifying the significance of discovered frequent subtrees. The frequently-occurring subtrees discovered with the frequent subtree mining are often too numerous to be utilized efficiently and effectively for the application at hand

(Hadzic, 2011; Hadzic et al., 2011; Ikasari, Hadzic, & Dillon, 2011). (Hashimoto, Takigawa, Shiga, Kanehisa, & Mamitsuka, 2008) proposed and developed an application of statistical hypothesis testing to re-rank the significant frequent subtrees. This approach ranks the significant patterns according to *P-values* obtained from the *Fisher's Exact* test of significance. The significant patterns were then used for Glycan classifications problems. Recently (Yan, Cheng, Han, & Yu, 2008), proposed a mining framework called LEAP (Descending Leap Mine) for checking and mining significant frequent subgraphs which helps to discard redundant frequent subgraphs. For a predefined class label in XML documents, an efficient *XRules* classifier has been developed by (Zaki & Aggarwal, 2003). This approach offers promising results in terms of a structural classifier for semi-structured data, but nevertheless utilizes standard measures of interestingness based on support and confidence.

2.7.1 Summary of the Main Deficiency Interestingness Evaluation of Association Rules for Semi-Structured Data

With the limited work that has been done on evaluating the subtrees' interestingness for semi-structured data, a number of the criticisms of pattern interestingness measures in relational data also apply to semi-structured data. Such criticisms include the fact that, while interesting subtrees may be found from an XML database, by meeting certain constraints criteria and satisfying the measure of interestingness, there is still a need to understand the roles that these constraints and parameters play and the way in which they should be utilized. Furthermore, many misleading, uninteresting and insignificant frequent rules may still be produced, as is the case for any frequent pattern-based approach (Han & Kamber, 2001). The problem arises because some subtrees are discovered due to pure coincidence resulting from certain randomness in the particular dataset being analyzed. Moreover, in semi-structured data additional challenges exist due to the structural aspects inherent in data, which are mostly not accounted for by existing interestingness measures. Thus, as mentioned earlier at the end of Section 2.6.4, in order to guarantee the discovery of reliable and interesting frequent subtrees, what is needed is an understanding of the various implications of applying each parameter and constraint, and a systematic sequence of usage. Moreover, an understanding of the differences and advantages/disadvantages of basing the association rules on different subtree types (i.e. induced, embedded,

disconnected) is needed with respect to resulting rules' accuracy, coverage and generalization power.

2.8 Relationship between Feature Subset Selection and Rule Interestingness

The feature subset selection as describes in (Han & Kamber, 2001) is a ways to minimize the number of features within the dataset by removing irrelevant or redundant features/attributes. In general, the objective of feature subset selection as defined in (Han & Kamber, 2001) is "*to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes*". Han and Kamber in (Han & Kamber, 2001) asserted that, domain expertise can be employed in order to pick up useful attributes. However, because the data mining task involves a large volume of data and unpredictable behaviour of data during data mining, this task is often expensive and time consuming.

The test of statistical significance is one of the prominent approaches in evaluating attributes/features usefulness. Stepwise forward selection, stepwise backward selection and a combination of both are three commonly used heuristic techniques utilized in statistical significance tests such as linear regression and logistic regression (Han & Kamber, 2001). Moreover, the application of correlation analysis such as the chi-squared test is also valuable in identifying redundant variables for features subset selection. Another powerful technique for this purpose is the Symmetrical Tau (Zhou & Dillon, 1991), which is a statistical-heuristic feature selection criterion. It measures the capability of an attribute in predicting the class of another attribute. Additionally, information gain is another attributes' relevance analysis method employed in the popular ID3 (Quinlan, 1986) and C4.5 (Quinlan, 1993) as reported in (Han & Kamber, 2001), for selecting the most prominent class distinguishing attributes as split nodes in the decision tree.

While the original purpose of features subset selection is to reduce the number of attributes to only those attributes relevant for a certain data mining task, they nevertheless can be utilized to measure the interestingness of rules/pattern generated. For example, if the rules/pattern themselves consist of irrelevant attributes, the

aforementioned measure can also give some indication that the rules/pattern is not interesting. Moreover, (Geng & Hamilton, 2006) stated that there are three roles of interestingness measures. The first is their ability to discard uninteresting patterns during the mining process, thereby narrowing the search space and improving the mining efficiency. The second role is to calculate the interestingness scores for each pattern, which allows the ranking of patterns according to specific needs. The final role is the use of interestingness measures during the post-processing stage to select interesting patterns.

In this section, the focus is on the first role of the rules interestingness measures in selecting interesting frequent patterns. Interestingness measures such as the chi-squared test (Brin, Motwani, & Silverstein, 1997), Symmetrical Tau (Zhou & Dillon, 1991) and Mutual Information (Tan et al., 2002), are capable of measuring the interestingness of rules and at the same time identifying useful features for frequent patterns.

Since frequent patterns are generated based solely on frequency without considering their predictive power, the use of frequent patterns without selecting appropriate features will still result in a huge feature space which leads to larger volume and complexity of rules. This might not only slow down the model learning process, but even worse, the classification accuracy deteriorates (another kind of overfitting issue since the features are numerous) (Cheng et al., 2007)

2.9 Conclusion

This chapter has provided the literature review of the current state of research in the problem areas related to the work of this thesis. As mentioned in Chapter 1, the main problem area on which this thesis focuses is the evaluation of data mining rules, paying particular attention to association rules generated from both relational and semi-structured data (focusing on XML and tree-structured data in general). A general overview of the basic foundations of topics related to association rule mining and frequent pattern mining was first provided. Different ways of mining frequent patterns from relational data and semi-structured data were discussed. The strengths and weaknesses of the different approaches were indicated.

The remainder of the chapter examined the state of the research into specific problems of interestingness measures and validity of rules addressed in this thesis. A survey on general issues of interestingness and validity of rules is provided, followed by a detailed discussion of the classification of interestingness measures based on the objective, subjective and semantic measures. In addition, extensive and more detailed discussions are focused on statistic-based measures to evaluate the rules' interestingness and validity. A discussion is provided with respect to the rules' interestingness and validity from semi-structured data. In the last part of the chapter, a feature subset selection problem is discussed in terms of arriving at more interesting sets of rules.

References

Aggarwal, C. C., & Yu, P. S. (1998). A new framework for itemset generation. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. Seattle, Washington, United States: ACM.

Agrawal, R., Imieliski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec., 22*, 207-216.

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc.

Agresti, A. (2007). *An introduction to categorical data analysis* (2nd ed.). Hoboken, N.J.: Wiley-Interscience.

Alvarez, S. A. (2003). Chi-Squared Computation for Association Rules: Preliminary Results. In *Technical Report BC-CS-2003-0*: Computer Science Department, Boston College.

Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H., & Arikawa, S. (2002). Efficient Substructure Discovery from Large Semi-structured Data. In R. L. Grossman, J. Han, V. Kumar, H. Mannila & R. Motwani (Eds.), *Second SIAM International Conference on Data Mining*. Arlington, VA, USA: SIAM.

Aumann, Y., & Lindell, Y. (2003). A Statistical Theory for Quantitative Association Rules. *J. Intell. Inf. Syst., 20*, 255-283.

Bathoorn, R., Koopman, A., & Siebes, A. (2006). Reducing the Frequent Pattern Set. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*: IEEE Computer Society.

Bay, S. D., & Pazzani, M. J. (2001). Detecting Group Differences: Mining Contrast Sets. *Data Mining and Knowledge Discovery, 5*, 213-246.

Bayardo, R. J. (1998). Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*. Seattle, Washington, United States: ACM.

Bayardo, R. J., Agrawal, R., & Gunopulos, D. (2000). Constraint-Based Rule Mining in Large, Dense Databases. *Data Mining and Knowledge Discovery, 4*, 217-240.

Blanchard, J., Guillet, F., Gras, R., & Briand, H. (2005). Using Information-Theoretic Measures to Assess Association Rule Interestingness. In *Proceedings of the 5th IEEE International Conference on Data Mining*: IEEE Computer Society.

Braga, D., Campi, A., Ceri, S., Klemettinen, M., & Lanzi, P. (2003). Discovering interesting information in XML data with association rules. In *Proceedings of the 2003 ACM Symposium on Applied Computing*. Melbourne, Florida: ACM.

Brijs, T., Vanhoof, K., & Wets, G. (2003). Defining interestingness for association rules. *International journal of information theories and applications, 10(4)*, 370-376.

Brin, S., Motwani, R., & Silverstein, C. (1997). Beyond market baskets: generalizing association rules to correlations. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*. Tucson, Arizona, United States: ACM.

Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*. Tucson, Arizona, United States: ACM.

Burdick, D., Calimlim, M., Flannick, J., Gehrke, J., & Yiu, T. (2005). MAFIA: a maximal frequent itemset algorithm. *IEEE Transactions on Knowledge and Data Engineering, 17*, 1490-1504.

Cao, L. (2010). Domain-Driven Data Mining: Challenges and Prospects. *Knowledge and Data Engineering, IEEE Transactions on, 22*, 755-769.

Chen, L., Bhowmick, S., & Chia, L.-T. (2004). Mining Association Rules from Structural Deltas of Historical XML Documents. In H. Dai, R. Srikant & C. Zhang (Eds.), *Advances in Knowledge Discovery and Data Mining* (Vol. 3056, pp. 452-457): Springer Berlin / Heidelberg.

Cheng, H., Yan, X., Han, J., & Hsu, C.-W. (2007). Discriminative frequent pattern analysis for effective classification. In *23rd International Conference on Data Enginering (ICDE 2007)* (pp. 716-725). Pisctaway, NJ, USA: IEEE.

Cheng, H., Yan, X., Han, J., & Yu, P. S. (2008). Direct discriminative pattern mining for effective classification. In *24th International Conference on Data Engineering (ICDE 2008)* (pp. 169-178). Piscataway, NJ, USA: IEEE.

Chi, Y., Muntz, R. R., Nijssen, S., & Kok, J. N. (2005). Frequent Subtree Mining - An Overview. *Fundamenta Informaticae, 66*, 161-198.

Feng, L., Dillon, T., Weigand, H., & Chang, E. (2003). An XML-Enabled Association Rule Framework. In *Database and Expert Systems Applications* (pp. 88-97).

Fukuda, T., Morimoto, Y., Morishita, S., & Tokuyama, T. (1996). Mining optimized association rules for numeric attributes. In *Proceedings of the 15th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. Montreal, Quebec, Canada: ACM.

Garriga, G. C., Kralj, P., & Lavrač, N. (2008). Closed Sets for Labeled Data. *Journal of Machine Learning Research, 9*, 559-580.

Geng, L., & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Comput. Surv., 38*, 9.

Goodman, A., Kamath, C., & Kumar, V. (2008). Data Analysis in the 21st Century. *Stat. Anal. Data Min., 1*, 1-3.

Gouda, K., & Zaki, M. J. (2001). Efficiently Mining Maximal Frequent Itemsets. In *First IEEE International Conference on Data Mining (ICDM'01)* (Vol. 0, pp. 163).

Hadzic, F. (2011). A Structure Preserving Flat Data Format Representation for Tree-Structured Data. In *2011 Workshop on Quality Issues, Measures of Interestingness and Evaluation of Data Mining Models (QIMIE'11)*. Shenzhen, China: Frontiers of Computer Science in China Journal, Springer.

Hadzic, F., Tan, H., & Dillon, T. (2008). Mining Unordered Distance-Constrained Embedded Subtrees. In *Proceedings of the 11th International Conference on Discovery Science*. Budapest, Hungary: Springer-Verlag.

Hadzic, F., Tan, H., & Dillon, T. S. (2011). *Mining of Data with Complex Structures* (Vol. 333): Springer-Verlag Berlin Heidelberg.

Hämäläinen, W., & Nykänen, M. (2008). Efficient Discovery of Statistically Significant Association Rules. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*: IEEE Computer Society.
Han, J., Cheng, H., Xin, D., & Yan, X. (2007). Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery, 15*, 55-86.
Han, J., & Fu, Y. (1995). Discovery of Multiple-Level Association Rules from Large Databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc.
Han, J., & Kamber, M. (2001). *Data mining: concepts and techniques*. San Francisco: Morgan Kaufmann Publishers.
Han, J., Kamber, M., & Pei, J. (2012). *Data mining: concepts and techniques* (3rd ed.). Waltham, Mass.: Elsevier/Morgan Kaufmann.
Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. Dallas, Texas, United States: ACM.
Hashimoto, K., Takigawa, I., Shiga, M., Kanehisa, M., & Mamitsuka, H. (2008). Mining significant tree patterns in carbohydrate sugar chains. *Bioinformatics, 24*, i167-173.
Hido, S., & Kawano, H. (2005). AMIOT: induced ordered tree mining in tree-structured databases. In *Fifth IEEE International Conference on Data Mining* (pp. 8 pp.).
Hilderman, R., & Hamilton, H. (1999). *Knowledge discovery and interestingness measures: A survey*. Boston,MA: Kluwer Academic.
Hilderman, R., & Hamilton, H. (2001). Evaluation of Interestingness Measures for Ranking Discovered Knowledge. In *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*: Springer-Verlag.
Hosmer, D. W., & Lemeshow, S. (1989). *Applied logistic regression*. New York: Wiley.
Ikasari, N., Hadzic, F., & Dillon, T. S. (2011). Incorporating Qualitative Information for Credit Risk Assessment through Frequent Subtree Mining for XML. In A. Tagarelli (Ed.), *XML Data Mining: Models, Method, and Applications*: IGI Global.
Jaroszewicz, S., & Simovici, D. (2001). A General Measure of Rule Interestingness. In L. De Raedt & A. Siebes (Eds.), *Principles of Data Mining and Knowledge Discovery* (Vol. 2168, pp. 253-265): Springer Berlin / Heidelberg.
Ke, Y., Cheng, J., & Ng, W. (2008). An information-theoretic approach to quantitative association rule mining. *Knowl. Inf. Syst., 16*, 213-244.
Kirsch, A., Mitzenmacher, M., Pietracaprina, A., Pucci, G., Upfal, E., & Vandin, F. (2009). An efficient rigorous approach for identifying statistically significant frequent itemsets. In *Proceedings of the 28th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. Providence, Rhode Island, USA: ACM.
Knijf, J. D., & Feelders, A. J. (2005). Monotone Constraints in Frequent Tree Mining. In M. Poel & A. Nijholt (Eds.), *BENELEARN:Proceedings of the 14th Annual Machine Learning Conference of Belgium and the Netherlands* (pp. 13-20). Enschede, The Netherlands.
Lallich, S., Teytaud, O., & Prudhomme, E. (2007). Association Rule Interestingness: Measure and Statistical Validation. In *Quality Measures in Data Mining* (pp. 251-275).
Lavrač, N., Flach, P., & Zupan, B. (1999). Rule Evaluation Measures: A Unifying View. In *Inductive Logic Programming* (pp. 174-185).
Lenca, P., Meyer, P., Vaillant, B., & Lallich, S. (2008). On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. *European Journal of Operational Research, 184*, 610-626.
Li, J., Shen, H., & Topor, R. W. (2002). Mining the optimal class association rule set. *Knowledge-Based Systems, 15*, 399-405.
Liu, B., Hsu, W., Mun, L.-F., & Lee, H.-Y. (1999). Finding interesting patterns using user expectations. *Knowledge and Data Engineering, IEEE Transactions on, 11*, 817-832.
Maddouri, M., & Gammoudi, J. (2007). On Semantic Properties of Interestingness Measures for Extracting Rules from Data. In B. Beliczynski, A. Dzielinski, M. Iwanowski & B.

Ribeiro (Eds.), *Adaptive and Natural Computing Algorithms* (Vol. 4431, pp. 148-158): Springer Berlin / Heidelberg.

Mannila, H., Toivonen, H., & Verkamo, A. I. (1994). Efficient Algorithms for Discovering Association Rules. In U. M. Fayyad & R. Uthurusamy (Eds.), *AAAI Workshop* (pp. 181-192). Seattle, Washington: AAAI Press.

McGarry, K. (2005). A survey of interestingness measures for knowledge discovery. *Knowl. Eng. Rev., 20*, 39-61.

Meggido, N., & Srikant, R. (1998). Discovering Predictive Association Rules. In *4th International Conference on Knowledge Discovery in Databases and Data Mining* (pp. 274-278).

Moreno, M. N., Segrera, S., Lopez, V. F., & Polo, M. J. (2006). A method for mining quantitative association rules. In *Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization*. Lisbon, Portugal: World Scientific and Engineering Academy and Society (WSEAS).

Nakamura, A., & Kudo, M. (2005). Mining Frequent Trees with Node-Inclusion Constraints. In *Advances in Knowledge Discovery and Data Mining* (pp. 850-860).

Novak, P. K., Lavrač, N., & Webb, G. I. (2009). Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Patterns and Subgroup Mining. *J. Mach. Learn. Res., 10*, 377-403.

Omiecinski, E. R. (2003). Alternative interest measures for mining associations in databases. *Knowledge and Data Engineering, IEEE Transactions on, 15*, 57-69.

Ozaki, T., & Ohkawa, T. (2009). Mining Mutually Dependent Ordered Subtrees in Tree Databases. In *New Frontiers in Applied Data Mining: PAKDD 2008 International Workshops, Osaka, Japan, May 20-23, 2008. Revised Selected Papers* (pp. 75-86): Springer-Verlag.

Padmanabhan, B., & Tuzhilin, A. (1998). A Belief-Driven Method for Discovering Unexpected Patterns In *4th International Conference on Knowledge Discovery and Data Mining (KDD-98)* (pp. 94-100).

Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. (1999). Discovering Frequent Closed Itemsets for Association Rules. In C. Beeri & P. Buneman (Eds.), *Database Theory — ICDT'99* (Vol. 1540, pp. 398-416): Springer Berlin / Heidelberg.

Pei, J., Han, J., & Mao, R. (2000). CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets. In *ACM {SIGMOD} Workshop on Research Issues in Data Mining and Knowledge Discovery* (pp. 21-30).

Piatetsky-Shapiro, G. (1991). Discovery, analysis; and presentation of strong rules. *Knowledge discovery in database*, 229.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81-106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*: Morgan Kaufman.

Roberto J. Bayardo, Jr., & Rakesh, A. (1999). Mining the most interesting rules. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Diego, California, United States: ACM.

Roiger, R., & Geatz, M. (2003). *Data mining: a tutorial-based primer*. Boston: Addison Wesley.

Sahar, S. (1999). Interestingness via what is not interesting. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Diego, California, United States: ACM.

Siebes, A., Vreeken, J., & Leeuwen, M. V. (2006). Item Sets That Compress. In *Proceedings of the SIAM Conference on Data Mining 2006 (SDM'06)* (pp. 393-404). Maryland, USA.

Silberschatz, A., & Tuzhilin, A. (1996). What makes patterns interesting in knowledge discovery systems. *Knowledge and Data Engineering, IEEE Transactions on, 8*, 970-974.

Silverstein, C., Brin, S., & Motwani, R. (1998). Beyond Market Baskets: Generalizing Association Rules to Dependence Rules. *Data Min. Knowl. Discov., 2*, 39-68.

Smyth, P., & Goodman, R. M. (1992). An information theoretic approach to rule induction from databases. *Knowledge and Data Engineering, IEEE Transactions on, 4*, 301-316.

Srikant, R., & Agrawal, R. (1995). Mining Generalized Association Rules. In *Proceedings of the 21th International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc.

Srikant, R., & Agrawal, R. (1996). Mining quantitative association rules in large relational tables. *SIGMOD Rec., 25*, 1-12.

Srikant, R., Vu, Q., & Agrawal, R. (1997). Mining Association Rules with Item Constraints. In *3rd Int'l Conference on Knowledge Discovery in Databases and Data Mining* (pp. 67-73). Newport Beach, California: AAAI Press.

Suciu, D. (1998). Semistructured Data and XML. In K. Tanaka & S. Ghandeharizadeh (Eds.), *The 5th International Conference on Foundations of Data Organization (FODO'98)* (pp. 1-12). Kobe, Japan.

Tan, H., Dillon, T., Hadzic, F., Chang, E., & Feng, L. (2006). IMB3-Miner: Mining Induced/Embedded Subtrees by Constraining the Level of Embedding. In *Advances in Knowledge Discovery and Data Mining* (pp. 450-461).

Tan, H., Hadzic, F., Dillon, T. S., & Chang, E. (2008). State of the art of data mining of tree structured information. *International Journal of Computer Systems Science and Engineering, 23*.

Tan, P.-N., Kumar, V., & Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Edmonton, Alberta, Canada: ACM.

Termier, A., Rousset, M. C., & Sebag, M. (2002). TreeFinder: a first step towards XML data mining. In *Proceedings of the (ICDM) International Conference on Data Mining* (pp. 450-457).

Toivonen, H. (1996). Sampling Large Databases for Association Rules. In *Proceedings of the 22th International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc.

Webb, G. I. (2003). Preliminary investigations into statistically valid exploratory rule discovery. In *Australasian data mining workshop(AudDM03)* (pp. 1-9). University of Technology,Sydney.

Webb, G. I. (2007). Discovering Significant Patterns. *Machine Learning, Springer*, 1-33.

Wei, J.-M., Yi, W.-G., & Wang, M.-Y. (2006). Novel measurement for mining effective association rules. *Knowledge-Based Systems, 19*, 739-743.

Weiß, C. (2008). Statistical mining of interesting association rules. *Statistics and Computing, 18*, 185-194.

Xiong, H., Tan, P.-N., & Kumar, V. (2006). Hyperclique pattern discovery. *Data Min. Knowl. Discov., 13*, 219-242.

Yan, X., Cheng, H., Han, J., & Yu, P. S. (2008). Mining significant graph patterns by leap search. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. Vancouver, Canada: ACM.

Yun, C., Yirong, Y., & Richard, R. M. (2004). HybridTreeMiner: An Efficient Algorithm for Mining Frequent Rooted Trees and Free Trees Using Canonical Forms. In *Proceedings of the 16th International Conference on Scientific and Statistical Database Management*: IEEE Computer Society.

Yun, C., Yirong, Y., & Richard, R. M. (2005). Canonical forms for labelled trees and their applications in frequent subtree mining. *Knowl. Inf. Syst., 8*, 203-234.

Yun, H., Ha, D., Hwang, B., & Ryu, K. H. (2003). Mining association rules on significant rare data using relative support. *Journal of Systems and Software, 67*, 181-191.

Zaki, M. J. (2000a). Generating non-redundant association rules. In *Proceedings of The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Boston, Massachusetts, United States: ACM.

Zaki, M. J. (2000b). Scalable algorithms for association mining *IEEE Transactions on Knowledge and Data Engineering, 12*, 372-390.
Zaki, M. J. (2004). Efficiently Mining Frequent Embedded Unordered Trees. *Fundam. Inf., 66*, 33-52.
Zaki, M. J. (2005). Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications. *IEEE Transactions on Knowledge and Data Engineering, 17*, 1021-1035.
Zaki, M. J., & Aggarwal, C. C. (2003). XRules: an effective structural classifier for XML data. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Washington, D.C.: ACM.
Zaki, M. J., & Hsiao, C.-J. (2002). CHARM: An Efficient Algorithm for Closed Itemset Mining. In *2nd SIAM International Conference in Data Mining*.
Zhang, H., Balaji, P., & Alexander, T. (2004). On the discovery of significant statistical quantitative rules. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Seattle, WA, USA: ACM.
Zhang, J., Ling, T. W., Bruckner, R. M., Tjoa, A. M., & Liu, H. (2004). On Efficient and Effective Association Rule Mining from XML Data. In *Database and Expert Systems Applications* (pp. 497-507).
Zhou, X. J., & Dillon, T. S. (1991). A statistical-heuristic feature selection criterion for decision tree induction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*, 834-841.

CHAPTER 3: GENERAL CONCEPTS, DEFINITIONS AND PROBLEM TO BE ADDRESSED

3.1 Introduction

This thesis presents a study of the interestingness and validity of rules generated by the association rules mining technique. In this chapter, the essential concepts, problem definitions, and the problems to be addressed are discussed as the basis for the development of subsequent chapters.

Since the problems of rules interestingness and validity will be approached from the perspective of relational data and semi-structured data, the general concepts and definitions of 'relational' and 'semi-structured' will be given separately in Section 3.2 and Section 3.3. Section 3.3 will also discuss the problem of modelling XML documents, and the parallelism between XML and tree structure. This clarifies the focus within the problem from the semi-structured data perspective, being that of XML documents and tree-structured data in general. Nevertheless, the study is applicable to any documents/data that can be effectively modelled as a rooted ordered labelled tree, as is the case in XML. Section 3.4 and 3.5 provide the concepts and the definitions of association rule mining for relational data and tree-structured data, respectively. An overview of the problems to be addressed in this thesis is given in Section 3.6. Section 3.7 is reserved for the discussion of the methodologies chosen in this thesis for approaching the problems described in Section 3.6. Finally, this chapter is concluded with a summary in Section 3.8.

3.2 General Concepts and Definitions of Relational Data

In the thesis, the problems are defined based on two separate data types. This sub-topic is devoted to defining several important terms relating to the general terms and concepts regarding a relational data model. The relational data model was inspired by the concept of mathematical relations. Relational data has a structured format where the scheme of the data is fixed. (Han & Kamber, 2001) defined the relational data model as compromising a set of tables, each of which is assigned a unique name. Each table consists of a set of attributes (columns or fields) and usually stores a large set of tuples (records or rows). Each tuple in a relational table represents an object

identifier by means of a unique key and is described by a set of attribute values. Additionally, each record is an ordered list of values, with one value for each field (Frawley, Piatetsky-Shapiro, & Matheus, 1992).

3.2.1 Relational Data

The following definition of a relational data model is taken from (Vossen, 1991). For a relational data model: Let X be a set of attributes, $X = \{A_1, ..., A_m\}$, and let each attributes $A \in X$ have a non-empty domain $dom(A)$ with at least two elements; thus, every domain permits at least assertions of the form 'true/false' or 'yes/no'. A **tuple** over X is an injective mapping $\mu : X \to dom(X)$, for which the following holds: $(\forall A \in X)\ \mu(A) \in dom(A)$. For a given set X of attributes, let $Tup(X)$ denote the set of all tuples over X; the injectivity assumption ensures that $Tup(X)$ is indeed a set. A **relation** r over X is a finite set of tuple over X, that is, $r \subseteq Tup(X)$**;** the set of all relations over X is denoted by $\mathrm{Rel}(X)$.

Table 3.1: Sample table

Callnumber	**…**	**Author**	**…**	**Title**	**…**	**Publ_Co**
1	…	Date	…	Data Mining	…	WA
2	…	Ulman	…	Intro DBMS	…	CSP
3	…	Kroenke	…	DB Process	…	SRA
.	…	…	…	…	…	…

Table 3.1 illustrates that a relation over a set X of attributes can be represented as a table, if an ordering is imposed on the elements of X. When representing a relation in this form, the attributes will always be included as a 'headline'; the tuple or elements of the relation form the remaining rows of the table.

3.2.2 Data Pre-processing

The first phase of discovering the rules' interestingness and validity for the relational data is to ensure that only appropriate and clean data are used for the association rule mining process. In this chapter, two issues regarding the pre-processing techniques to be utilized in this thesis are defined.

3.2.2.1 Missing Data and Continuous Data

Data in the real world are often incomplete, noisy and inconsistent (Han & Kamber, 2001). The larger the dataset, the more likely that it contains some missing values. Malfunctioning measurement equipment, change in experimental design during data collection and the collecting of several similar but non-identical datasets are several reasons for the occurrence of missing data (Witten & Frank, 2005). In this thesis, the missing data in the database is defined as follows:

Suppose the focus is on relating a target variable Y to input attribute $X = (x_1,..,x_p)$, and there is missing data on a subset of X for some of the records in the database.

Another aspect of data pre-processing involves effective ways of dealing with continuous data. A variable X is said to be continuous if its set of possible values is an entire interval of numbers. This will be formally clarified with a definition of a continuous function as stated in (Borowski & Borwein, 1989):
A real function $y = f(X)$ is said to be continuous at a point A iff it is defined at $X = A$ and for both $X > A$ and $X < A$ the following condition holds:

$$\lim_{X \to A} f(X) = f(A).$$

that is precisely if,

for every $\varepsilon > 0$ there exists a $\delta > 0$ such that
$|f(X) - f(A)| < \varepsilon$ for all X such that $|X - A| < \delta$.

Given the lower bound X_L and the upper bound X_U of the domain of a continuous attribute X, values of X are obtained by a function that is continuous over all the points in interval $[X_L, X_U]$. In other words, a valid value (x_{val}) for X is any real number in the range $[X_L, X_U]$ (i.e. $X_L, \leq x_{val} \leq X_U$).

Based on the definition, the number of possible values for continuous attributes is infinite. To ensure that a manageable data size is obtained by reducing the number of distinct values per attribute, a common approach is to partition the ranges of continuous attributes into intervals (Han & Kamber, 2001). This is referred to as

binning, and several existing techniques for this include equal-width binning, equal-frequency binning and clustering-based binning (Han & Kamber, 2001).

In this thesis, for all continuous attributes involved in the relational data analysis, the equal-width binning approach method is applied. The equal-width binning approach groups the data into several buckets or bins of the same interval size. The equal width binning will be implemented based on the following steps (Refaat, 2007); 1) Calculate the range of variable to be binned; 2) Using the specified number of bins, calculate the boundary (width) of each bin; 3) Using specified boundaries, assign each value of the variable to a bin for each record.

Formally, the process can be expressed as follows. Let the domain of possible values of a variable X be bounded by range $[X_L, X_U]$ (where $X_L < X_U$), then several bins B_1, B_2, ..., B_n, are selected such that width(B_i) = (X_U - X_L,) / n, The interval of each bin is represented as $[Bi_L, Bi_U]$, and hence $X_L = B_{1L}$, $B_{1U} < B_{2L}$,...., $B_{nU} = X_U$ and hence each value x_{val} of X is assigned to a bin B_i, *if* $Bi_L \leq x_{val} \leq Bi_U$.

3.2.2.2 Data Partition

Sampling is that part of statistical practice concerned with the selection of individual observations intended to yield some knowledge about a population of concern, especially for the purposes of statistical inference (Cochran, 1963). A good sample must be more or less representative of the population from which it was selected (Ehrenberg, 1975). Although there is very little chance that the sample and population are identical, both of them are expected to be close. The sampling distribution provides a way of measuring the closeness between sample and population. It plays a crucial role in the process because an approximate measure will enable us to make a statistical inference. Here, three types of sampling approaches are compared (Table 3.2).

Table 3.2: Sampling Types reproduced from (Keller & Warrack, 2003)

Sampling Types	Sampling Plan
Simple Random:	A sample is selected in such a way that every possible sample with the same number of observations is equally likely to be chosen.
Stratified Random Sampling	A sample is selected by separating the population into mutually exclusive sets, or strata, and then drawing simple random samples from each stratum.
Cluster Sampling	A simple random sample from groups of clusters of elements.

In this thesis, the experimental relational data used is readily available for data mining evaluation purposes (Frank & Asuncion, 2010), and a simple random sampling type will suffice. The holdout evaluation approach is utilized (Webb, 2007) whereby the available dataset is divided into training set and testing set. The former is used for association rules generation, statistical analysis, and redundancy and contradictive assessment methods; and then the latter is used to verify the accuracy and coverage rates of the discovered rules on "unseen data".

3.3 General Concepts and Definitions of Semi-structured Data

While relational database systems have been utilized and highly recognized in relational data domains, various kinds of new data structures have emerged. The new data structures such as spatial data, hypertext and multimedia data, time related data, scientific data and World Wide Web data require advanced and efficient methods for storing, handling and manipulating such data. Semi-structured data representation offers an easy way to express relationships in these new data structures. Structure comparison of semi-structured data objects can often reveal valuable information (Tan, Hadzic, Dillon, & Chang, 2008). The general definitions of relational, semi-structured and unstructured data have been discussed in Chapter 1. Table 3.3 summarizes the differences between relational, semi-structured and unstructured data

Table 3.3: Relational, Semi-structured and Unstructured Data

Type of Data	Field Name	Field Value
Relational	The *Field Name* is defined	• Format is totally defined and prescribed • Simple data type
Semi-structured	The *Field Name* is recognized as *Tag Name*, and the *Tag Name is* defined	• Field value recognized as *Tag Value* • May be in free format text, prescribed or un-prescribed format • In many cases the field value is un-prescribed
Unstructured	No *Tag Name* is defined	• No *Tag Value* is defined

Semi-structured data as defined by (Suciu, 1998) is data that has no absolute schema or class fixed in advance, is implicit and irregular, nested and heterogeneous. Moreover, (Zhang, Ling, Bruckner, Tjoa, & Liu, 2004) define semi-structured data as a heterogeneous collection of ill-structured data that have no rigid structures. The following definition is taken from (Suciu, 1998): Semi-structured data is represented by a collection of *objects*. Each object can be atomic or complex. The value of an atomic object may consist of several types such as integer, string, image, sound, etc. The value of a complex object is a set of (attribute, object) pairs.

In this thesis, the scope with respect to semi-structured data is limited to XML documents and tree-structured data in general, rather than including more complex data types such as images and graphs. Hence, the respective thesis study is applicable to any semi-structured data where the objects are organized in a hierarchical way. The information within such data can be effectively represented as a rooted, ordered and labelled tree structure (Hadzic, Tan, & Dillon, 2011). Table 3.4 summarizes the analogy between the semi-structured data in general and XML as one of its specific types, as was illustrated in (Suciu, 1998).

Table 3.4: Analogy between Semi-structured Data Model and XML reproduced from (Suciu, 1998)

Semi-Structured Data Model	**XML**
Attribute	Tag
Object	Element
Atomic values: string, int, float, video	Character data, strings

In the next section, an XML document is utilized as an example to indicate how the information from XML documents can be modelled as a tree. The definitions of general tree concepts and the frequent subtree mining problem are provided in Section 3.5.

3.3.1 Modelling XML Documents

XML is a popular approach used to represent semi-structured data. Originally, XML "Extensible Mark-up Language" was a W3C standard for data exchange in the web. In this section, XML concepts and definitions in relation to mining association rules are discussed. Figure 3.1 shows an example of an XML fragment from the SIGMOD Record database.

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<SigmodRecord>
  <volume>15</volume>
  <number>2</number>
        <article  Code="152006">
           <title>Abstraction in recovery management</title>
           <authors>
             <author AuthorPosition="01">J Eliot B Moss</author>
             <author AuthorPosition="03">Marc H Graham</author>
             <author AuthorPosition="02">Nancy D Griffeth</author>
           </authors>
         </article>
         <article Code="152037">
            <title>A formal view integration method</title>
            <authors>
              <author AuthorPosition="02">Bernhard Convent</author>
              <author AuthorPosition="01">Joachim Biskup</author>
            </authors>
         </article>
</SigmodRecord>
```

Figure 3.1: Example of XML fragment

The following sections demonstrate how XML data can be viewed as a tree structure. This allows one to approach XML data from the perspective of the tree-structured format.

3.3.1.1 XML Nodes

(Feng, Dillon, Weigand, & Chang, 2003) have differentiated XML nodes into *simple* and *complex* nodes. Nodes that have no edges emanating from them are considered as simple or basic nodes. In a tree structure format, this type of node is called a 'leaf node'. Complex nodes can be identified as internal nodes. There are two important relationships that can be constructed from the complex nodes, namely parent-child and ancestor-descendant. This relationship is discussed in detail in Section 3.5.1.

From Figure 3.1, representatives of simple nodes are <volume>, <number>, <title>, and <author>. These do not have any children and/or descendants. The complex node example is <SigmodRecord>, <article> and <authors>.

3.3.1.2 Element-attribute Relationships

There is a significant value for relationships between element-attribute in XML (Hadzic et al., 2011). As the element-attribute relationships are represented as a tree structure, these can be depicted as a node with multi-labels and the level of relationship among the element-attributes are of equal value and are equally important.

3.3.1.3 Element-element Relationships

In constructing the hierarchical relationships using a tree structure, one needs to know the relationship between elements in the XML. Generally, parent-child relationships and ancestor-descendant relationships are two types of relationship that exist between two elements. A parent-child relationship is one where there are two elements connected by one edge. Conversely, two elements that are connected by more than one edge are defined as having an ancestor-descendant relationship. In defining both types of relationship, these two elements need to be from different

levels. For example, from Figure 3.1, the relationship between elements <authors> and <author> is that of parent-child, while the relationship between elements <article> and <author> is that of ancestor-descendant. If two elements are on the same level and belong to the same parent, the relationship between them is a sibling relationship (e.g. <volume> and <number> in Figure 3.1). Since there is no edge connecting sibling nodes, it is more a virtual relationship. An example of both element-element and element-attributes relationships is shown in Figure 3.2

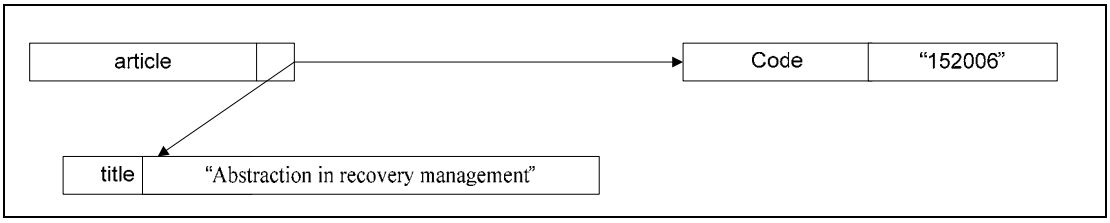


Figure 3.2: Illustration of element-element (article-title) and element-attribute (article-code) relationships

3.3.1.4 Tree-structured Items

XML is constructed based on tree-structured items. One can differentiate XML data from relational data by the existence of atomic items and a one-to-one relationship between items in itemsets (Agrawal & Srikant, 1994; Han & Kamber, 2001). Conversely, XML data contains more complicated hierarchical relationships than do relational data between tree-structured items. Examples of tree-structured items from Figure 3.1 are shown below in Figure 3.3.

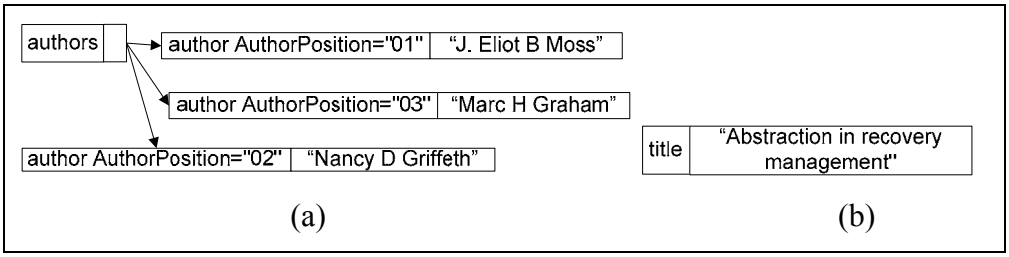


Figure 3.3: Illustration of tree-structured items with size 4 (a) and with sizes 1 (b)

3.3.2 Parallelism between XML and Tree Structure

XML data can be represented as a tree-like structure, as a tree is an acyclic connected graph. The parallelism between XML and tree structure as defined by (Hadzic et al., 2011) was reproduced and utilized in this thesis. The definition of a rooted tree and how it parallels XML data is described first. A rooted tree is a tree in which one of the vertices is distinguished from others, and is called the root. The XML data is a rooted and well-formed tree as defined by (Feng et al., 2003). In the case of XML data, 'node' refers to a tag or element. If there is an ordering imposed on the child nodes of each node, then it can be considered as a rooted ordered tree. Hence, if there are k nodes as children of say node A, a node at the left-most position would be at position 0, and the nodes at its right will have an incrementing position numbering up to the $(k\text{-}1)^{th}$ child. Conversely, a rooted unordered tree has no such ordering imposed on children nodes. A labelled tree is one in which each of its nodes is associated with a label. Two or more nodes may have the same label.

Based on the aforementioned characteristics, an XML document has a hierarchical document structure comprised of certain elements. The XML may contain further embedded elements to which a number of attributes can be attached. Moreover, elements that form a sibling relationship may have a certain ordering imposed on them. Each element of an XML document has a name and value. In certain applications, if only the XML structure is to be considered, then only the element names need to be utilized (Hadzic et al., 2011).

Hence, it has been demonstrated that there are strong parallelisms between XML data and the tree structure. The XML data can be modelled through the abovementioned constructs and definitions. Thus, from this point forward, this thesis will concentrate on discussing the tree structure. The XML is assumed to be an instance of tree structure, so it can be assumed that the techniques developed for the tree structure can work similarly for XML.

3.4 Association Rule Mining

Association rule mining in its most fundamental structure is used to discover interesting relationships among items in a given dataset (Han & Kamber, 2001).

There are two major processes in association rule mining as defined by (Agrawal, Imieliski, & Swami, 1993). The first step is to find all frequent itemsets. The occurrences of these itemsets need to meet at least certain pre-determined thresholds. Next, a frequent rule is considered interesting and strong if it satisfies certain criteria such as the *support* and *confidence* thresholds. The first task is a pre-requisite for the second and is the more complex task. The general concept of and terms used in association rule mining will be described next.

3.4.1 Frequent Itemset Mining from Relational Data

Let *D* denote a database of transactions, where each transaction has unique identifier (*tid*) and contains a set of items. Let $I = \{i_1, i_2, ..., i_{|D|}\}$ the set of distinct items in D. The set of all *tids* is denoted $T = \{i_1, i_2, ..., i_n\}$. A set $X \subseteq I$ is called an *itemset*. An itemset with *k*-items is called a *k*-itemset. Let $t(X) \subseteq T$, denote the set consisting of all the transaction *tids* which contain *X* as a subset (referred to as *tidset* of *X* in (Gouda & Zaki, 2001)). The *support* of an itemset *X*, denoted as $\sigma(X)$, is the number of transactions in which that itemset occurs as a subset. Thus $\sigma(X) = |t(X)|$. The frequent itemset mining task can then be defined as: given a database of transactions *D,* and user specified *minimum support* threshold σ, find all itemsets *X* from *D* where $\sigma(X) \geq \sigma$.

Note that there are two ways of expressing the support ratio σ: either as an absolute value or a percentage. The absolute value reflects the exact number of transactions that need to contain an itemset *X* (as reflected in the definition above), while the percentage value is in respect to the percentage of total number of transactions that must contain *X*. Frequent itemset mining the most essential and crucial step in mining association rules, and will determine the overall performance of association rules mining (Kantardzic, 2002).

3.4.2 Rule Generation

The second phase in discovering association rules is straightforward. The confidence measure is used to determine the strength of the implication of a rule of form $x \rightarrow y$.

It is based on the conditional probability of a transaction containing a part of the frequent itemset identified as the consequent of the rule (y), if the transaction contains the part of the itemset identified as antecedent of the rule (x), and is calculated as *confidence* $(x \rightarrow y) = \sigma(x \rightarrow y) / \sigma(x)$.

The association rule generation definition is extracted from (Han & Kamber, 2001). Given the *minimum support* threshold σ, the minimum *confidence* threshold τ, and the set of frequent itemsets $L = \{X_1, X_2, ..., X_{|L|}\}$ discovered where $\sigma(X_i) \geq \sigma \ \forall i = \{1, ..., |L|\}$, association rules can be generated as follows:

- For each frequent itemset $X_i \in L$, generate all non-empty subsets of X_i.
- For every non-empty subset s of X_i, output the rule "$s \rightarrow (l - s)$" if $\frac{\text{sup\_}count(l)}{\text{sup\_}count(s)} \geq min\_conf$, where min_conf is the minimum confidence threshold.

The rule with low *support* and high *confidence* is often considered as an interesting rule. In addition to *confidence*, numerous methods exist for measuring the interestingness of rules as was discussed in Chapter 2.

An early solution to this two-phase problem has been proposed by (Agrawal et al., 1993), namely the Apriori algorithm. The Apriori algorithm has been favoured for frequent itemset generation as it offers a good performance on sparse data.

However, since a large number of rules are returned from this type of rule generation approach, an additional rules pruning scheme might be needed. The large volume of rules may nevertheless impose rule complexity problems. The complexity of rules is defined as:

Definition 1: For a transaction that contains *n*-items, the space complexity is usually of the order 2^n.

These complexity issues have motivated researchers to focus on discovering closed and maximal frequent itemsets, as they are much smaller in size and the complete set of frequent itemsets can still be obtained.

3.4.3 Maximal and Closed Frequent Itemsets

A frequent itemset is called *maximal* if it is not a subset of any other frequent itemset, while a frequent itemset is called *closed* if it has no proper superset with the same *support*. The mining of maximal itemsets (Bayardo, 1998; Burdick, Calimlim, Flannick, Gehrke, & Yiu, 2005; Gouda & Zaki, 2001) and closed itemsets (Pasquier, Bastide, Taouil, & Lakhal, 1999; Pei, Han, & Mao, 2000; Zaki & Hsiao, 2002) has been proposed and implemented to reduce the complexity of the rule generation task. This will result mainly in a reduction of the mining computational cost without incurring loss of information, as all frequent itemsets can be generated from a set of maximal or closed patterns. In addition, for closed patterns, the exact support information for each frequent itemset can be worked out. Both the maximal itemset mining and the closed itemset mining are preferred frameworks for generating association rules from hard and dense datasets.

The majority of the works mentioned earlier tend to focus more on the structural and analytical comparative study of the algorithms performance in generating the rules as discussed in (Yahia, Hamrouni, & Nguifo, 2006). Even the closed and maximal itemsets are known to reduce the rule set size but the question still remains whether they thereby lose the coverage rate and have good generalization power. The use of closed itemsets for prediction and discrimination as described in Chapter 2 was investigated in (Garriga, Kralj, & Lavrač, 2008), but no comparison was done with maximal or frequent itemsets, and the incorporation of statistical analysis, redundancy and contradictive assessment methods in closed itemsets was not studied. Thus, an evaluation of the usefulness of maximal/closed itemsets for the classification task, and their generalization power and coverage rate are essential in order to produce high quality rules. Formal definitions of frequent itemsets, maximal itemset and closed itemset are provided in Table 3.5.

Table 3.5: General comparison between Frequent Itemset, Maximal Itemset and Closed Itemset characteristics

Frequent Itemset	**Maximal Itemset**	**Closed Itemset**
An itemset X is frequent if its support is more than or equal to some threshold minimum support (*min_sup*) value, i.e if $\sigma(X) > min\_sup$	If X is frequent, an itemset X is a *Maximal* itemset if it is not a subset of any other frequent itemset.	If X is frequent, an itemset X is a Closed *itemset* if there exists no itemset X' such that X' is a proper superset of X, and every transaction containing X also contains X'.

3.5 Frequent Subtree Mining from Tree-structured Data

Due to the hierarchical document structure, XML documents are frequently modelled using a rooted ordered labelled tree. (Feng et al., 2003) proposed an XML-enabled association rule framework. It extends the notion of associated items to XML fragments to present associations among trees rather than simple-structured items of atomic values. Unlike classical association rules where associated items are usually denoted using simple structured data from the domains of basic data types, the items in XML-enabled association rules can have a hierarchical tree structure. The adaptation of association mining to the XML document as shown in (Feng et al., 2003) results in a more flexible and powerful representation of both simple and complex structured association relationships inherent in XML documents.

The main problem in association mining from semi-structured documents such as XML, is that of frequent pattern discovery, where a pattern corresponds to a subtree in this case, and a transaction to a fragment of the database tree whereby an independent instance is described. This problem is more complex than in traditional frequent pattern mining from relational data because structural relationships need to be taken into account. It is known as the ***frequent subtree mining*** problem, and can be generally stated as (Hadzic et al., 2011): Given a tree database T and minimum support threshold (σ), find all subtrees that occur at least σ times in T. Furthermore, depending on the domain of interest and the task that is to be accomplished in a particular application, different types of subtrees can be mined using different support definitions (Hadzic, Dillon, & Chang, 2008).

3.5.1 General Tree Concepts

To lay the necessary ground for describing the general aspects of the frequent subtree mining problem, definitions of related tree concepts are provided in this section. These definitions were derived from (Tan et al., 2008). A tree is described as an acyclic connected graph with one node defined as the root. It consists of a set of *nodes* (or *vertices*) that are connected by *edges*. There are two nodes associated with each *edge*. A *path* is defined as a finite sequence of edges and in a tree there is a single unique path between any two nodes. The *length of a path p* is the number of edges in *p*. A rooted tree has its top-most node defined as the *root* that has no incoming edges and for every other node, there is path between the root and that node. A node *u* is said to be a *parent* of node *v*, if there is a directed edge from *u* to *v*. Node *v* is then recognised as a *child* of node *u*. Nodes with no children are referred to as *leaf* nodes; otherwise, they are called *internal nodes.* The s*ibling* nodes are those nodes with the same parent. The *fan-out/degree* of a node is the number of children of that node. The *ancestors* of a node *u* are the nodes on the path between the root and *u*, excluding *u* itself. The *descendants* of a node *v* can then be defined as those nodes that have *v* as their ancestor. Nodes with a common ancestor that is not a parent are referred to as *cousins*. A tree is *ordered* if the children of each internal node are ordered from left to right. In an ordered tree, the last child of an internal node is referred to as the *rightmost* child. The *rightmost path* of *T* is the path connecting the rightmost leaf node with the root node. The *level/depth* of a node is the length of the path from root to that node. The *Height of a tree* is the greatest *depth* of its nodes.

3.5.2 Subtree Types

Induced and *Embedded* are two types of subtrees that have consistently been used in frequent subtree mining. An *induced* subtree preserves the parent-child relationship on each node of the original tree. Additionally, the *embedded* subtree preserves both the ancestor–descendant relationship over several levels and the parent-child relationship. In an *ordered tree* the children of every internal node are ordered from left to right and the ordering descriptor is essential. Examples of induced and embedded subtree are given in Figure 3.4.

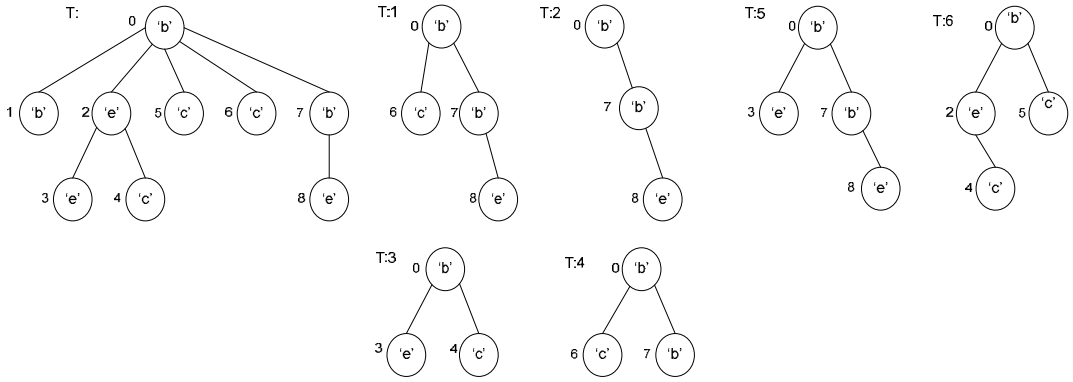


Figure 3.4: Example of induced subtree (*T1, T2, T4, T6*) and embedded subtrees (*T3, T5*) of tree *T*

The formal definitions of induced and embedded subtrees are as follows (Tan, Dillon, Hadzic, Chang, & Feng, 2006):

Induced Subtree. A tree $T' = (v', V', L', E')$ is an ***ordered induced subtree*** of tree $T = (v, V, L, E)$ iff (1) $V' \subseteq V$, (2) $L' \subseteq L$, and $L'(v) = L(v)$, (3) $E' \subseteq E$, (4) $\forall v' \in V', \forall v \in V$ and v' is not the root node parent(v') = *parent(*v), (5) the left to right ordering of sibling in T' should be preserved. Induced tree T' of T, can be obtained by repeatedly removing leaf nodes or the root node if its removal does not create a forest in T.

Embedded Subtree. A tree $T' = (v', V', L', E')$ is an ***ordered embedded subtree*** of a tree T if and only if it satisfies property 1,2,3,5 of induced subtree and it generalizes property (4) such that (4) $\forall v' \in V', \forall v \in V$ and v' is not the root node ancestor(v')= *ancestor(*v).

3.5.3 Support Definition

Generally, two support definitions have been used to determine the support of a subtree t, usually denoted as $\sigma(t)$ for frequent subtree mining framework. These are the transaction-based support and the occurrence match support (also referred to as weighted support) (Chi, Muntz, Nijssen, & Kok, 2005; Tan, Dillon, Hadzic, Chang, & Feng, 2005; Zaki, 2005). The term *transaction* was originally introduced in the data management field in reference to an atomic interaction with a database management system. Conversely, in the data mining field, the term 'transaction' has

adopted a different meaning. To clarify its use in relation to tree mining, the following definition is suitable: A transaction is a set of one or more items obtained from a finite item domain, and hence a dataset is a collection of transactions (Bayardo, Agrawal, & Gunopulos, 2000). Hence, in terms of a tree database, a transaction would correspond to a fragment of the database tree whereby an independent instance is described.

In this thesis, the application of transaction-based support is utilized. The definition of the ***transaction-based support*** *(TS)* is as follows: the transactional support (σ) of a subtree t, denoted as $\sigma_{tr}(t)$ in a tree database T_{db} is equal to the number of transactions in T_{db} that contain at least one occurrence of subtree t.

Definition 2: Let the notation $t \prec k$, denote the support of subtree t by transaction k, then for TS, $t \prec k = 1$ whenever k contains at least one occurrence of t, and 0 otherwise. Suppose that there are N transactions k_1 to k_N of tree in Tdb, the $\sigma_{tr}(t)$ in T_{db} is defined as:

$$\sum_{i=1}^{N} t \prec k_i$$

The support definition in frequent itemset mining from relational data can be determined by the existence of an item within a transaction. Thus, in engaging to frequent subtree mining from traditional frequent itemset mining, the application of transaction support offers a better way of defining the support thresholds. Additionally, as a transition from relational data to XML data, mutual properties, such an instance in relational data can be described as one transaction in XML data. This has made transaction-based support the focus of many tree mining works and it is simpler than the occurrence-match support (Tan et al., 2008).

The ***occurrence-based support*** (OC) takes into account the repetition of items in a transaction and counts the subtree occurrences in the database as a whole. Hence, for OC, the support (σ) of a subtree t, denoted as $\sigma_{oc}(t)$ in a tree database T_{db} is equal to the number of occurrences of t in all transactions in T_{db}.

Definition 3: Let the function *g(t,k)* denote the total number of occurrences of subtree t in transaction k. Suppose that there are N transactions k_1 to k_N of tree in T_{db}, $\sigma_{oc}(t)$ in T_{db} can be defined as:

$$\sum_{i=1}^{N} g(t,ki)$$

Due to the nature of the domain being considered and the data used, the focus in this thesis is mainly on validating rules obtained by mining **ordered induced/embedded subtrees** under a **transaction-based support definition.**

3.6 Problems to be addressed

The problem focused upon and addressed in this thesis is the evaluation of association rules generated from both relational and tree-structured data. This research is intended to investigate how association rules mining, statistical analysis, and redundancy and contradictive assessment methods can be utilized, and to develop a proper sequence of use of these techniques to arrive at a more reliable and interesting set of association rules based on:

(a) Frequent itemset mining, specifically the Apriori, Maximal and Closed approaches discovered from the relational data (details will be given in Chapters 5 and 6); and

(b) Frequent subtree mining and frequent subtrees generated from structure-preserving flat format for tree-structured data (details will be given in Chapter 7).

Discovering useful and interesting patterns is one of the main tasks of data mining applications. Ideally, a pattern is considered interesting and useful if it is comprehensible, valid on test data and new unseen data, potentially useful, actionable and novel (Han & Kamber, 2001). However, (Han & Kamber, 2001) claim that, while patterns discovered from the data mining approach are considered strong, not all of them are interesting. (Lenca, Meyer, Vaillant, & Lallich, 2008) assert that there are mainly two problems in dealing with pattern selection, namely the quantity and the quality of the rules. The quantity of the rules refers to the

problem of generating a large volume of output. The quality issues are concerned with the rules potentially reflecting real significant associations in the domain under investigation. Some of the rules discovered are due to pure coincidence resulting from certain randomness in the particular dataset being analyzed. The rules can be either a true discovery or merely an artefact of random association.

Our work in the area of rules interestingness measures is motivated by the objective interestingness measures which are based on probability theory, statistics and information theory. Various objective interestingness criteria have been used to limit the nature of the rules extracted. Generally speaking, interesting rules can be interpreted as those rules that have a sound statistical basis and are neither redundant nor contradictive. Such an approach requires statistical analysis, and redundancy and contradictive assessment methods to verify and evaluate the usefulness and quality of the rules discovered. This aim is to filter out the redundant, misleading, random and coincidentally occurring rules, while at the same time sustaining the accuracy of the rule set and retaining valuable rules.

Many interestingness- and constraint-based measures have been successfully utilized in previous works. However, there is still a need to understand the roles of these parameters and the way in which they should be utilized. Thus, in this thesis, the problem of developing systematic ways to verify the usefulness of rules obtained from association rule mining will be addressed (Shaharanee, Hadzic, & Dillon, 2009; Shaharanee, Hadzic, & Dillon, 2011). These problems include, understanding the role which different parameters play, the way in which different statistical measures can be utilized and the sequence of their use. A unified framework, that combines several techniques to assess the quality and remove any redundant and unnecessary rules, is proposed. This framework will create a means whereby interestingness- and constraint-based parameters can be utilized and sequenced.

The problems and the implication of different confidence values and the time at which the constraint is applied will also be addressed. In addition, while confidence measures are often used to reduce the rule set size to only those reflecting highly confident association, no study has been performed on the implication of using different confidence values and the differences of applying this constraint at various

stages of the rule verification process. Thus, this thesis also focuses on the problems of evaluating the impact on classification accuracy, generalization power, and rule coverage rate, when rules are generated using: frequent, closed and maximal itemset mining algorithms for relational data; frequent subtree mining and frequent subtrees generated from flat data table based on database structure model (*DSM)*(Hadzic, 2011) approach from tree-structured data; and, when different confidence measures are used and applied at different stages of the verification process.

3.6.1 Classification and Prediction Problems for Evaluating the Frequent Patterns

The number of patterns/association rules generated through frequent itemsets mining and frequent subtree mining can be quite large, while usefulness of each rule for the classification/prediction task may be limited. As a large volume of rules will be removed based on the statistical analysis, and redundancy and contradictive assessment method, another crucial issue arises: whether the quality of the rules obtained from the proposed framework has been compromised. Here, the quality of rules is demonstrated based on their accuracy and coverage values. Accuracy rate (AR) is typically defined as the number of correctly classified instances, while the number of incorrectly classified instances is referred to as a misclassification rate (MR). Additionally, coverage rate (CR) refers to the percentage of captured/covered instances from the database. Thus, our aim is to evaluate these extracted rules in terms of correctly predicting the class value from the training datasets (known as classification accuracy), and correctly predicting the class value from the testing/unseen dataset (known as predictive accuracy). They are also evaluated for their coverage rate on both training and testing datasets.

In our framework, the rule set is reduced by applying the features subset selection, statistical analysis, redundancy and contradictive assessment method. Simple rules are preferred as they are easier to comprehend and are expected to perform better on unseen data since they are more general and increased the generalization ability (i.e., typically rule set that contains redundant rules is more specialized). The trade-off is measured between the accuracy rate (AR) and coverage rate (CR) of the rule sets.

When reducing the rule set, both the *AR* and *CR* should be maximized. One can simplify the rules by reducing the overall rule set size and the number of attribute constraints in the rule. Decreasing the number of rules usually leads to the increase in *AR* of that rule but at the cost of a decrease in *CR* of that rule. Conversely, if the number of rules is too large, they may lack the specificity to distinguish some domain characteristics, and hence the *AR* would decrease. Generally speaking, an optimized rule set should be either more accurate than the original rule set and/or the balance between the trade-off factors should be much greater. For example, if there are many rules with small *CR* but very high *AR*, a rule set with a significantly smaller number of rules may be preferred even at the cost of a decreased AR.

In this thesis, the rule accuracy and coverage will be measured at every stage and for each sequence of filtering involved. This measure is crucial as it can determine the quality of the discovered rules. Additionally, this analysis also exhibits the balancing/optimization issues with regards to the trade-off between accuracy rate and coverage rate. Consequently, the task of choosing optimal stopping criteria based on a *minimum confidence* threshold will be discussed and presented to ensure that an acceptable level of accuracy and coverage rates can be achieved. The detailed analysis and explanation of the rules quality and optimization issues will be outlined in Chapters 5, 6 and 7.

In reference to the problems to be addressed, the frequent patterns from both relational and tree-structured data are considered in the evaluation process; hence, the following are the definitions for both types of frequent patterns evaluated in this thesis.

Relational Data: Let us denote the set of frequent *k*–itemsets as F_k , and the set of all frequent itemsets as *FI* . A frequent itemset is called *maximal* if it is not a subset of any other frequent itemset from *FI*. The set of all maximal itemsets is denoted as *MFI*. A frequent itemset is called *closed* if it has no proper superset in *FI* with the same *support*. The set of all frequent closed itemsets is denoted as *CFI*.

In terms of the relational data, the focus is on evaluating the rules discovered using the Apriori, Maximal and Closed approaches and that satisfy the minimum *support*

thresholds. The set of the frequent patterns generated by each aforementioned approach is denoted as FI, MFI and CFI. The datasets with predefined class label are utilized, where one of the attributes from the dataset is considered as a class to be predicted. Thus, only the patterns/rules from FI, MFI and CFI that contain this class attribute are considered.

Let the frequent itemsets from Apriori, Maximal and Closed that have a class label (value) be denoted as FIC, $MFIC$ and $CFIC$, respectively. Let the accuracy of these be denoted as *ac(FIC)*, *ac(MFIC)* and *ac(CFIC)*, and coverage rate as *cr(FIC)*, *cr(MFIC)* and *cr(CFIC)*. The general aims of removing low quality (e.g. not interesting, redundant) rules with respect to the accuracy and coverage rate of both training and testing dataset can be defined as follows:

Aim 1. Given FIC with accuracy *ac(FIC)*, obtain $\tilde{FIC}$, such that $\tilde{FIC} \supset FIC$, $ac(\tilde{FIC}) \geq (ac(FIC) - \varepsilon)$ and $cr(\tilde{FIC}) \geq (cr(FIC) - \varepsilon)$

Aim 2. Given $MFIC$ with accuracy *ac(MFIC)*, respectively, obtain $\tilde{MFIC}$ such that $\tilde{MFIC} \supset MFIC$, $ac(\tilde{MFIC}) \geq (ac(MFIC) - \varepsilon)$ and $cr(\tilde{MFIC}) \geq cr(MFIC) - \varepsilon)$

Aim 3. Given $CFIC$ with accuracy *ac(CFIC)*, respectively, obtain $\tilde{CFIC}$ such that $\tilde{CFIC} \supset CFIC$, $ac(\tilde{CFIC}) \geq (ac(CFIC) - \varepsilon)$ and $cr(\tilde{CFIC}) \geq (cr(CFIC) - \varepsilon)$

Note that in the above definition ε is an arbitrary user defined small value and ε is used to reflect the noise that is often present in real-world data.

Tree-structured Data: Let the set of frequent subtree patterns extracted from tree-structured data be denoted as *SF*. Please note that for the first problem setting the patterns from *SF* have not been assigned a particular class label to be used for a prediction/classification task, and as such, simply reflect the frequently occurring associations that may not necessarily have a sound statistical basis.

Hence, in the first problem setting for deriving frequent subtrees from tree-structured data, the aim is to reduce the *SF* by filtering out the patterns that are not statistically significant with respect to the statistical measures used.

In the second problem setting, the discovered frequent subtrees are defined as a subtree which consists of a certain preferred node. One of the attributes of the data is considered as a class to be predicted for classification task purposes. Hence, only those patterns from *SF* that contain this class attribute are considered, as they will represent the set of values that frequently occur together when a particular class value is present.

For tree-structured data, the focus is on evaluating frequent subtrees, and rules based on embedded and induced subtrees that satisfy minimum *support* and *confidence* thresholds. Let us denote the subtree patterns from the frequent subtree set *SF* that has a class label (value), as *SFC,* their accuracy as *ac*(*SFC)* and coverage rate as *cr*(*SFC*). The aim of removing low quality rules (based on frequent subtree patterns) with respect to the accuracy and coverage rate of both training and testing dataset can be defined as follows:

Aim 4. Given *SFC* with accuracy *ac(SFC)*, obtain $SFC' \supset SFC$, such that $ac(\tilde{SFC'}) \geq (ac(SFC) - \varepsilon)$ and $cr(SFC') \geq (cr(SFC) - \varepsilon)$ (ε is an arbitrary user defined small value used to reflect the noise that is often present in real-world data).

In this thesis, the aim is to investigate the use of statistical analysis, and the redundancy and contradictive assessment methods for the above problems, and to determine how the existing interestingness measures and parameters can be utilized effectively and in the correct sequence.

3.6.2 Feature Subset Selection to Determine Relevant Attributes

As the focus of this thesis is on evaluating the frequent patterns, one important property of the frequent pattern-based classifier is that it generates frequent patterns without considering their predictive power (Cheng, Yan, Han, & Hsu, 2007). This property will result in a huge feature space for possible frequent patterns.

Feature subset selection is one of the steps performed in the pre-processing stage of the data mining process to remove any irrelevant attributes. If the whole dataset were used as input, this would produce a large number of rules, many of which are created or made unnecessarily complex by the presence of irrelevant and/or redundant attributes. Determining the relevant and irrelevant attributes poses a great challenge to many data mining algorithms (Roiger & Geatz, 2003). If the irrelevant attributes are left in the dataset, they can interfere with the data mining process and the quality of the discovered patterns may deteriorate, creating problems such as overfitting (Cheng et al., 2007). Furthermore, if a large volume of attributes is present in a dataset, this will slow down the data mining process. To overcome these problems, it is important to find the necessary and sufficient subset of features so that the application of association rules mining will be optimal and no irrelevant features will be present within the discovered rules. This would prevent the generation of rules that include any irrelevant and/or redundant attributes.

The feature subset selection problem to be addressed in this thesis can be more formally described as:
Given a relational database D, $AT = \{at_1, at_2, ..., at_{|AT|}\}$ the set of input attributes in D, and $Y = \{y_1, y_2, ..., y_{|Y|}\}$ the class attribute with a set of class labels in D. Let an association rule mining algorithm be denoted as $AR_{AL,}$ the set of association rules for predicting the value of a class attribute Y from D extracted using AR_{AL} as $AR(D)$, and accuracy of $AR(D)$ as $ac(AR(D))$. The problem of feature subset selection is to reduce D into D' such that $AT' \subseteq AT$ and $ac(AR(D') \geq ac(AR(D') - \varepsilon$, where ε is an arbitrary user defined small value to reflect noise present in real-world data. In other words, the task is to find the optimal set of attributes, $AT_{OPT} \subseteq AT$, such that the accuracy of the association rule set using AR_{AL} is maximized.

3.7 Chosen Methodologies

The purpose of this section is to give a brief overview of some common methodologies for approaching the problems discussed in this chapter, and to indicate the direction taken in the work performed in this thesis. More details about

the characteristics of the particular methodology adopted in this thesis will be provided in the proposed framework overview in Chapter 4.

3.7.1 Feature Subset Selection

Feature subset selection is an important pre-processing step in the data mining process. The feature subset selection task is utilized in this thesis purposely to determine irrelevant attributes in predicting the class variable. The removal of these attributes will result in a much smaller dataset, thereby reducing the number of rules that need to be generated from the association rule mining algorithm, while closely maintaining the integrity of the original data (Han, Kamber, & Pei, 2012). Additionally, rules described with fewer attributes are also expected to perform better when classifying future cases; hence, they will have better generalization power than do the more specific rules that take many attributes into account. Besides, the patterns extracted will also be simpler and easier to analyse and understand. (Zhou & Dillon, 1991) initiated a statistical–heuristic feature selection recognized as Symmetrical Tau. This measure was derived from the Goodman and Kruskal Asymmetrical Tau measure of association, for cross-classification task in the statistical area. The Symmetrical Tau measure has proven to be useful for feature subset selection problems in decision tree learning.

The Symmetrical Tau statistical-heuristic feature relevance measure will be utilized in this thesis to provide the relative usefulness of attributes in predicting the value of the class attribute, and discard any of the attributes whose relevance value is fairly low. This prevents the generation of rules which then would need to be discarded anyway once it was found that they contain some irrelevant attributes. The common properties and advantages of using the Symmetrical Tau measure for feature subset selection problems will be discussed in detail in the next chapter.

3.7.2 Frequent Pattern Mining

Frequent pattern mining algorithms are utilized here to mine the association rules from both relational and tree-structured data. Within **relational data**, three algorithms are employed to systematically generate candidate rules, namely, the

Apriori, Maximal and Closed. Each of the algorithms is capable of generating frequent rules according to its detailed characteristics as mentioned in Section 3.4. While for **tree-structured** data, this involves *frequent* subtrees, and rules based on *embedded* and *induced* subtrees as described earlier in Section 3.5. This will offer a variation of characteristics of rules to be measured based on their interestingness and validity.

3.7.3 Rules Evaluation based on Statistical Analysis, Redundancy and Contradictive Assessment Methods

When a set of rules is generated from both relational and tree-structured data, it needs to be verified with proper statistical analysis. Thus, a proper statistical test is developed to verify each set of rules respectively. (Agresti, 1996; Hosmer & Lemeshow, 1989) establish and summarise statistical methods, such as the chi-squared test for correlation and measures of association, that have long played a prominent role, and they also emphasise the logistic regression modelling techniques.

3.7.3.1 Hypothesis Testing

Statistical inference is the process of inferring information about a population from a sample. Statistical inference can be utilized to obtain an estimate if one is willing to accept less than 100% accuracy. Because information about populations can usually be described by parameters, the statistical technique used generally deals with drawing inferences about population parameters from sample statistics. In reality, calculating the parameters for a population is virtually impossible because populations tend to be very large. Most parameters are not only unknown but also unknowable. Hypothesis testing is a most important tool when applying statistics to real-life problems. Most often, decisions are required to be made concerning populations on the basis of sample information. Statistical tests are used in arriving at these decisions.

There are two types of hypotheses: null hypothesis and alternative hypothesis. The testing procedure begins with the assumption that the null hypothesis is true. The goal is to determine whether there is enough evidence to infer that the alternative hypothesis is true. There are two possible choices:

- Conclude that there is enough evidence to support the alternative hypothesis
- Conclude that there is not enough evidence to support the alternative hypothesis

and two possible errors:

- Type I error = α, reject a true null hypothesis
- Type II error = β, do not reject a false null hypothesis

(Keller & Warrack, 2003) define testing the hypothesis approaches as:
Rejection region: If the test statistics fall with that range, the decision is to reject the null hypothesis in favour of the alternative hypothesis.
P-Value: test of probability of observing a test statistic at least as extreme as the one computed given that the null hypothesis is true.

3.7.3.2 Correlation Analysis

A strong association rule is defined as a rule that satisfies the minimum support and minimum confidence values. However, the *support-confidence* framework can be misleading in that it may identify a rule $A \rightarrow B$ as interesting when, in fact, the occurrence of A does not imply the occurrence of B (Han & Kamber, 2001).

The definition given by (Han & Kamber, 2001) is reproduced in describing the problem of 'misleading' strong association rules. The occurrence of itemset A is independent of the occurrence of itemset B if $P(A \cup B) = P(A)P(B)$; otherwise, itemsets A and B are dependent and correlated as events. The correlation between the occurrence of A and B can be measured by computing

$$corr(x, y) = \frac{P(x \cup y)}{P(x)P(y)} \quad (1)$$

If the resulting value in Equation 1 is less than 1, then the occurrence of A is negatively correlated with the occurrence of B. If the resulting value is greater than 1, then A and B are positively correlated, meaning the occurrence of one implies the occurrence of the other. If the resulting value is equal to 1, then A and B are independent and there is no correlation between them. Given the correlation in

Equation 3, the chi-squared statistic can be used to determine whether the correlation is statistically significant.

3.7.3.3 Regression Analysis

Regression methods have become an integral component of any data analysis process concerned with describing the relationship between a response variable and / or a more explanatory variable. It is often the case that the outcome variable is discrete, taking on two or more possible values. The purpose of developing regression models is to find the best fitting and most parsimonious and reasonable model to describe the relationship between an outcome (dependent/ response/input) variable and set of independent (predictor/explanatory/output) variables.

Here, the focus is on the development of the logistic regression model which is a regression model used when the dependent variable is dichotomous/binary/binomial (binary logistic regression) or has more than two levels (multinomial logistic regression) and the independents are of any type. As mentioned earlier, the goal of regression modelling is to select the dependent/input variables that produce the 'best' model within the context of the particular problem. This involves two primary tasks as asserted by (Hosmer & Lemeshow, 1989) which are: "1) *a basic plan for selecting thee variables for the model and* 2) *set of methods for accessing the adequacy of the model both in term of its individual variables and its overall fit*".

The application of logistic regression can be divided into five important tasks which are: to predict a categorical dependent variable on the basis of continuous and/or categorical independents; to determine the effect size of the independent variables on the dependent; to rank the relative importance of independent; to access the interaction effects and to understand the impact of covariate control variables.

In the context of this thesis, a logistic regression model is selected based on the most parsimonious model that explains the data. This can be achieved by minimizing the number of variables, which will result in a more generalized model. A model that consists of a large volume of variables will generate a large estimated standard error and becomes more dependent on the observed data (Hosmer & Lemeshow, 1989).

The development of the logistic regression model and the variable selection will be described in detail in Chapters 4 and 5.

3.7.3.3 Redundancy and Contradictive Removal

Frequent patterns in a data set are often redundant and unrelated (Wei, Yi, & Wang, 2006). In Chapter 2, it is shown that the weakness of the traditional association rule mining framework is that it produces many redundant rules (Zaki, 2000). While statistical tests such as correlation and regression analysis are able to discard non significant rules, the redundant rules still exist. Redundant rules as defined by (Webb, 2007) are those rules that include items in the antecedent that are entailed by the other elements of the antecedents.

Thus, in order to prevent the generation of redundant rules, the definition of *Productive* rules (Webb, 2007) is utilized which concern the *minimum improvement* constraints with improvement greater than zero. The improvement of rule $x \to y$ is defined as improvement $(x \to y) = confidence(x \to y) - \max_{z \subset x}(confidence(x \to y))$ (Bayardo et al., 2000).

Initial work on discovering contradictive rules such as that of (Padmanabhan & Tuzhilin, 1998) utilized the belief system in discovering any rules that conflict with user knowledge. This approach utilized the subjective-based measure to reduce the number of discovered association rules. Contradictive assessment are utilized in this thesis to identify and to remove any two or more rules that have the same pre-condition (i.e. antecedents) and imply different class values (consequents). The discussion on selecting a relevant data source for the data mining process from (Zhang & Zhang, 2001) is followed when defining contradictive problems.

The problem of contradictive rules can be defined as follows: Let $F(X) = \{fX_1, fX_2, \ldots, fX_{|F(X)|}\}$ be a set of class labelled rules from dataset *D*. Any two or more rules from $F(X)$, $fX_j, fX_k \in F(X)$, are contradictive rules if $fX_j = x \to y$ and $fX_k = x \to \neg y$, where $j,k = (1,...,|F(X)|)$ and $j \neq k$.

3.8 Conclusion

This chapter has presented essential concepts, definitions and problem definitions necessary for understanding the problems that will be approached in this thesis. The problem areas can be generally split into: association mining, feature subset selection, rules interestingness and validity. For each general area, the specific problem that will be addressed in this thesis was defined.

References

Agrawal, R., Imieliski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec., 22*, 207-216.

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc.

Agresti, A. (1996). *An introduction to categorical data analysis*. New York: Wiley.

Bayardo, R. J. (1998). Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*. Seattle, Washington, United States: ACM.

Bayardo, R. J., Agrawal, R., & Gunopulos, D. (2000). Constraint-Based Rule Mining in Large, Dense Databases. *Data Mining and Knowledge Discovery, 4*, 217-240.

Borowski, E. J., & Borwein, J. M. (1989). *Dictionary of mathematics*. London: Collins.

Burdick, D., Calimlim, M., Flannick, J., Gehrke, J., & Yiu, T. (2005). MAFIA: a maximal frequent itemset algorithm. *IEEE Transactions on Knowledge and Data Engineering, 17*, 1490-1504.

Cheng, H., Yan, X., Han, J., & Hsu, C.-W. (2007). Discriminative frequent pattern analysis for effective classification. In *23rd International Conference on Data Enginering (ICDE 2007)* (pp. 716-725). Pisctaway, NJ, USA: IEEE.

Chi, Y., Muntz, R. R., Nijssen, S., & Kok, J. N. (2005). Frequent Subtree Mining - An Overview. *Fundamenta Informaticae, 66*, 161-198.

Cochran, W. G. (1963). *Sampling techniques* (2d ed.). New York: Wiley.

Ehrenberg, A. S. C. (1975). *Data reduction: analysing and interpreting statistical data*. London: Wiley.

Feng, L., Dillon, T., Weigand, H., & Chang, E. (2003). An XML-Enabled Association Rule Framework. In *Database and Expert Systems Applications* (pp. 88-97).

Frank, A., & Asuncion, A. (2010). {UCI} Machine Learning Repository. In: University of California, Irvine, School of Information and Computer Sciences.

Frawley, W. J., Piatetsky-Shapiro, G., & Matheus, C. J. (1992). Knowledge Discovery in Databases: An Overview. In *AI Magazine* (Vol. 13): AAAI.

Garriga, G. C., Kralj, P., & Lavrač, N. (2008). Closed Sets for Labeled Data. *Journal of Machine Learning Research, 9*, 559-580.

Gouda, K., & Zaki, M. J. (2001). Efficiently Mining Maximal Frequent Itemsets. In *First IEEE International Conference on Data Mining (ICDM'01)* (Vol. 0, pp. 163).

Hadzic, F. (2011). A Structure Preserving Flat Data Format Representation for Tree-Structured Data. In *2011 Workshop on Quality Issues, Measures of Interestingness and Evaluation of Data Mining Models (QIMIE'11)*. Shenzhen, China: Frontiers of Computer Science in China Journal, Springer.

Hadzic, F., Dillon, T. S., & Chang, E. (2008). Knowledge Analysis with Tree Patterns. In *Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences*: IEEE Computer Society.

Hadzic, F., Tan, H., & Dillon, T. S. (2011). *Mining of Data with Complex Structures* (Vol. 333): Springer-Verlag Berlin Heidelberg.

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

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: concepts and techniques* (3rd ed.). Waltham, Mass.: Elsevier/Morgan Kaufmann.

Hosmer, D. W., & Lemeshow, S. (1989). *Applied logistic regression*. New York: Wiley.

Kantardzic, M. (2002). *Data mining: concepts, models, methods and algorithms*. New York: Wiley.

Keller, G., & Warrack, B. (2003). *Statistics for management and economics* (6th ed.). Pacific Grove, CA: Thomson/Brooks/Cole.

Lenca, P., Meyer, P., Vaillant, B., & Lallich, S. (2008). On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. *European Journal of Operational Research, 184*, 610-626.

Padmanabhan, B., & Tuzhilin, A. (1998). A Belief-Driven Method for Discovering Unexpected Patterns In *4th International Conference on Knowledge Discovery and Data Mining (KDD-98)* (pp. 94-100).

Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. (1999). Discovering Frequent Closed Itemsets for Association Rules. In C. Beeri & P. Buneman (Eds.), *Database Theory — ICDT'99* (Vol. 1540, pp. 398-416): Springer Berlin / Heidelberg.

Pei, J., Han, J., & Mao, R. (2000). CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets. In *ACM {SIGMOD} Workshop on Research Issues in Data Mining and Knowledge Discovery* (pp. 21-30).

Refaat, M. (2007). *Data preparation for data mining using SAS*. San Francisco: Morgan Kaufmann Publishers.

Roiger, R., & Geatz, M. (2003). *Data mining: a tutorial-based primer*. Boston: Addison Wesley.

Shaharanee, I. N. M., Hadzic, F., & Dillon, T. S. (2011). Interestingness measures for association rules based on statistical validity. *Knowledge-Based Systems, 24*, 386-392.

Shaharanee, I. .N M., Hadzic, F., & Dillon, T. (2009). Interestingness of Association Rules Using Symmetrical Tau and Logistic Regression. In A. Nicholson & X. Li (Eds.), *AI 2009* (Vol. 5866, pp. 442-431): LNAI.

Suciu, D. (1998). Semistructured Data and XML. In K. Tanaka & S. Ghandeharizadeh (Eds.), *The 5th International Conference on Foundations of Data Organization (FODO'98)* (pp. 1-12). Kobe, Japan.

Tan, H., Dillon, T., Hadzic, F., Chang, E., & Feng, L. (2005). MB3-Miner: efficiently mining eMBedded subTREEs using Tree Model Guided candidate generation. In *Proceedings of the 1st International Workshop on Mining Complex Data 2005 (MCD 2005)*. Houston, Texas, USA: IEEE Computer Society Press.

Tan, H., Dillon, T., Hadzic, F., Chang, E., & Feng, L. (2006). IMB3-Miner: Mining Induced/Embedded Subtrees by Constraining the Level of Embedding. In *Advances in Knowledge Discovery and Data Mining* (pp. 450-461).

Tan, H., Hadzic, F., Dillon, T. S., & Chang, E. (2008). State of the art of data mining of tree structured information. *International Journal of Computer Systems Science and Engineering, 23*.

Vossen, G. (1991). *Data models, database languages and database management systems*. Wokingham, England: Addison-Wesley Publishing Co.

Webb, G. I. (2007). Discovering Significant Patterns. *Machine Learning, Springer*, 1-33.

Wei, J.-M., Yi, W.-G., & Wang, M.-Y. (2006). Novel measurement for mining effective association rules. *Knowledge-Based Systems, 19*, 739-743.

Witten, I. H., & Frank, E. (2005). *Data mining: practical machine learning tools and techniques* (2nd ed.). Amsterdam ; Boston, MA: Morgan Kaufman.

Yahia, S. B., Hamrouni, T., & Nguifo, E. M. (2006). Frequent closed itemset based algorithms: a thorough structural and analytical survey. *SIGKDD Explor. Newsl., 8*, 93-104.

Zaki, M. J. (2000). Generating non-redundant association rules. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Boston, Massachusetts, United States: ACM.

Zaki, M. J. (2005). Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications. *IEEE Transactions on Knowledge and Data Engineering, 17*, 1021-1035.

Zaki, M. J., & Hsiao, C.-J. (2002). CHARM: An Efficient Algorithm for Closed Itemset Mining. In *2nd SIAM International Conference in Data Mining*.

Zhang, C., & Zhang, S. (2001). Collecting Quality Data for Database Mining. In *AI 2001: Advances in Artificial Intelligence* (pp. 131-142).

Zhang, J., Ling, T. W., Bruckner, R. M., Tjoa, A. M., & Liu, H. (2004). On Efficient and Effective Association Rule Mining from XML Data. In *Database and Expert Systems Applications* (pp. 497-507).

Zhou, X. J., & Dillon, T. S. (1991). A statistical-heuristic feature selection criterion for decision tree induction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*, 834-841.

CHAPTER 4 OVERVIEW OF THE PROPOSED FRAMEWORK

4.1 Introduction

This chapter provides a high level overview of the framework, details of which will be provided in the subsequent chapters. Generally, as clarified in Chapter 3, the work in this thesis is focused on evaluating association rules generated from both relational and tree-structured data. The aims are to investigate how data mining and statistical measurement techniques can be utilized in combination and to develop a method detailing a proper sequence of use of these techniques. Achieving these aims will ensure a more reliable and interesting set of rules. In Section 4.2, the overview of the proposed framework to tackle the problem of evaluating association rules from relational data is given. The extension of the framework in order to evaluate association rules from tree-structured data is discussed in Section 4.3. Finally, a summary of the chapter is provided in Section 4.4.

4.2 Conceptual Model and Framework for Relational Data Problems

With respect to the association rule mining from relational data, the first part of the work in this thesis is focused on the problems of evaluating the frequent itemsets, which, as previously discussed, is the essential problem caused by the large volume of discovered rules and patterns.

Figure 4.1 shows Framework *A*. Firstly, any necessary pre-processing is applied to the selected data, to ensure clean and consistent data. The dataset is then divided into two subsets. The first subset is used for the feature subset selection task, frequent itemsets generation, statistical analysis, redundancy and contradictive assessment, and rules filtering based on a confidence threshold. The second subset is treated as "unseen data" and is used for testing the final optimized rule set. This testing dataset acts as sample data used to verify the accuracy and coverage rate of the discovered rules. The relevance of the input attributes in predicting the class attributes is calculated using the Symmetrical Tau (ST) technique (Zhou & Dillon, 1991). The rules are then generated using three frequent itemset mining algorithms, as another aim is to investigate the difference in utilizing the different patterns for the classification task. The discovered rules are then evaluated using statistical analysis.

Furthermore, redundancy and contradictive assessment methods are employed to discard redundant and contradictive rules.

As previously discussed in Chapter 3, this thesis focuses on establishing a systematic way to evaluate association rules, in which the feature subset selection method, statistical analysis, redundancy and contradictive assessment methods, and confidence-based filtering can be effectively utilized. The thesis also provides a framework for defining the sequence of the use of these techniques. Within this framework, discarding the attributes using feature subset selection application before generating the rules will ensure that only relevant attributes and attributes capable of predicting the class variable are preserved in the dataset. Moreover, this will reduce the size of the dataset, which consequently reduces the number of rules to be accessed. Otherwise, the association rules mining process will end up with too many rules which then would need to be post-pruned once it was found that they include irrelevant attributes. This task is important as it will remove a number of random associations that may be generated based on the irrelevant attributes. By reducing the number of rules which arise from random associations, it makes the later statistical analysis, redundancy and contradictive assessment methods, and confidence-based filtering task more manageable.

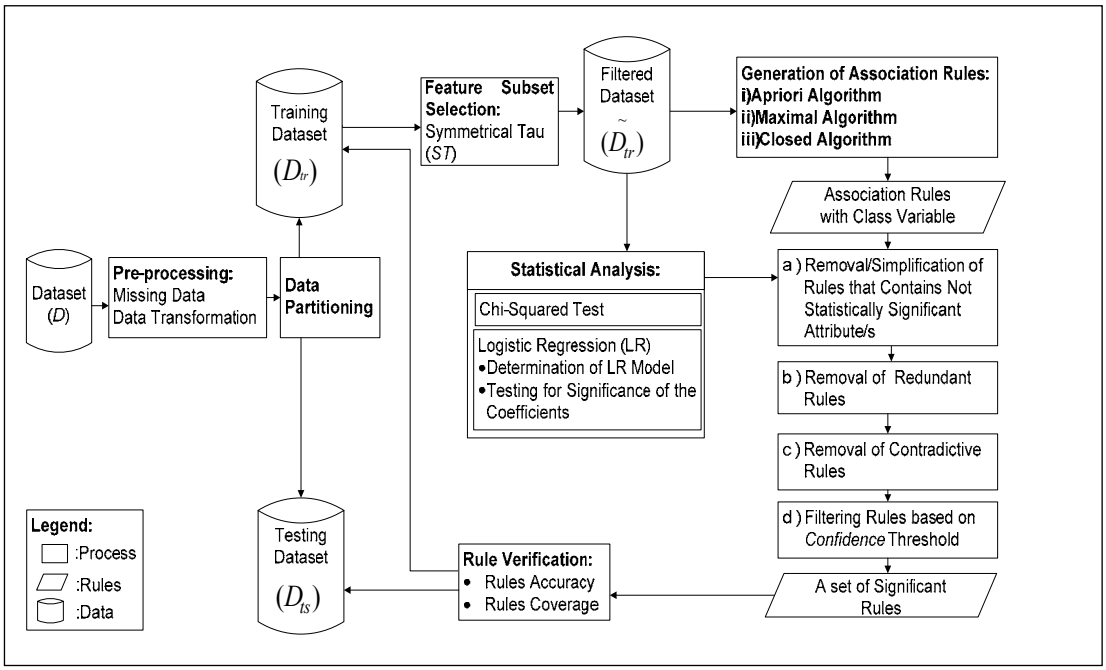


Figure 4.1: Framework *A* for Relational Data for Rules Interestingness Analysis

4.2.1 Pre-processing

Pre-processing is an important task in data mining to ensure only correct, clean and consistent data are mustered into the data mining process. Here, the pre-processing is applied to each attribute/s in the training dataset in order to obtain clean and consistent data. The pre-processing technique includes the removal of missing values and discretization of attributes with continuous values.

4.2.1.1 Missing Data and Data Transformation

Loss of information may occur when data is missing in data mining analysis. A missing value may indicate that either the data item exists but is unaccounted for, or it may contain no values at all. Data transformation is another important step in data pre-processing techniques. A common data transformation technique is to change the continuous values into specific ranges or classes, often known as the 'binning methods'. The binning method smoothes a sorted data value by consulting all the values around it (Han & Kamber, 2001). A detailed explanation of data transformation and missing data handling techniques used in the framework will be given in the next chapter.

4.2.1.2 Data Partition

The partitioning of data into the training dataset and the testing dataset plays an important role in evaluating the data mining model. The training dataset is often larger than the testing dataset. Data selected for each set are randomly selected, giving each instance an equal chance of being included in either the training or the testing dataset. The training dataset is used for frequent itemset generation, statistical analysis, and redundancy and contradictive assessments; while the testing dataset comprises sample data not previously considered during the rule determination stage and is used to verify the accuracy and coverage rate of the discovered rules.

4.2.2 Symmetrical Tau for Feature Subset Selection

Typically, a statistical-heuristic measure can be utilized to determine the relevance of input attributes by determining their importance in predicting the class label in the

training dataset. Consequently, any irrelevant attributes are removed from the dataset.

4.2.2.1 Asymmetrical Tau

Goodman and Kruskal proposed their measure of association, namely the Asymmetrical Tau, for cross-classification tasks in the statistical area (Sestito & Dillon, 1994). The Asymmetrical Tau is a measure of the relative usefulness of one variable in improving the ability to predict the classifications of members of the population with respect to a second variable (Goodman & Kruskal, 1954).

The category of variable *B* can be predicted from the category variable *A* by assuming *B* is statistically independent of *A* or assuming *B* is a function *A*. Thus, the degree of association is defined as the relative improvement in predicting the *B* category obtained when the *A* category is known, as opposed to when the *A* category is not known (Sestito & Dillon, 1994). The information contained in a contingency table is used. Typically a contingency table classifies a number of samples according to two criteria, i.e. it provides a two-way classification. If one criterion has *I* values and the other has *J*, then an *I* * *J* contingency table is created.
Let:

- there be *I* rows and *J* columns in a contingency table, for two attributes *A* and *B,* respectively*;*
- P(*ij*) denotes the probability that an individual belongs to both row category *I* and column category *j*;
- P (*i*+) and P (*j*+) the marginal probability in row category *I* and column category *j*, respectively.

The Asymmetrical Tau measure for predicting the class of attribute *B* from attribute *A* is defined as (Zhou & Dillon, 1991):

$$Tau \quad A|B = \frac{\sum_{j=1}^{J} \sum_{i=1}^{I} \frac{P(ij)^2}{P(+j)} - \sum_{i=1}^{I} P(i+)^2}{1 - \sum_{i=1}^{I} P(i+)^2}$$

The Asymmetrical Tau measure for predicting the class of attribute *A* from attribute *B* is defined as (Zhou & Dillon, 1991):

$$Tau \quad B|A = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\frac{P(ij)^2}{P(i+)} - \sum_{j=1}^{J}P(+j)^2}{1 - \sum_{j=1}^{J}P(+j)^2}$$

However, as proven by (Zhou & Dillon, 1991), this measure is less impressive when utilized for feature selection problems for decision trees, as it tends to favor features with more values. Hence, they proposed a new approach which combines two asymmetrical measures in order to obtain a balanced feature selection criterion. This measure will be discussed in the next section.

4.2.2.2 Symmetrical Tau

The Symmetrical Tau feature selection technique will be utilized in the proposed approach to ascertain the relative usefulness of attributes in predicting the value of the class attribute, and discard any of the attributes whose relevance value is fairly low. This would prevent the generation of rules which then would need to be discarded anyway once it was found that they include irrelevant attributes.

In this thesis, Symmetrical Tau (Zhou & Dillon, 1991) is utilized for feature subset selection purposes. The Symmetrical Tau (Zhou & Dillon, 1991) is a statistical-heuristic feature selection criterion. It measures the capability of an attribute to predict the class of another attribute. Let there be *R* rows and *C* columns in the contingency table for two attributes x and *y*. The probability that an individual belongs to row category *r* and column category *c* is represented as $P(rc)$, and $P(r+)$ and $P(+c)$ are the marginal probabilities in row category *r* and column category *c* respectively. The measure is based on the probabilities of one attribute value occurring together with the value of the second attribute. In this sense, the *y* attribute can be seen as a representative of the class attribute, and the Symmetrical Tau measure for the capability of input attribute in predicting the class attribute is defined as (Zhou & Dillon, 1991).

$$\tau(x,y) = \frac{\sum_{c=1}^{C}\sum_{r=1}^{R}\frac{P(rc)^2}{P(+c)} + \sum_{r=1}^{R}\sum_{c=1}^{C}\frac{P(rc)^2}{P(r+)} - \sum_{r=1}^{R}P(r+)^2 - \sum_{c=1}^{C}P(+c)^2}{2 - \sum_{r=1}^{R}P(r+)^2 - \sum_{c=1}^{C}P(+c)^2}$$

The higher values of the Symmetrical Tau measure would indicate better discriminating criteria (features) for the class that is to be predicted in the domain. The Symmetrical Tau has many more desirable properties in comparison with other feature subset selection techniques as reported by (Zhou & Dillon, 1991). It is capable of handling noise with the built-in statistical tools; dynamic error estimation conveys potential uncertainties in classification; it handles multi-valued attributes fairly; it is not proportional to the sample size; its proportional-reduction-in-error nature allows for an overall measure of a particular attribute's sequential variation in predictive ability, thereby determining which attributes have become less useful for prediction and should be deleted; and, it is capable of handling a Boolean combination of logical features.

In this thesis, the aim is to utilize the Symmetrical Tau criterion feature to address the feature subset selection problem. This will act as a filtering tool whereby irrelevant attributes are detected and removed prior to the association rules mining process. The process involves the ranking of attributes based on the decreasing value of Symmetrical Tau and a cut-off point is established below which all attributes are considered irrelevant and are removed from the dataset prior to mining for association rule generation. A further detailed discussion of its use will be provided in Chapter 5.

4.2.3 Rules Generation

Rules generation is mainly concerned with the discovery of rules using the Apriori, Maximal and Closed association rule mining approaches. Here, an overview of Apriori, Maximal and Closed rules is given. In this thesis, the main focus is on investigating the usefulness of patterns from *closed* sets compared with *maximal* sets and frequent itemsets for classification tasks, with respect to their classification accuracy, generalization capability and coverage rate, and the incorporation of interestingness measures in such patterns.

4.2.3.1 Apriori Algorithm

Association rule discovery finds all rules that satisfy specific constraints such as the minimum *support* and *confidence* threshold, as is the case with the Apriori algorithm (Agrawal, Imieliski, & Swami, 1993). It consists of two main phases: frequent itemsets discovery and association rule generation, of which the former task is more complex. The Apriori-based algorithm has been useful for frequent itemsets generation as it performs well on sparse data in discovering frequent patterns that are comprised of rather smaller itemsets. As mentioned earlier, the generation of frequent rules can be constrained by *support* and *confidence* threshold values. Consequently, in this thesis, two variants of frequent itemset discovery will be utilized: the first variant is constrained by both minimum *support* and *confidence* values and the second variant is constrained by only minimum *support* value. The reason behind the development of two Apriori variants is to investigate the implication of using the *confidence* measure as a constraint at different phases of the rule filtering process.

4.2.3.2 Maximal and Closed Algorithms

With the emergence of dense data which often results in frequent patterns containing larger itemsets, the performances of the Apriori algorithm tends to degrade as explained in (Zaki & Hsiao, 2002). Maximal and Closed algorithms are known for their capabilities in reducing the number of frequent itemset candidates that need to be enumerated. Even with a reduction of the complexity of rules, no information is lost since the complete set of frequent items can be obtained from both closed and frequent itemsets, and closed frequency values can also be worked out. In this study, the focus is on evaluating the usefulness of maximal/closed itemsets for the task of classification and for their classification/predictive accuracy and coverage rate.

4.2.4 Rule Evaluation

In this thesis, interesting rules are considered to be those rules that have a sound statistical basis and are neither redundant nor contradictive. Such an approach requires additional measures based on statistical independence and correlation analysis techniques to verify and evaluate the usefulness and quality of the rules

discovered. This will filter out the redundant, misleading, random and coincidentally-occurring rules, while at the same time sustaining the accuracy of the rule set and retaining valuable rules. Here, an overview of each measure involved is given, while the details pertaining to the utilization of the selected measures will be provided in Chapter 5.

4.2.4.1 Chi-squared Test for Correlation

When using the association rule mining method, rules discovered from Apriori, Maximal and Closed approaches may be very large and despite having constraint parameters such as support and confidence, they might still be misleading. Thus, as demonstrated by (Han & Kamber, 2001), correlation measures can be used to address these issues. Here, the chi-squared χ^2 statistic value will be applied to determine if the correlation between items is statistically significant. The chi-squared test was proposed in 1900 by Karl Pearson (Agresti, 1996) and is used for hypothesis testing of independence. The chi-square value is calculated as:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where; E_i is the expected and O_i is the observed frequency.

The properties of a contingency table were described in Section 4.2.2.1, on which the chi-squared calculation is based. To compute the χ^2 value, the squared difference between the observed and expected value from *I* and *J* in the contingency table is divided by the expected values. The contingency table can now be used to test whether the two criteria are independent. A low value means that the observed value and expected frequencies are in close agreement; while a high value means that there is greater discrepancy between the observed and expected frequencies (Chapter 5 provides a detailed explanation of the chi-squared test for correlation).

Within this thesis, the chi-squared test will be used to discover the properties of data attributes, principally in terms of data dependency. The inclusion of attributes that failed the chi-squared test in the association rules may indicate misleading, irrelevant

and insignificant rules since the attributes are considered as redundant, replicated and highly dependent on each other (Han & Kamber, 2001).

4.2.4.2 Logistic Regression analysis for Classification

Another statistical analysis that will be employed in this framework is logistic regression analysis. Generally, as described in (Roiger & Geatz, 2003), statistical regression is a supervised learning technique that generalizes a set of numeric data by creating a mathematical equation which relates one or more input attributes to a single numeric output attribute. However, as the focus of this thesis is limited to association rules with certain class attributes, a prominent regression approach, namely logistic regression, will be developed. Logistic regression as defined by (Roiger & Geatz, 2003) is a nonlinear regression technique that associates a conditional probability score with each data instance. Logistic regression is used to estimate the probability that a particular outcome will occur. The dependent variable in logistic regression is the odd ratio, while the outcome variable is binary or dichotomous. In some cases, the class variable may take on two or more possible values (Hosmer & Lemeshow, 1989).

In this thesis, logistic regression analysis will be utilized in the framework to describe the relationship between the target/class variable and the set of input variables (often in statistical terms recognized as covariates) (Hosmer & Lemeshow, 1989). Logistic regression involves;

(i) Determination of the logistic regression model by fitting the data; and

(ii) Testing for significance of the coefficients.

Equation 1 defines the logistic regression model. (A detailed discussion of the logistic regression model with reference to a dataset will be provided in the next chapter).

- **Logistic Regression Model:**

$$\ln(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \varepsilon \,, \qquad (1)$$

where;

y = Natural logarithm of the odds ratio,

$\beta_0, \beta_1, ..., \beta_k$ = Coefficients of the input attributes,

ε = Error variable,

Y = Dichotomous class attribute,

$x_1, x_2 ..., x_k$ = Input attributes

The coefficients are estimated using a statistical technique called **maximum likelihood estimation.** In this work, the SAS@ software will be used to develop the logistic regression models.

After estimating the coefficients and fitting the model, the next task is to conduct an assessment of the significance of the variables in the model. This will determine whether the input variables in the model are 'significantly' related to the class variable. Here, several models are developed. The best model will be selected based on the best fitting and most parsimonious and reasonable model to describe the relationship between the class variable and input variables.

To conclude, the application of logistic regression provides an additional measure of the classification power between the input and target attributes. The logistic regression models will be developed and fitted with the data; and the significance of the coefficients will be tested. This will ensure that irrelevant, random and insignificant attributes that failed the statistical analysis are removed. Association rules that consist of any insignificant input variables based on the logistic regression model are discarded and removed.

4.2.4.3 *Productive* Rules for Redundancy Removal

Redundancy removal plays an important role in this framework as a means of reducing the number of redundant rules produced by association rule mining algorithms. As described earlier in Chapter 3, the term *productive* as proposed by (Webb, 2007) will be utilized in performing the redundancy removal.

With the application of *productive* rules, the rules *improvement* values are calculated. If the values are less than or equal to zero, the rules are considered redundant and will be removed from further analysis. Additionally, a *productive* rule is capable of

identifying items that have been included in the antecedent and are actually independent of the consequence (Webb, 2007).

4.2.4.4 Contradictive Rule Removal

While redundancy removal offers a means of reducing the number of rules, another essential task involved in this framework is the contradictive rule removal. Contradictive rules as defined in Chapter 3 refer to rules that contradict each other. Contradictive rules and redundant rules may occur because of an initial low setting of the minimum support thresholds.

The purpose of the contradictive rule removal is to identify and to remove any occurrence of two or more rules that have the same pre-condition (i.e. antecedents) and imply different class values (consequents). As will be demonstrated in later experiments, the accuracy rate for rule sets without the contradictive rules is relatively higher compared with that of rule sets contaminated by contradictive rules.

4.2.4.5 Rules Accuracy and Rules Coverage

When a large volume of rules is removed using the above statistical analysis, and redundancy and contradictive assessment methods, another crucial issue arises: whether the quality of the rules obtained from the proposed framework has been compromised. Here, the quality of rules is demonstrated in terms of their accuracy and coverage values. Earlier in Chapter 3, the definition of the problem of rules accuracy and rules coverage was outlined.

Within this task, the values for rule accuracy and coverage will be measured at every stage and sequence involved. This measure is crucial as this can determine the quality of the discovered rules. Additionally, this analysis also reveals the balancing/optimization issues with regards to the trade-off between accuracy rate and coverage rate. Consequently, the task of choosing optimal stopping criteria based on a *minimum confidence* threshold will be discussed and presented to ensure that an acceptable level of accuracy and coverage rate can be achieved. A detailed analysis

and explanation of the quality of rules and optimization issues will be presented in Chapter 5 and Chapter 6.

4.3 Conceptual Model and Framework for Tree-structured Data Problems

Refer to the problem to be addressed in Chapter 3 in evaluating the interestingness of association rules from tree-structured data. The frequent subtrees generated can be based on a standard frequent subtree mining algorithm or on frequent subtrees obtained from a structure-preserving flat format for tree-structured data. Therefore, two experimental setting are provided as follows; in the first experimental setting, a standard frequent subtree mining algorithm is used to discover the frequent subtrees (i.e. IMB3-Miner (Tan, Dillon, Hadzic, Chang, & Feng, 2006)). Hence, the evaluation process is very similar to the Framework *A* as shown in Figure 4.1 except that the association rules are based on frequent subtrees rather than on the frequent itemsets mining algorithm. Therefore, it is not necessary to replicate the steps involved here, and the focus is on the conceptual model for the second experimental setting.

In the second experimental setting, Framework *B* as depicted in Figure 4.2 is used to evaluate the interestingness of subtree patterns generated from the structure-preserving flat format for tree-structured data. The overview of the process of Framework *B* is as follows: While the majority of the tasks involved in Framework *B* is similar to those in Framework *A*, the major difference is the application of a *Database Structure Model* (*DSM*) (Hadzic, 2011) to obtain a structure-preserving flat data format (*FDT*) for tree-structured data (shown in Figure 4.2 with the square dash line region).

The *DSM* is extracted from the tree-structured data to preserve the structural characteristics of the data. The extracted *DSM* is used to create the flat representation of the tree structured data. An example of the conversion process will be given in Chapter 7. Once the tree-structured data has been converted to a flat table format (*FDT*), the exact sequence of feature subset selection, frequent pattern mining and rule filtering process as given and justified in Section 4.2 (Framework *A*) for relational data, is also applied to tree-structured data in Framework *B*. The

association rule mining algorithm is utilized to discover frequent rules from the *FDT*. The extracted frequent rules are mapped onto the *DSM* to re-generate the pre-order string encoding of subtrees, thereby representing them as subtrees of the tree database.

These frequent rules may contain both valid and invalid rules (disconnected subtree) identified as *FullTree*. In addition to that, the rules based on embedded subtrees and the rules based on induced subtrees (the rule set that excludes invalid/disconnected subtrees) have also been revealed within the extracted frequent rules. These three frequent rule sets are then evaluated using statistical analysis, and the redundancy and contradictive assessment methods as described in Framework *A*. The combination of these rule evaluation strategies will help to determine the accurate and high quality rules.

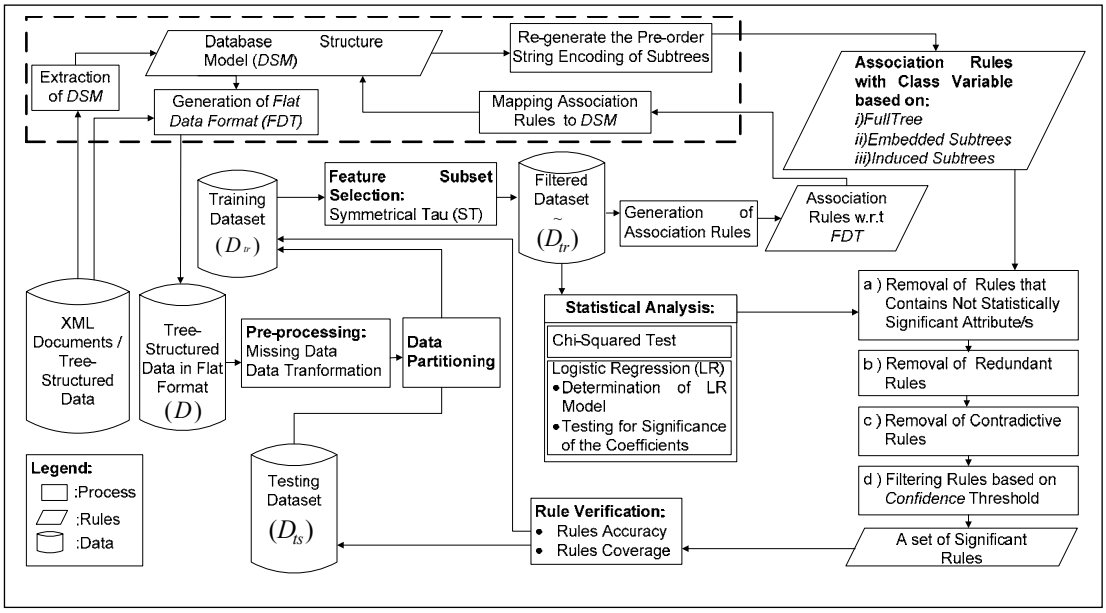


Figure 4.2: Framework *B* for Tree-structured Data for Rules Interestingness Analysis

4.3.1 Modeling XML and Tree-structured Data

As stated earlier in Chapter 3, the focus of this thesis is restricted to the evaluation of the association rules discovered from XML documents and tree-structured data, thus modeling issues on how to represent the XML document into tree format that later can be used for frequent subtree mining, was discussed in Chapter 3. These include

the definition of tree-related concepts, and definitions of and the parallelism between XML and tree-structured data. XML documents are commonly modeled as rooted ordered labeled trees (Hadzic, Tan, & Dillon, 2011). In the field of frequent subtree mining, the pre-ordered string encoding (Zaki, 2005) has become a well accepted format when modeling tree-structured data. A specific example of modeling tree-structured data will be given in Chapter 7.

4.3.2 Frequent Subtrees Generation

Generally, the problem of frequent subtree mining can be stated as the task of finding all subtrees that occur in a tree database at least as many times as the user-specified minimum support threshold as formally defined in Chapter 3. There has been great development in frequent subtree mining algorithms as reported in (Chi, Muntz, Nijssen, & Kok, 2005) with each algorithm being tailored for a specific application.

However, similar to traditional association rule mining, the frequent subtree patterns may be unable to recognize relevant and interesting patterns due to the huge volume of patterns generated. Thus, as mentioned in Section 4.1, with respect to the problem of frequent subtree mining, in this work the aim is to develop a proper sequence of use of statistical techniques, together with redundancy and contradictive assessment methods to arrive at a more reliable and interesting set of subtree patterns. The development/refinement of frequent subtree mining algorithms is outside the scope of this thesis.

4.3.3 Tree-structured Data Format Conversion

For given tree-structured data, the enumeration of all possible subtrees in a complete, non-redundant and efficient way is the major problem one needs to tackle (Tan, Hadzic, Dillon, Chang, & Feng, 2008). At a lower support threshold, one may experience a significant delay in the subtree patterns analysis and interpretation process. Additionally, as a large number of frequent subtree patterns may be discovered, many of which may not be useful, one needs to filter out many of the irrelevant/uninteresting patterns.

The flat data format (relational or vectorial data) was proven to be acceptable and successful when utilized with many well-established data mining techniques. Thus, an effective way proposed by (Hadzic, 2011) knows as *Database Structure Model* (*DSM*) is utilized in this thesis to represent tree-structured data in a structure-preserving flat data format. This approach offers a way of preserving tree-structured and attribute-value information. With the application of *DSM*, the structural characteristics are preserved during the data mining process. The extracted rules from the data mining application can be mapped onto the *DSM* to re-generate the pre-order string encoding of subtrees. This conversion tool created the means and opportunity for analyzing tree-structured data which will broaden the current data mining/analysis techniques (Hadzic, Hacker, & Tagarelli, 2011; Hadzic & Hecker, 2011). Details of the conversion process will be discussed in Chapter 7.

4.3.4 Subtrees Evaluation

IMB3-Miner (Tan et al., 2006) will be used to generate frequent subtree patterns (details of the experiment will be provided in Chapter 7) in the first experimental setting. The subtree patterns discovered by the IMB3 algorithm can aid in discovering potentially useful pattern structures in XML documents, which makes it useful and handy in discovering interesting similarities and differences. However, with the more complex data used in the later experiments, the evaluation process became infeasible, the applicability of the proposed framework to evaluate the frequent subtrees from the traditional frequent-subtree-mining-based approach deteriorated. Thus in the later experiments, the *DSM* approach is utilized in order to convert the tree-structured data to a flat data format.

As frequent subtrees have been discovered with either IMB3-Miner or Flat Data Format (*FDT*) based on the *DSM* approach, the question still remains whether these patterns have been discovered due to pure coincidence resulting from certain randomness in the particular dataset being analyzed. Furthermore, they are often quite large in number, which can be detrimental to the analysis procedures. Interesting rules, as defined in Section 4.2.4, are those rules that have a sound statistical basis and are neither redundant nor contradictive. Statistical analysis, and redundancy and contradictive-based assessment methods, will be utilized to verify

and evaluate the usefulness and quality of the rules discovered. Thus, similar methods and approaches discussed in Section 4.2.4 will be utilized to evaluate the frequent subtree patterns. This will filter out the redundant, misleading, random and coincidentally-occurring rules, while at the same time maintaining the accuracy of the rule set and retaining valuable rules.

4.3.4.1 Rules Accuracy and Rules Coverage

Referring to the similar problem of evaluating the interestingness of rules from relational data, an important issue that needs to be addressed is whether the quality of the rules obtained from the proposed framework have been compromised. A measure needs to be applied to verify whether the removal of a large volume of rules based on statistical analysis, and redundancy and contradictive assessment methods, will enable the discovery of all the interesting and significant subtree patterns.

As such, the quality of the subtree pattern will be demonstrated based on their accuracy and coverage values. The formal definition of subtree patterns accuracy and optimization for tree-structured data are discussed in Chapter 3. Additionally, experimental evaluation will be performed in Chapter 7 in demonstrating the detailed analysis and explanation of the subtree patterns' quality and optimization issues.

The values for rule accuracy and coverage will be measured at every stage and sequence of this task. This measure is crucial as it can determine the quality of the discovered rules. Additionally, this analysis will reveal the balancing/optimization issues with regards to the trade-off between accuracy rate and coverage rate.

4.4 Conclusion

This chapter has provided a brief overview of the way in which the problems in Chapter 3 will be addressed. Such discussions serve the purpose of outlining the direction which will be taken by the proposed solutions to the defined problems. The developed frameworks can be divided into two main categories evaluating interestingness of association rules from relational and tree-structured data.

The work in this thesis is focused on the development of a framework for evaluating the interestingness of frequent patterns. The approach to frequent pattern mining is further subdivided depending on whether relational data or tree-structured data is involved. Within the relational data, the focus is on evaluating the frequent pattern generated from Apriori, Maximal and Closed patterns. Extracting frequent patterns from tree-structured data is a moderately new research field. The initial frequent subtree patterns evaluated in this thesis are generated from the traditional frequent subtree mining algorithm, namely the IMB3-Miner, but in the later experiment, the *DSM* approach is utilized. Consequently, the rule evaluation processes are also employed for the frequent rules, rules based on embedded, induced and disconnected subtrees generated from flat data format (*FDT*) using the *DSM* approach.

With respect to the evaluation of both patterns from relational and tree-structured data, this chapter indicated the proposed solution to the problem of feature subset selection, statistical analysis, redundancy and contradictive assessment, and rules accuracy and coverage. The feature subset selection problem will be approached through the use of Symmetrical Tau (Zhou & Dillon, 1991) measure. The chi-squared test and logistic regression will be employed for statistical analysis. The redundancy and contradictive assessment methods will be utilized to remove redundant and contradictive rules. The quality of patterns will be measured according to their accuracy and coverage.

Within this framework, a proper sequence for the use of these techniques is developed so as to arrive at a more reliable and interesting set of rules generated from: (a) association rule algorithms, specifically the Apriori, Maximal and Closed approaches discovered from relational data, and (b) frequent subtrees using the IMB3-Miner algorithm and frequent subtrees based on the *DSM* approach discovered from tree-structured data.

References

Agrawal, R., Imieliski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec., 22*, 207-216.

Agresti, A. (1996). *An introduction to categorical data analysis*. New York: Wiley.

Chi, Y., Muntz, R. R., Nijssen, S., & Kok, J. N. (2005). Frequent Subtree Mining - An Overview. *Fundamenta Informaticae, 66*, 161-198.

Goodman, L., & Kruskal, W. (1954). Measures of Association for Cross Classifications. In *Journal of the American Statistical Association* (Vol. 49, pp. 732-764): American Statistical Association.

Hadzic, F. (2011). A Structure Preserving Flat Data Format Representation for Tree-Structured Data. In *2011 Workshop on Quality Issues, Measures of Interestingness and Evaluation of Data Mining Models (QIMIE'11)*. Shenzhen, China: Frontiers of Computer Science in China Journal, Springer.

Hadzic, F., Hacker, M., & Tagarelli, A. (2011). XML Document clustering using structure-preseving flat representation of XML content and structure. In *7th International Conference on Advanced Data Mining and Applications*. Beijing, China.

Hadzic, F., & Hecker, M. (2011). Alternative Approach to Tree-Structured Web Log Representation and Mining. In *EEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. Lyon, France.

Hadzic, F., Tan, H., & Dillon, T. S. (2011). *Mining of Data with Complex Structures* (Vol. 333): Springer-Verlag Berlin Heidelberg.

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

Hosmer, D. W., & Lemeshow, S. (1989). *Applied logistic regression*. New York: Wiley.

Roiger, R., & Geatz, M. (2003). *Data mining: a tutorial-based primer*. Boston: Addison Wesley.

Sestito, S., & Dillon, T. S. (1994). *Automated knowledge acquisition*. New York: Prentice Hall.

Tan, H., Dillon, T., Hadzic, F., Chang, E., & Feng, L. (2006). IMB3-Miner: Mining Induced/Embedded Subtrees by Constraining the Level of Embedding. In *Advances in Knowledge Discovery and Data Mining* (pp. 450-461).

Tan, H., Hadzic, F., Dillon, T. S., Chang, E., & Feng, L. (2008). Tree model guided candidate generation for mining frequent subtrees from XML documents. *ACM Trans. Knowl. Discov. Data, 2*, 1-43.

Webb, G. I. (2007). Discovering Significant Patterns. *Machine Learning, Springer*, 1-33.

Zaki, M. J. (2005). Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications. *IEEE Transactions on Knowledge and Data Engineering, 17*, 1021-1035.

Zaki, M. J., & Hsiao, C.-J. (2002). CHARM: An Efficient Algorithm for Closed Itemset Mining. In *2nd SIAM International Conference in Data Mining*.

Zhou, X. J., & Dillon, T. S. (1991). A statistical-heuristic feature selection criterion for decision tree induction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*, 834-841.

CHAPTER 5: DETAILED SOLUTIONS TO VERIFY THE ASSOCIATION RULE FROM RELATIONAL DATA

5.1 Introduction

This chapter describes the framework developed for evaluating association rules derived from relational data. The motivation behind our proposed method is to investigate how association rules mining, statistical analysis, redundancy and contradictive assessment methods can be utilized, and to develop a proper sequence of use of these techniques to arrive at a more reliable and interesting set of rules generated by association rule algorithms, specifically the Apriori, Maximal and Closed approaches.

One of the aims of the work is to investigate the implication of using different confidence values and the time at which the constraint is applied. The *confidence* measures play a significant role in limiting the number of rules from frequent itemsets mining. This may discard some of the rules that cover a smaller subset of data objects from the domain at hand. The rules covering a smaller subset of data may be necessary to detect contradictions in the formed associations and to discard those contradictive rules. Thus, to study these effects, two variants of frequent itemset discovery have been applied: the first variant is the Apriori framework that consists of both minimum *support* and minimum *confidence* thresholds, and the second variant is the Apriori framework with only a minimum *support* threshold (defined in Section 5.7.1) (where the confidence measure is applied at a later stage after contradictive rules have been removed).

Maximal and Closed frequent itemset mining is known to reduce the rule set size, but the question still remains whether, by doing so, their coverage rate, accuracy and *generalization power* has decreased. The majority of the Maximal and Closed works mentioned in Chapter 2 tend to focus more on the structural and analytical comparative study of the algorithms' performance in generating the rules. In this study, the focus is on evaluating the usefulness of maximal/closed approaches for the classification task and their generalization power and coverage rate. The aim is to study the differences in these aspects with respect to extracting frequent itemset/closed/maximal patterns.

The changes in *confidence* values have a direct impact on the size, accuracy and coverage rate of the rule set. The use of high *confidence* thresholds, typically results in a reduced number of rules with high accuracy but smaller coverage rate. On the other hand, with a low *confidence* threshold, a larger coverage rate is achieved but at the cost of a reduction in accuracy. In addition, smaller sets of rules are preferred as they typically have better generalization power. Thus, given this trade-off between the rule accuracy and rule coverage, the choice of an optimal confidence value is of great importance.

The relevant task in the proposed framework starts with the data pre-processing task, the determination of relevant attributes, the generation of frequent item rules, the rules interestingness and constraint measurement, and the rules' accuracy and coverage rate determination. The task is necessary as, in general, interesting rules can be interpreted as those rules that have a sound statistical basis and are neither redundant nor contradictive This statistics-based approach requires sampling process, hypothesis development, model building and finally evaluation of the usefulness and quality of the rules discovered. This will filter out the redundant, contradictive, misleading, random and coincidentally-occurring rules, while at the same time maintain the accuracy and coverage rate of the rule set. In the next section, a detailed explanation of the steps involved and the formal definition of the conceptual Framework *A* is provided.

5.2 Relational Data

The association rule mining, as explained in Chapter 3, is capable of discovering interesting relationships between items in a given dataset. Thus, the definition of the relational dataset format that is used in the framework is as follows:

Definition 1 Given a relational database D, $I = \{i_1, i_2, ..., i_{|D|}\}$ the set of distinct items in D, $AT = \{at_1, at_2, ..., at_{|AT|}\}$ the set of input attributes in D, and $Y = \{y_1, y_2, ..., y_{|Y|}\}$ the class attribute with a set of class labels in D. Assume that D contains a set of n records $D = \{x_r, y_r\}_{r=1}^{n}$, where $x_r \subseteq I$ is an item or a set of items and $y_r \in Y$ is a class

label, then $|x_r| = |AT|$ and $x_r = \{at_1val_r, at_2val_r, ..., at_{|AT|}|val_r\}$ contains the attribute names and corresponding values for record r in D for each attribute at in AT.

Here, the relational data is arranged in a row and column format. Each column is designated for attributes with their values, while the final column contains the class attributes with a set of possible class labels. Each row is reserved for the items and represents one record often referred to as an 'instance'. Figure 5.1 shows an example subset from the Wine dataset based on Definition 1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Alcohol	MalidAcid	Ash	Alcalinity	Magnesiur	Phenols	Flavanoids	NonFlavan	Proanthoc	Color	Hue	Diluted	Proline	Class
2	14.23	1.71	2.43	15.6	127	2.8	3.06	0.28	2.29	5.64	1.04	3.92	1065	1
3	13.2	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.4	1050	1
4	13.16	2.36	2.67	18.6	101	2.8	3.24	0.3	2.81	5.68	1.03	3.17	1185	1
5	14.37	1.95	2.5	16.8	113	3.85	3.49	0.24	2.18	7.8	0.86	3.45	1480	1
6	13.24	2.59	2.87	21	118	2.8	2.69	0.39	1.82	4.32	1.04	2.93	735	1
7	14.2	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.75	1.05	2.85	1450	1

Figure 5.1: Relational data example for (subset of) Wine dataset

5.3 Pre-processing

Pre-processing is utilized in this thesis to ensure that only appropriate data in suitable format is made available to the association rule mining process. In this framework, two problems arise with respect to the selected datasets. The first problem is the existence of missing data and the second problem is the need for data transformation from continuous data type into categorical data type. Thus, the general pre-processing steps are formalized as follows:

STEP 1: The pre-processing is applied to each at_i in D, where $at_i \in AT, (i = (1,...,|AT|))$ in order to obtain clean and consistent data. These pre-processing techniques include the removal of missing values and discretization of attributes with continuous values.

5.3.1 Missing Data Handler

In dealing with the missing data, the *delete selected cases or variables* and *data imputation* methods are utilized as proposed by Brown and Kros in (Wang, 2003). Thus in this thesis, the cases deletion method in SAS Enterprise Miner@ software is utilized. By using only a complete records/transaction in the database, this will ensure that the association rules mining process discovers rules that are based on the actual data rather than surmised data in the records/transactions (Refaat, 2007).

The second method of handling missing data is by data imputation (Lakshminarayan, Harp, & Samad, 1999). With this technique, the missing data regarding certain observations is estimated based on the valid values of other variables (Wang, 2003). Table 5.1 shows the specific approaches utilized in this thesis for handling missing data in the different relational datasets used. Note that for Wine and Iris, there were no missing values in the dataset.

Table 5.1: Missing data handling done for different datasets

Dataset	**Missing Data**	**Type of Missing Data Handler**
Wine	No	-
Adult	Yes	Delete Cases
Mushroom	Yes	Data imputation - Distribution based missing value approach
Iris	No	-

In the Adult dataset, 3620 records out of 48842 with unknown values are removed. While, for the Mushroom dataset, there are 2480 records with missing values denoted as "?". The distribution-based missing value method is utilized. In this approach, the replacement values are calculated based on the random percentiles of the distribution of variables (Refer to Table 5.2). This preserves the empirical distribution of the data.

Table 5.2: Missing data in Mushroom Dataset

Before Imputation

	SRoot
2453	
2454	bulbous
2455	
2456	
2457	
2458	
2459	
2460	
2461	
2462	
2463	bulbous

After Imputation

	SRoot
2453	bulbous
2454	bulbous
2455	bulbous
2456	bulbous
2457	equal
2458	bulbous
2459	club
2460	bulbous
2461	bulbous
2462	equal
2463	bulbous

5.3.2 Data Transformation

The measurement level of input variables for each of the datasets used for generating the association rules from relational data are as follows: all input attributes in 'Wine' and 'Iris' datasets are continuous type; all input attributes in 'Mushroom' dataset are categorical type; while in 'Adult' datasets, there is a mixture of continuous and categorical attributes. A continuous measurement level often provides an in-depth analysis; however, this may create a large number of distinct values per input attribute (Dillon, Hossain, Bloomer, & Witten, 1998; Han & Kamber, 2001). An equal width binning approach is utilized for each continuous attribute in the Wine, Adult and Iris datasets. There are different types of binning methods available for data transformation. However, these might require more pre-processing in order to understand the quality of the bins (Dillon et al., 1998) rather than to determine the interestingness of the rules. Since the interestingness of the rules is the primary focus of this thesis, we will utilize the equal width binning approach.

The data for each attribute are sorted and then smoothed by consulting its nearest boundary (Han & Kamber, 2001). Figure 5.2 demonstrated the 5 equal bins application on Alcohol attribute. The 5 equal bins are created by dividing the data values into 5 equally-spaced intervals based on the difference between the minimum and maximum values; thus, the number of records in each bin is typically unequal. Table 5.3 shows an example of the original continuous values (before binning) for attributes Alcohol with respect to their new bin (after binning). (i.e. Alcohol with

continuous values of 14.23 is now included into bin (14.07-high) representing the range of value between 14.07 and the maximum value in Alcohol attribute).

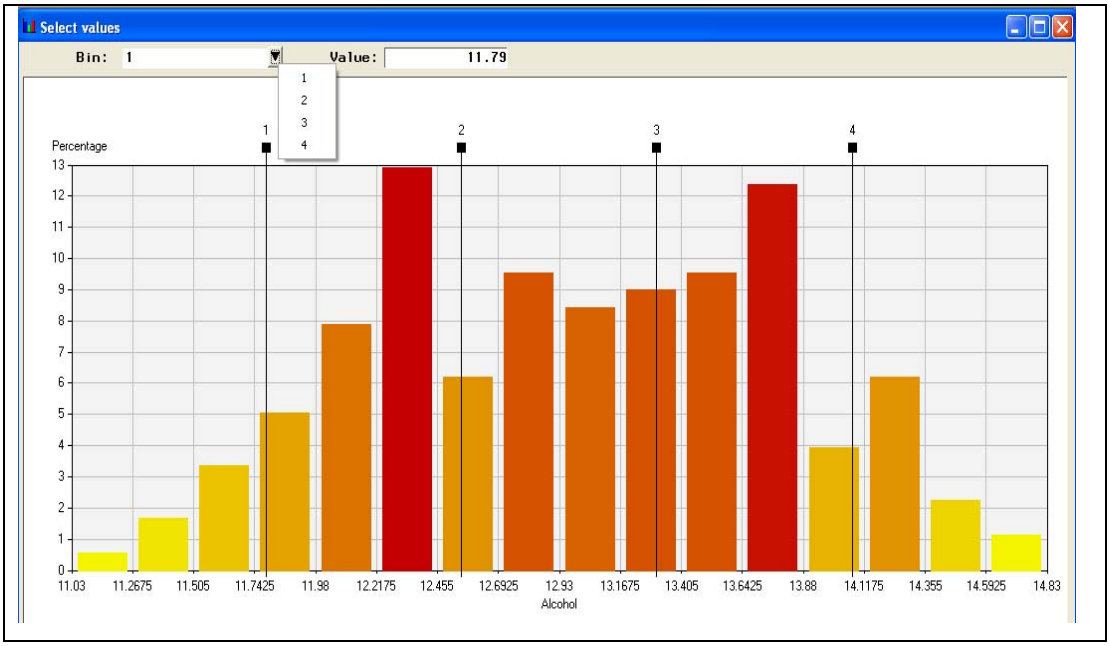


Figure 5.2: 5 Equal Bins for Alcohol attribute in Wine dataset

Table 5.3: Binning in Wine Dataset

Alcohol	
Before Binning	**After Binning**
14.23	(14.07-high)
11.03	(low-11.79)
13.16	(12.55 - 13.31)
12.16	(11.79 - 12.55)
13.67	(13.31-14.07)
14.20	(14.07-high)
14.39	(14.07-high)

5.4 Data Partition

Partitioning the dataset into training and testing sets allows one to generate a pattern from the training sample and validate it with a testing dataset. Records in the training dataset are selected using simple random sampling. This approach offers all data an equal chance of being included in the training sample. The SAS Enterprise Miner@ software is utilized to partition the data into the training and testing sets.

STEP 2: Data Partitioning. The training dataset is denoted as $D_{tr} \subseteq D$ and the testing dataset as $D_{ts} \subseteq D$.

5.5 Feature Subset Selection

The feature subset selection problem was discussed and defined in Chapter 3; moreover, the Symmetrical Tau features selection techniques were overviewed in Chapter 4. Here, the focus is on describing the feature subset selection process by applying the Symmetrical Tau and comparing its results with those obtained by the Mutual Information approaches. A detailed evaluation and comparison of both techniques will be presented in Chapter 6. Based on the proposed framework, this feature subset selection is needed in order to determine the relevance of attributes by classifying their importance to characterize an association. Both techniques are capable of measuring the capability of an input attribute in predicting the class of another attribute. This step is defined as:

STEP 3: Determine the relevance of each at_i by determining its importance in predicting the value of the class attribute Y in D_{tr}, where $at_i \in AT, (i = (1,...,|AT|))$ using a statistical-heuristic measure. Any irrelevant attributes are removed from the dataset, and are represented in the filtered database as $\tilde{D}_{tr}$, $\tilde{I} \subseteq I$.

5.5.1 Symmetrical Tau Utilization

Let there be *R* rows and *C* columns in the contingency table for attributes at_i and Y. The probability that an individual belongs to row category *r* and column category *c* is represented as $P(rc)$, and $P(r+)$ and $P(+c)$ are the marginal probabilities in row category *r* and column category *c* respectively. The measure is based on the probability of one attribute value occurring together with the value of the second attribute. In this sense, the Y attribute can be seen as a representative of the class attribute, and the Symmetrical Tau measure for the capability of input attribute at_i in predicting the class attribute Y is defined by (Zhou & Dillon, 1991) as follows:

$$\tau(at_i, Y) = \frac{\sum_{c=1}^{C}\sum_{r=1}^{R}\frac{P(rc)^2}{P(+c)} + \sum_{r=1}^{R}\sum_{c=1}^{C}\frac{P(rc)^2}{P(r+)} - \sum_{r=1}^{R}P(r+)^2 - \sum_{c=1}^{C}P(+c)^2}{2 - \sum_{r=1}^{R}P(r+)^2 - \sum_{c=1}^{C}P(+c)^2} \quad (1)$$

Higher values of the Symmetrical Tau measure would indicate better discriminating criteria (feature) for the class that is to be predicted in the domain. Symmetrical Tau has many more desirable properties in comparison to other feature subset selection techniques, as was reported in (Zhou & Dillon, 1991). It is utilized here to indicate the relative usefulness of attributes in predicting the value of the class attribute, and to discard any of the attributes whose relevance value is fairly low. This would prevent the generation of rules which then would need to be discarded anyway once it was found that they include irrelevant attributes.

5.5.2 Mutual Information

In this thesis, the capabilities of Symmetrical Tau as the determinant of the relevance of attributes are evaluated by comparing it with an information-theoretic measure, namely the Mutual Information. The definition of Mutual Information was given in Chapter 2.

The information-theoretic measures are principally comprehensible and useful since they can be interpreted in terms of information. For a rule interestingness measure, the relation is interesting when the antecedent provides a great deal of information about the consequent (Blanchard, Guillet, Gras, & Briand, 2005). Although several information-theoretic measures exist, the Symmetrical Tau is only compared with the Mutual Information measurement technique which is the most well-known of the techniques. The Mutual Information measure is calculated based on (Ke, Cheng, & Ng, 2008; Tan, Kumar, & Srivastava, 2002):

$$M(at_i, Y) = \frac{\sum_r \sum_c P(at_i, Y)\log\frac{P(at_i, Y)}{P(at_i)P(Y)}}{\min(-\sum_r P(at_i)\log P(at_i) - \sum_c P(Y)\log P(Y))} \quad (2)$$

The information that at_i gives us about Y is the reduction in uncertainty about Y due to knowledge of at_i and similarly for the information that Y tells about at_i (Ke, et al., 2008). The greater the values of *M,* the more information at_i and Y contain about each other (Ke et al., 2008).

5.6 Data Format for Frequent Item Mining

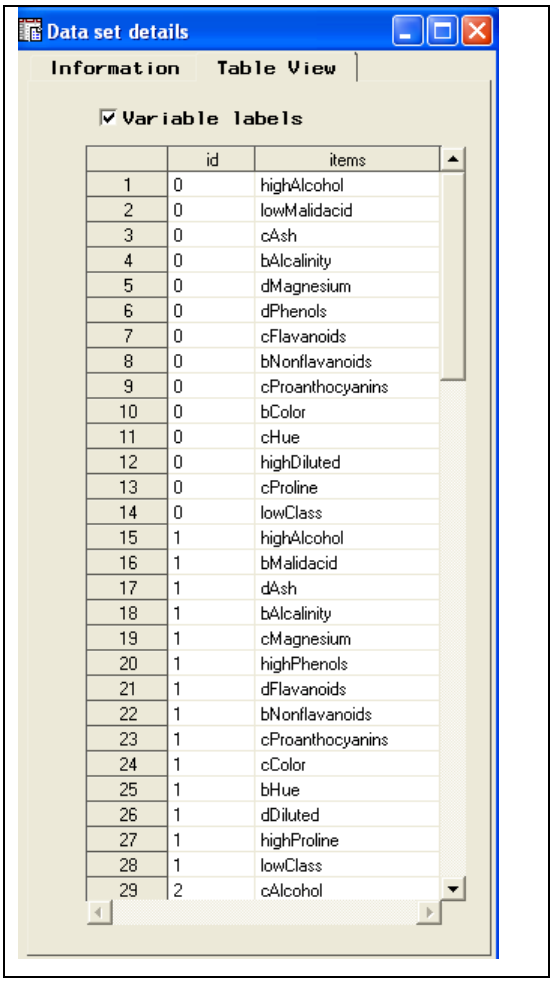
5.6.1 Data Format for Apriori Algorithm

The relational data format for the Wine dataset after the pre-processing (i.e. data transformation is displayed in Figure 5.3.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Alcohol	Malidacid	Ash	Alcalinity	Magnesium	Phenols	Flavanoids	Nonflavanoids	Proanthocyanins	Color	Hue	Diluted	Proline	Class
2	high	low	c	b	d	d	c	b	c	b	c	high	c	low
3	high	b	d	b	c	high	d	b	c	c	b	d	high	low
106	d	d	c	d	b	b	low	c	b	c	low	low	b	high
107	c	d	c	c	c	b	low	c	b	d	low	low	b	high
108	high	d	d	d	b	b	low	high	b	d	low	low	b	high

Figure 5.3: Example of data format for Wine dataset

Figure 5.4 presents the data format used by the Apriori algorithm to generate the frequent itemsets. On the left of the table the record identifier is shown, often referred to as the transaction ID (t_{id}). On the right of the table in Figure 5.4, attribute value-name (starting with capital) pair is shown respective to the t_{id} where it occurred. This data format for association rule analysis is commonly referred to as the *transactional* data.



	id	items
1	0	highAlcohol
2	0	lowMalidacid
3	0	cAsh
4	0	bAlcalinity
5	0	dMagnesium
6	0	dPhenols
7	0	cFlavanoids
8	0	bNonflavanoids
9	0	cProanthocyanins
10	0	bColor
11	0	cHue
12	0	highDiluted
13	0	cProline
14	0	lowClass
15	1	highAlcohol
16	1	bMalidacid
17	1	dAsh
18	1	bAlcalinity
19	1	cMagnesium
20	1	highPhenols
21	1	dFlavanoids
22	1	bNonflavanoids
23	1	cProanthocyanins
24	1	cColor
25	1	bHue
26	1	dDiluted
27	1	highProline
28	1	lowClass
29	2	cAlcohol

Figure 5.4: Data format (Wine) for Apriori algorithm

5.6.2 Data Format for the Maximal and the Closed Algorithms

The Charm (Zaki & Hsiao, 2002) and GenMax (Gouda & Zaki, 2001) algorithms employ the integer-based format for faster processing to generate the Closed and Maximal itemsets, respectively. As such, each attribute value-name pair is mapped to a unique integer. An example of an integer-string mapping table for a Wine dataset is shown in Table 5.4. Figure 5.5 illustrates the example of an integer-based format suited to the application of Charm and GenMax algorithms.

Table 5.4: An example of attribute value-name pair mapped to a unique integer for Wine dataset (for the first record)

attribute-value-name	integer
highAlcohol	4
lowMalidAcid	10
cAsh	11
bAlcalinity	16
dMagnesium	23
dPhenol	27
cFlavanoids	30
bNonflavanoids	35
cProanthocyanins	41
bColor	45
cHue	51
highDiluted	58
cProline	61
lowClass	66
…	…
N	…

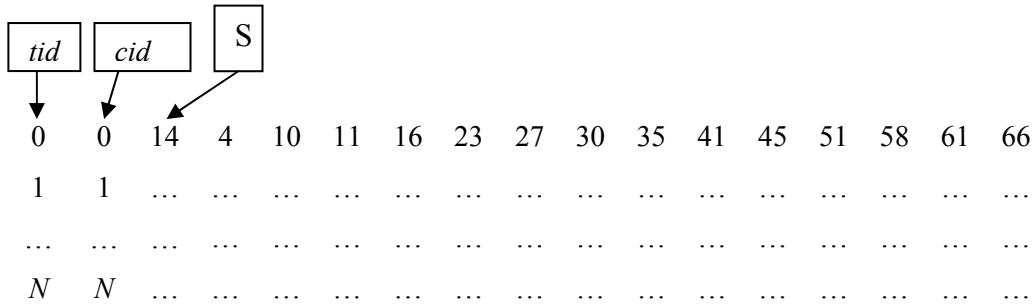


Figure 5.5: Example of Wine dataset after mapping with integer values formatted as used in Charm (Zaki & Hsiao, 2002), and GenMax (Gouda & Zaki, 2001). *Tid:* transaction-id*; cid:* omitted (i.e., equal to *tid*); S: size of string

5.7 Frequent Itemsets Mining

As discussed in Section 5.5, only relevant attributes are used in discovering a set of frequent items. Hence, the filtered training dataset $\tilde{D}_{tr}$, $\tilde{I} \subseteq I$ (i.e., irrelevant attributes removed) is utilized for the generation of frequent itemsets. In the definitions that follow, the frequent patterns/association rules discussed correspond to those that have a class label. There are four types of rule sets discovered from each dataset, namely the Apriori rules (two variants), Maximal rules and Closed rules. Each of the rules is defined as follows:

5.7.1 Apriori Algorithm

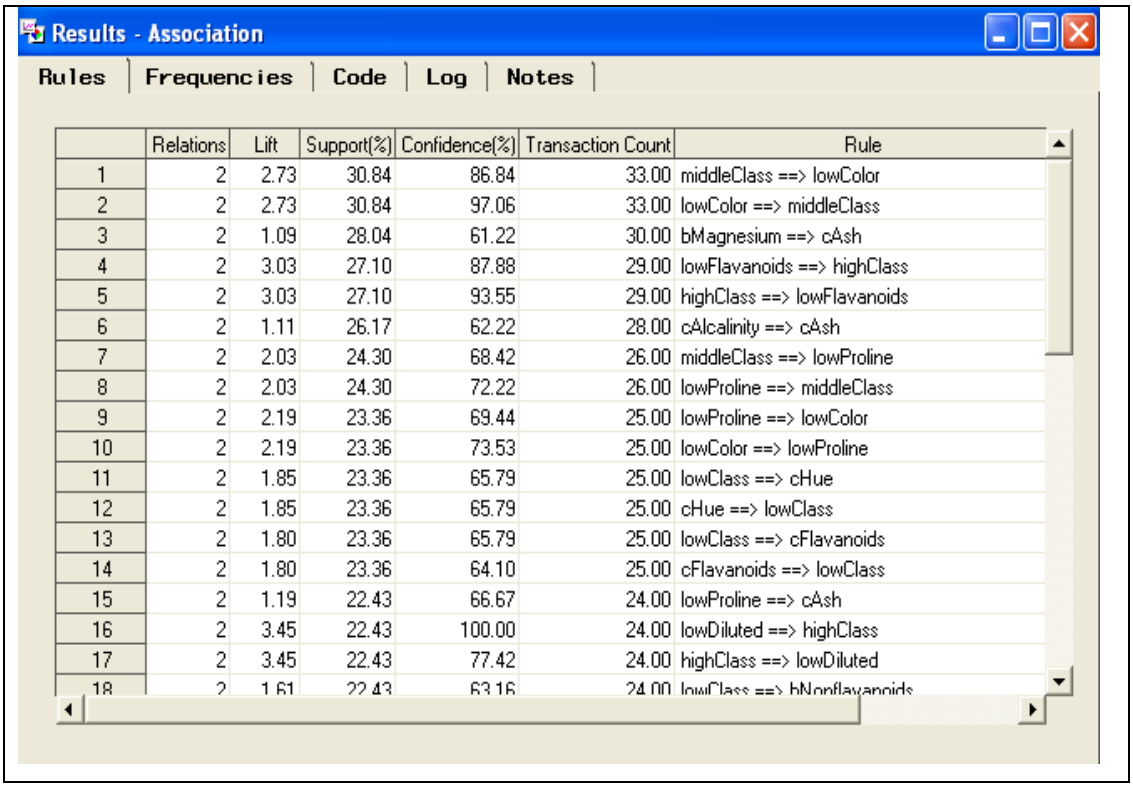
STEP 4a: Apriori (support and confidence). For a given $\tilde{D}_{tr}$, the association rules were generated using the Apriori framework using minimum support (*min*_sup) and minimum confidence (*min_conf*) thresholds, and the set of obtained association rules is denoted by $F(A)$.

Definition 2 Association rule is denoted as $fA = x \to y$, where x is the antecedent (item or a set of items) and y the consequent (class value), $\exists\{x_r, y_r\} \in \tilde{D}_{tr}, x \subseteq x_r, x_r = \{at_1 val_r, at_2 val_r, ..., at_{|AT|} val_r\}$ and $y = y_r \in Y$ is a class label, and $\forall fA_k \in F(A), (k = (1,...,|F(A)|))$, and fA_k satisfies the *min_sup* and *min_conf* thresholds.

STEP 4b: Apriori (support). For a given $\tilde{D}_{tr}$, the association rules were generated using the Apriori framework with only the *min_sup* threshold, and the set of obtained association rules is denoted by $F(B)$.

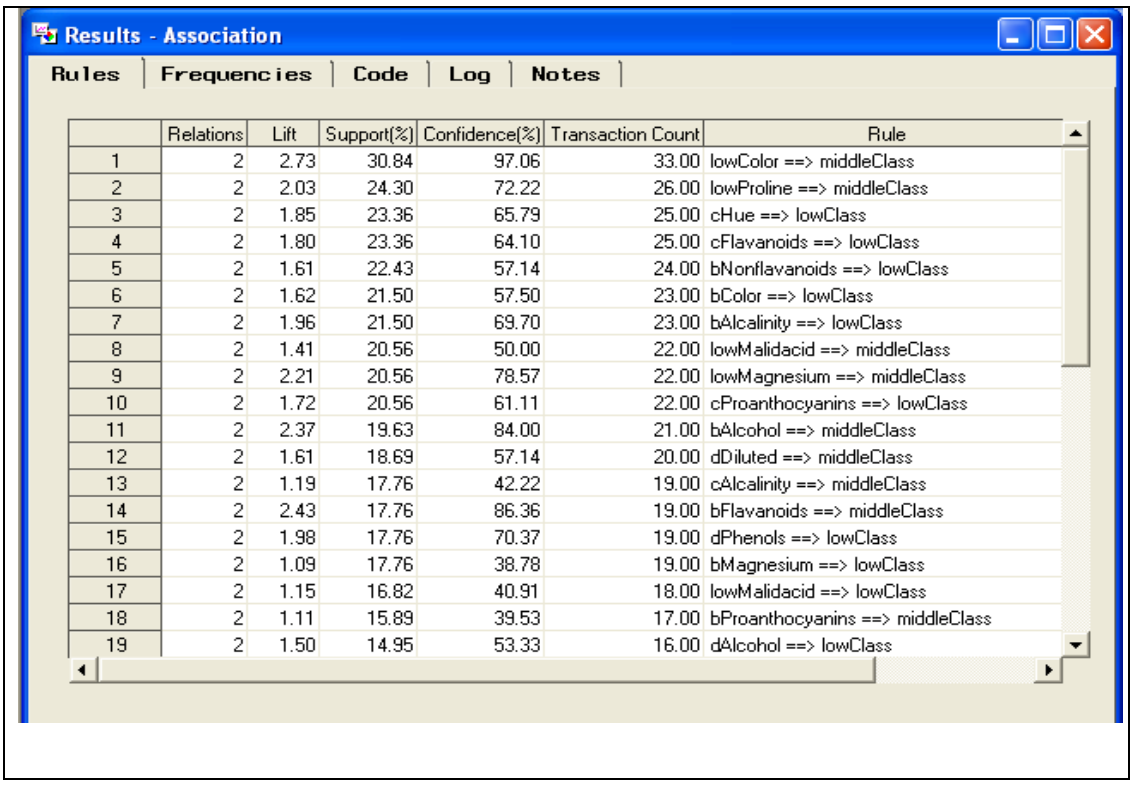
Definition 3 Association rule is denoted as $fB = x \to y$, where x is the antecedent (item or a set of items) and y the consequent (class value), $\exists\{x_r, y_r\} \in \tilde{D}_{tr}, x \subseteq x_r, x_r = \{at_1 val_r, at_2 val_r, ..., at_{|AT|} val_r\}$ and $y = y_r \in Y$ is a class label, and $\forall fB_k \in F(B), (k = (1,...,|F(B)|))$, and fB_k satisfies the *min_sup* threshold.

Figure 5.6 gives an example of association rules generated from the Wine dataset using the Apriori algorithm without setting any class to be predicted, while Figure 5.7 displays the frequent patterns with a class label being selected.



	Relations	Lift	Support(%)	Confidence(%)	Transaction Count	Rule
1	2	2.73	30.84	86.84	33.00	middleClass ==> lowColor
2	2	2.73	30.84	97.06	33.00	lowColor ==> middleClass
3	2	1.09	28.04	61.22	30.00	bMagnesium ==> cAsh
4	2	3.03	27.10	87.88	29.00	lowFlavanoids ==> highClass
5	2	3.03	27.10	93.55	29.00	highClass ==> lowFlavanoids
6	2	1.11	26.17	62.22	28.00	cAlcalinity ==> cAsh
7	2	2.03	24.30	68.42	26.00	middleClass ==> lowProline
8	2	2.03	24.30	72.22	26.00	lowProline ==> middleClass
9	2	2.19	23.36	69.44	25.00	lowProline ==> lowColor
10	2	2.19	23.36	73.53	25.00	lowColor ==> lowProline
11	2	1.85	23.36	65.79	25.00	lowClass ==> cHue
12	2	1.85	23.36	65.79	25.00	cHue ==> lowClass
13	2	1.80	23.36	65.79	25.00	lowClass ==> cFlavanoids
14	2	1.80	23.36	64.10	25.00	cFlavanoids ==> lowClass
15	2	1.19	22.43	66.67	24.00	lowProline ==> cAsh
16	2	3.45	22.43	100.00	24.00	lowDiluted ==> highClass
17	2	3.45	22.43	77.42	24.00	highClass ==> lowDiluted
18	2	1.61	22.43	63.16	24.00	lowClass ==> bNonflavanoids

Figure 5.6: Example Association rules for Wine dataset without specify any Class Labels



	Relations	Lift	Support(%)	Confidence(%)	Transaction Count	Rule
1	2	2.73	30.84	97.06	33.00	lowColor ==> middleClass
2	2	2.03	24.30	72.22	26.00	lowProline ==> middleClass
3	2	1.85	23.36	65.79	25.00	cHue ==> lowClass
4	2	1.80	23.36	64.10	25.00	cFlavanoids ==> lowClass
5	2	1.61	22.43	57.14	24.00	bNonflavanoids ==> lowClass
6	2	1.62	21.50	57.50	23.00	bColor ==> lowClass
7	2	1.96	21.50	69.70	23.00	bAlcalinity ==> lowClass
8	2	1.41	20.56	50.00	22.00	lowMalidacid ==> middleClass
9	2	2.21	20.56	78.57	22.00	lowMagnesium ==> middleClass
10	2	1.72	20.56	61.11	22.00	cProanthocyanins ==> lowClass
11	2	2.37	19.63	84.00	21.00	bAlcohol ==> middleClass
12	2	1.61	18.69	57.14	20.00	dDiluted ==> middleClass
13	2	1.19	17.76	42.22	19.00	cAlcalinity ==> middleClass
14	2	2.43	17.76	86.36	19.00	bFlavanoids ==> middleClass
15	2	1.98	17.76	70.37	19.00	dPhenols ==> lowClass
16	2	1.09	17.76	38.78	19.00	bMagnesium ==> lowClass
17	2	1.15	16.82	40.91	18.00	lowMalidacid ==> lowClass
18	2	1.11	15.89	39.53	17.00	bProanthocyanins ==> middleClass
19	2	1.50	14.95	53.33	16.00	dAlcohol ==> lowClass

Figure 5.7: Example association rules for Wine dataset with Class Labels

5.7.2 Maximal Itemset Mining Algorithm

STEP 4c: Maximal. For a given $\tilde{D}_{tr}$, the Maximal frequent patterns were produced using the *GenMax* algorithm (Gouda & Zaki, 2001), and the set of obtained patterns is denoted as $F(M)$. As explained earlier, only the patterns/rules with a class label are taken into account, and hence the definitions that follow are based on such patterns

Definition 4 Maximal Frequent Pattern is denoted as $fM = x \rightarrow y$, where x is the antecedent (item or a set of items) and y the consequent (class value), $\exists\{x_r, y_r\} \in \tilde{D}_{tr}, x \subseteq x_r, x_r = \{at_1 val_r, at_2 val_r, ..., at_{|AT|} val_r\}$ and $y = y_r \in Y$ is a class label, and $\forall fM_k \in F(M), (k = (1,...,|F(M)|))$, fM_k satisfies the *min_sup* threshold and is not a subset of any other frequent pattern.

5.7.3 Closed Itemset Mining Algorithm

STEP 4d: Closed. For a given $\tilde{D}_{tr}$, the Closed frequent patterns were produced using the *CHARM* algorithm (Zaki & Hsiao, 2002), and the set of obtained patterns is denoted as $F(C)$.

Definition 5 Closed Frequent Pattern is denoted as $fC = x \rightarrow y$, where x is the antecedent (item or a set of items) and y the consequent (class value), $\exists\{x_r, y_r\} \in \tilde{D}_{tr}, x \subseteq x_r, x_r = \{at_1 val_r, at_2 val_r, ..., at_{|AT|} val_r\}$ and $y = y_r \in Y$ is a class label, and $\forall fC_k \in F(C), (k = (1,...,|F(C)|))$, and fC_k satisfies the *min_sup* threshold and it has no superset with the same frequency. As the formats for both the Maximal rules and the Closed rules are identical, Figure 5.8 shows an example of Closed rules, as outputted by the *Charm* algorithm (Zaki & Hsiao, 2002), before mapping the number into specific attributes' name values and Figure 5.9 shows the Closed rules after the mapping process.

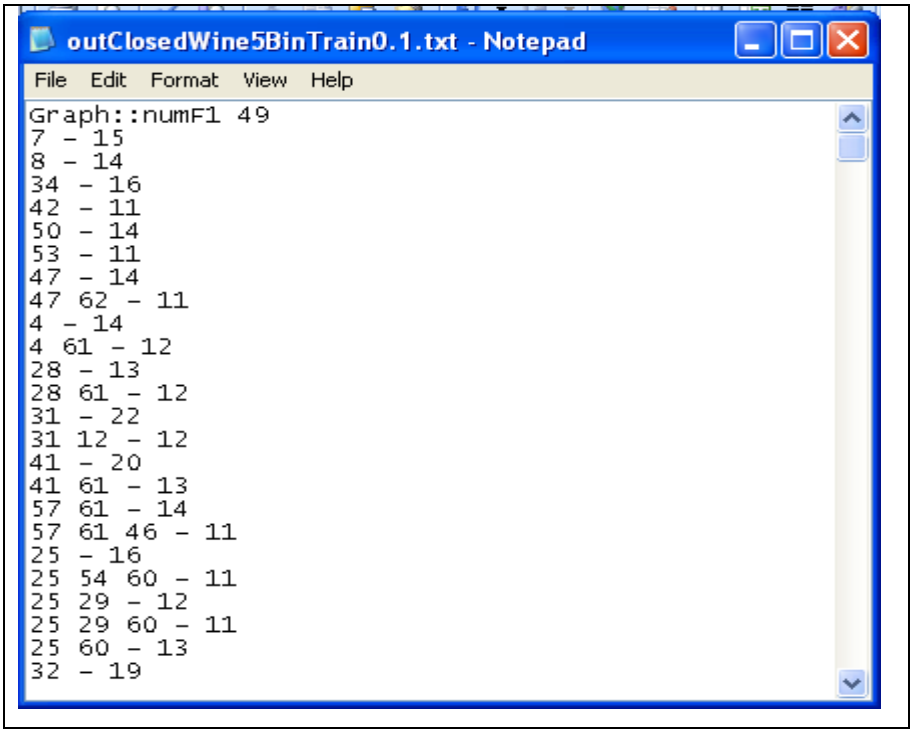


Figure 5.8: Frequent patterns for Wine dataset using Closed algorithm (before Mapping Process)

	A	B	D	E	G
1	**Rules**	**SET_SIZE**	**CONF**	**SUPPORT**	**COUNT**
2	dFlavanoids ==> lowClass	2	92.31	11.21	12
3	cColor ==> lowClass	2	65.00	12.15	13
4	cMagnesium ==> lowClass	2	64.00	14.95	16
5	bFlavanoids ==> middleClass	2	86.36	17.76	19
6	lowMagnesium ==> middleClass	2	78.57	20.56	22
7	lowFlavanoids ==> highClass	2	87.88	27.10	29
8	lowColor ==> middleClass	2	97.06	30.84	33
9	bColor ==> lowClass	2	57.50	21.50	23
10	bMagnesium ==> middleClass	2	26.53	12.15	13
11	cFlavanoids ==> middleClass	2	35.90	13.08	14
12	cFlavanoids ==> lowClass	2	64.10	23.36	25
13	bMagnesium ==> lowClass	2	38.78	17.76	19
14	cMagnesium & cFlavanoids ==> lowClass	3	100.00	10.28	11
15	bFlavanoids & lowColor ==> middleClass	3	100.00	14.95	16
16	lowMagnesium & lowColor ==> middleClass	3	100.00	17.76	19
17	bColor & lowFlavanoids ==> highClass	3	100.00	11.21	12
18	bMagnesium & lowFlavanoids ==> highClass	3	100.00	15.89	17
19	lowColor & bMagnesium ==> middleClass	3	100.00	10.28	11
20	lowColor & cFlavanoids ==> middleClass	3	92.86	12.15	13
21	bColor & cFlavanoids ==> lowClass	3	95.00	17.76	19
22	bColor & bMagnesium ==> lowClass	3	58.33	13.08	14

Figure 5.9: Frequent patterns for Wine dataset using Closed algorithm (after Mapping Process)

5.7.4 Minimum *Support* and Minimum *Confidence* Thresholds

The determination of optimal minimum support and minimum confidence thresholds in a data mining system are flexible and can be done interactively by the users (Han & Kamber, 2001). This includes allowing the user to test and modify the interestingness measures and their respective thresholds.

Thus, in this thesis, the focus is on investigating the role that the *confidence* measure plays in the rules evaluation process by studying the implications of using high/or low *confidence* measures and applying confidence-based filtering later during the rule optimization process. Setting a low *minimum support threshold* is preferred as this will ensure the discovery of low frequency itemsets, even though this will generate a larger volume of patterns. (Roiger & Geatz, 2003) assert that if more rules are desired, the coverage criterion can be lowered. As mentioned earlier in Chapter 3, the application of feature subset selection in the proposed framework will facilitate the removal of a large volume of patterns that have been derived from low minimum support thresholds. In addition, during the rule evaluation phase (explained next), many unnecessary rules will be removed, and hence it is preferable to set the *minimum support* threshold fairly low to avoid the possibility of missing any rare yet significant associations.

5.8 Rule Evaluation

5.8.1 Statistical Analysis

The statistical analysis is applied to the training dataset in order to identify the significant attributes to be used for the rules evaluation process. These involve two steps, the first step is based on chi-squared test and the second step is the logistic regression analysis. Both of these measures are used to identify the set of irrelevant features which in the framework will be utilized to identify rules that contain those features and are likely to be unnecessary and should be removed.

However, note that, for the Maximal rule set, discarding the rules that contain non-significant variables would reduce the number of rules significantly. This is due to the fact that, by definition, the Maximal rule set will not contain subsets of a frequent

pattern, and the preference is for longer patterns. Hence, if a maximal pattern contains an irrelevant attribute, and that pattern/rule is removed, the association between other relevant attributes in the pattern would not be present elsewhere in the rule sets and hence would be lost.

In Closed and especially Apriori, this would not be a problem as they can contain subsets of rules (as is always the case in Apriori), and hence another rule capturing the association, without the irrelevant attribute is likely to exist. Thus, for Maximal rules, rather than discarding a rule, only non-significant variables within the rules are removed, which will be reflected in the formulizations that follow. However, some rules may still be removed as a result of the process, as if all attributes within the precedent of a rule are detected as insignificant, the rule is removed. Removing one or more non-significant input attributes from a Maximal rule results in more simplified rules (less attributes in the precedent of a rule) in the rule sets. This can cause these simplified rules to already have a representative in the rule set, and in this case, only one rule is preserved.

5.8.1.1 Chi-Squared Test

A natural way to express the dependence between antecedent and the consequence of an association rule $x \to y$ is the correlation based on the chi-squared test for independence (Brijs, Vanhoof, & Wets, 2003). Thus, the step in chi-squared test is defined as follows:

STEP 5a: Chi-squared test. For a given $\tilde{D}_{tr}$, the occurrence of at_i where $at_i \in AT, (i = (1,...,|AT|))$ is independent of the occurrence of Y if $P(at_i \cup Y) = P(at_i)P(Y)$; otherwise at_i and Y are dependent and correlated (Han & Kamber, 2001). The correlation between at_i and Y is measured using Equation 3 as follows:

$$corr(at_i, Y) = \frac{P(at_i \cup Y)}{P(at_i)P(Y)} \quad (3)$$

For a given correlation value based on Equation 3, the chi-squared χ^2 statistic value was utilized to determine whether the correlation is statistically significant.

Hence, the chi-squared test discards any $fA_k \in F(A)$, $fB_k \in F(B)$ and $fC_k \in F(C)$ for which $\exists at_i$ contained in x of $x \to y$, the χ^2 value is not significant for Y (class attribute) (correlation analysis in Equation 3).

For Maximal frequent patterns, the chi-squared test simplifies any $fM_k \in F(M)$ where $\exists at_i$ contained in x of $x \to y$, the χ^2 value is not significant for Y (class attribute) (correlation analysis in Equation 3) in the following way: If $at_i \in x$ in $x \to y$ then the rule $fM_k \in F(M)$ is simplified to $x' \to y$, where $x' = x \setminus at_i$.

5.8.1.2 Logistic Regression

Another form of statistical analysis that was applied was the logistic regression. The relationship between the antecedent and consequent in association rule mining can be presented as a relationship between a target variable and the input variables in logistic regression. The following is the definition of the logistic regression model involved in the framework.

STEP 5b: Logistic regression analysis. For a given $\tilde{D}_{tr}$, several logistic regression models were developed based on Equation 4.

$$\ln(Y) = \beta_0 + \beta_1 at_1 + \beta_2 at_2 + ... + \beta_{|AT|} at_{|AT|} + \varepsilon \,, \tag{4}$$

where;

$\ln(Y)$ = Natural logarithm of the odds ratio,

$\beta_0, \beta_1, ..., \beta_{|AT|}$ = Coefficients of the input attributes,

ε = Error variable,

Y = Dichotomous class attribute,

$at_1, at_2 ..., at_{|AT|}$ = Input attributes

The co-efficient β of an input attribute at_i where $at_i \in AT, (i = (1,...,|AT|))$, i.e. $\beta_i at_i$ from Equation 4 is determined based on the log likelihood value given in Equation 5 (note that as per Definition 1 the value of attribute at_i occurring in record *r* is denoted as $at_i val_r$). The statistical hypothesis is then used to determine whether the input attributes are significantly related to the class attribute.

$$\beta_i at_i = \sum_{r=1}^{n} \{y_r \ln[\pi(at_i val_r)] + (1 - y_r) \ln[1 - \pi(at_i val_r)]\} \quad (5)$$

As mentioned in Chapter 4, a number of models can be developed from logistic regression analysis. Each model produces a different selection of variables. Such a result is possible because different variables may contain different/complementary information that contributes to the prediction of the value of the target variable. Thus, each model needs to be evaluated and the one which is the most parsimonious, fits the data well, and has the highest predictive capability, is selected. The selected model is denoted as $\ln(\tilde{Y})$.

Hence, logistic regression $\ln(\tilde{Y})$ discards any $fA_k \in F(A)$, $fB_k \in F(B)$ and $fC_k \in F(C)$ for which $\exists at_i$ contained in x of $x \to y$, the $\beta_i at_i$ value is not significant towards the class attribute Y (logistic regression analysis in Equation 4).

For Maximal frequent patterns, logistic regression $\ln(\tilde{Y})$ simplifies any rule, $fM_k \in F(M)$ where $\exists at_i$ contained in x of $x \to y$, the $\beta_i at_i$ value is not significant for the class attribute Y (logistic regression analysis in Equation 4), in the following way: If $at_i \in x$ in $x \to y$ then the rule $fM_k \in F(M)$ is simplified to $x' \to y$, where $x' = x \setminus at_i$.

From the set of frequent rules $F(A)$, $F(B)$, $F(M)$ and $F(C)$, the resulting sets that have been reduced according to the statistical analysis in steps 5a and 5b are denoted as $FS(A) = \{fsA_1, fsA_2, ..., fsA_{|FS(A)|}\}$, $FS(B) = \{fsB_1, fsB_2, ..., fsB_{|FS(B)|}\}$,

$FS(M) = \{fsM_1, fsM_2, ..., fsM_{|FS(M)|}\}$ and $FS(C) = \{fsC_1, fsC_2, ..., fsC_{|FS(C)|}\}$, respectively.

5.8.2 Redundancy and Contradictive Removal

The existence of the redundant and the contradictive rules is still a significant issue in presenting statistically valid patterns. A discussion of the problem of redundant and contradictive rules was presented in Chapter 3. Here, the details of the development of redundancy and contradictive rule removal are given.

STEP 6: *Productive* rules based on minimum improvement redundant rule constraint (Bayardo, Agrawal, & Gunopulos, 2000), discards any $fsA_k \in FS(A)$, $fsB_k \in FS(B)$, $fsM_k \in FS(M)$ and $fsC_k \in FS(C)$ if *confidence* $(x \rightarrow y) \leq \max_{z \subset x}(confidence(z \rightarrow y))$.

In other words, a rule $x \rightarrow y$ with confidence value c_1 is considered as redundant if there exists another rule $z \rightarrow y$ with confidence value c_2, where $z \subset x$ and $c_1 \leq c_2$.

From the set of statistically reduced frequent rules, $FS(A)$, $FS(B)$, $FS(M)$ and $FS(C)$ the resulting sets that have been reduced according to the minimum improvement redundant rule assessment in step 6 are denoted as $FR(A) = \{frA_1, frA_2, ..., frA_{|FR(A)|}\}$, $FR(B) = \{frB_1, frB_2, ..., frB_{|FR(B)|}\}$, $FR(M) = \{frM_1, frM_2, ..., frM_{|FR(M)|}\}$ and $FR(C) = \{frC_1, frC_2, ..., frC_{|FR(C)|}\}$, respectively.

STEP 7: Contradictive rule constraint (Zhang & Zhang, 2001) discards any two rules $frX_j, frX_k \in FR(X)$ if $frX_j = x \rightarrow y$ and $frX_k = x \rightarrow \neg y$, where $j, k = (1, ..., |FR(X)|)$, $X = (A, B, M, C)$ and $j \neq k$.

Please note that in the above formulation, the focus is on two rules for the sake of simplicity; the contradictive rule constraint will discard two or more contradictive rules as long as they all imply a different class value.

From the rule sets, $FR(A)$, $FR(B)$, $FR(M)$ and $FR(C)$ the resulting sets that have been reduced according to contradictive rule removal in step 7 are denoted as $FC(A) = \{fcA_1, fcA_2, ..., fcA_{|FC(A)|}\}$, $FC(B) = \{fcB_1, fcB_2, ..., fcB_{|FC(B)|}\}$, $FC(M) = \{fcM_1, fcM_2, ..., fcM_{|FC(M)|}\}$ and $FC(C) = \{fcC_1, fcC_2, ..., fcC_{|FC(C)|}\}$, respectively.

5.8.3 Filtering Rules Based on *Confidence* Threshold

In the later experiment in Chapter 6, as the rules are progressively reduced based on the minimum *confidence* threshold, an optimal stopping criteria based on a *confidence* threshold needs to be chosen to ensure that the accuracy rates (AR) and coverage rates (CR) are at an acceptable level.

STEP 8: *Confidence* Based Filtering. From the rule sets, $FC(A)$, $FC(B)$, $FC(M)$ and $FC(C)$, the rule set that have been progressively filtered based on the *confidence* threshold are denoted as $FCF(A) = \{fcfA_1, fcfA_2, ..., fcf_{|FCF(A)|}\}$, $FCF(B) = \{fcfB_1, fcfB_2, ..., fcfB_{|FCF(B)|}\}$, $FCF(M) = \{fcfM_1, fcfM_2, ..., fcfM_{|FCF(M)|}\}$ and $FCF(C) = \{fcfC_1, fcfC_2, ..., fcfC_{|FCF(C)|}\}$, respectively.

5.9 Rules Accuracy and Rules Coverage

The combination of statistical analysis, redundancy and contradictive assessment methods provide an appropriate means of discarding non-significant rules. However, the question still remains whether this great reduction of rules is at cost of a significant reduction in the rules' accuracy and rules' coverage. Hence, the rule coverage, classification and predictive accuracy of the different sets of rules obtained by this process are evaluated. This step can be defined as follows:

STEP 9: Determining the accuracy and coverage rate of rule sets. For each of the resulting rule sets, namely the rule sets $(F(A), F(B), F(M)$ and $F(C))$, rule sets after statistical analysis $(FS(A), FS(B), FS(M)$ and $FS(C))$, rule sets after

redundancy removal $\left(FR(A),\ FR(B),\ FR(M)\ \text{and}\ FR(C)\right)$, the rule sets after contradictive removal ($FC(A)$, $FC(B)$, $FC(M)$ and $FC(C)$) and rule set based on the *confidence* based filtering (*FCF(A)*, *FCF(B)*, *FCF(M)* and *FCF(C)*), calculate the classification accuracy by determining the percentage of correctly classified instances in D_{tr}, the predictive accuracy determining the percentage of correctly classified instances in D_{ts}, and the coverage rate (number of captured instances) in both D_{tr} and D_{ts}. The combination of these rule evaluation strategies will enable the association rule mining framework to determine the level of correctness and quality of rules. These rules will have a sound statistical basis and one can be more confident that they reflect the real-world situation.

5.9.1 Pseudo code for the Rules Accuracy and Rule Coverage

The algorithm that specifies the rules' accuracy and rules' optimization process is outlined in Figure 5.10. This algorithm is used to determine the accuracy (AR) and coverage rate (CR) of the selected rule sets for each algorithm.

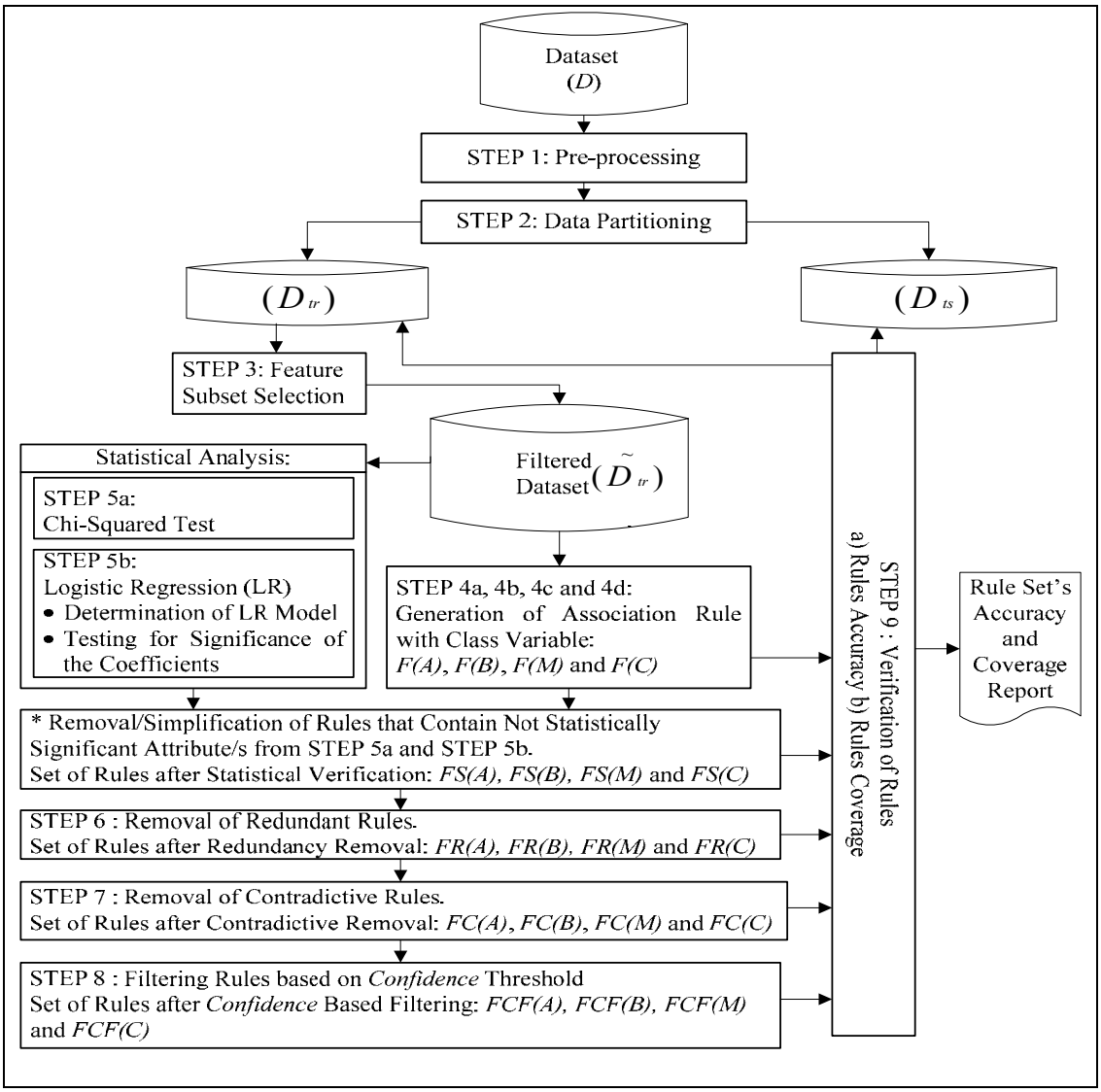
Input: A sets of rule (*F(A)*, *F(B), F(M)* and *F(C)*), a sets of statistical reduced set of rule (*FS(A)*, *FS(B)*, *FS(M)* and *FS(C)*); a sets of rule after redundancy removal (*FR(A)*, *FR(B)*, *FR(M)* and *FR(C)*); a sets of rule after contractive removal (*FC(A)*, *FC(B)*, *FC(M)* and *FC(C)*); a set of rules after *confidence* based filtering (*FCF(A)*, *FCF(B)*, *FCF(M)* and *FCF(C)*); Training and Testing dataset
Output: Accuracy (AR) and coverage rate (CR) of the rule set
    For each rule, scan the training and testing dataset
        Check whether rules classifies all the instances in dataset
        Calculate Misclassification Rate (MR) for each rule
    AR = (1- sum of all MRs)* 100
    CR = (1 – (Number of Uncovered Instances /Total Number of Instances)) * 100
return AR and CR

Figure 5.10: Pseudo code for the Rules Accuracy and Rules Coverage

5.10 Conclusion

This chapter has provided a detailed description of the way that the association rules derived from relational data are evaluated using the proposed framework. Pre-processing is undertaken so that only clean data and a suitable data format are used in the association rule mining process. The dataset is then divided into two partitions. The first partition is used for the feature subset selection method, association rule

generation, statistical analysis, and redundancy and contradictive assessment; while the second partition acts as sample data drawn from the database, used to verify the accuracy and coverage of the discovered rules. The selection of significant attributes is an important aspect of the association rule mining process. This will ensure that less frequent patterns are enumerated, and in most cases, it was utilized here to provide the relative usefulness of attributes in predicting the value of the class attribute, and discard any of the attributes whose relevance value is fairly low. This would prevent the generation of rules which then would need to be discarded anyway if they are found to include irrelevant attributes. The Apriori association algorithm is then utilized to generate the frequent itemsets including the maximal and closed frequent itemsets. The discovered rules are then evaluated using statistical analysis, and any rules determined to be statistically insignificant are discarded. The chi-squared test is used to discover the properties of data attributes; principally regarding the data dependency. The logistic regression analysis is then employed to provide the classification power of the data. The development of logistic regression modeling involves the model building strategies. Additionally, redundancy and contradictive assessment methods are employed to discard redundant and contradictive rules. Within this chapter, the way that the rules are evaluated based on the aforementioned steps is formalized. At the end of this chapter, the accuracy and coverage rate of reducing the number of rules based on the statistical analysis, redundancy and contradictive assessment, and filtering based on confidence threshold, are described. The whole process is illustrated in Figure 5.11 and the steps involved are summarized in the pseudo code in Figure 5.12. The experiments used to evaluate the developed framework are provided in the following chapter.



(*) = The removal/simplification of rule/s based on statistical analysis is done in the following sequence: i) First, rule/s that contains non-significant attribute/s based on STEP 5a is removed/simplified; ii) Second, rule/s that contains non-significant attribute/s based on STEP 5b is removed/simplified.

Figure 5.11: Rules Evaluation Process

Input: database D with class attribute
Output: Accuracy Rate (AR) and Coverage Rate (CR) for each of rule set *(F(A), F(B), F(M)* and *F(C))*; (*FS(A), FS(B), FS(M)* and *FS(C))*; *(FR(A), FR(B), FR(M)* and *FR(C))*; (*FC(A), FC(B), FC(M)* and *FC(C))*; and *(FCF(A), FCF(B), FCF(M) and FCF(C))*.

1. Apply the pre-processing technique towards each at_i in D, where $at_i \in AT, (i = (1,...,|AT|))$,

2. Divide the database D into D_{tr} and D_{ts},

3. Calculate Symmetrical τ of input attributes at_i in D_{tr}, where $at_i \in AT, (i = (1,...,|AT|))$ in predicting the value of the class attribute $Y \in D_{tr}$. Discard irrelevant at_i and denote the

filtered database as $\tilde{D}_{tr}$.

4. Generate a frequent set of rules *F(A)*, *F(B)*, *F(M)* and *F(C)* from $\tilde{D}_{tr}$.

5. Identify the non-significant attributes at_i in $\tilde{D}_{tr}$ where $at_i \in AT, (i = (1,...,|AT|))$ based on statistical analysis,

5.1. Chi-squared test;

Calculate the χ^2 values from $corr(at_i, Y)$ and identify at_i which are not significantly correlated with Y (class attribute),

for each $corr(at_i, Y) \in \tilde{D}_{tr}$, where $at_i \in AT, (i = (1,...,|AT|))$

if χ^2 is not significant then discard any $fA_k \in F(A)$, $fB_k \in F(B)$ and $fC_k \in F(C)$ that contains at_i in x of $x \rightarrow y$

(For Maximal patterns);

if χ^2 is not significant, then simplify any $fM_k \in F(M)$ that contain at_i in x of $x \rightarrow y$ to $x' \rightarrow y$, where $x' = x \setminus at_i$.

5.2. Logistic Regression;

(*Determination of Logistic Regression Model by Fitting the Data and Testing for Significance of the Coefficient*)

Develop and fit several Logistic Regression models from $\tilde{D}_{tr}$. Choose a model that

Fits the data with the highest predictive capabilities, denoted as ($\ln(\tilde{Y})$). Estimate

the coefficient of at_i, $\beta_i at_i$ in $\ln(\tilde{Y})$ and identify at_i which are not significantly correlated to Y (class attribute),

for each $\beta_i at_i$ values in $\ln(\tilde{Y})$ where $at_i \in AT, (i = (1,...,|AT|))$

if $\beta_i at_i$ is not significant then discard any $fA_k \in F(A)$, $fB_k \in F(B)$ and $fC_k \in F(C)$ that contains at_i in x of $x \rightarrow y$

(For Maximal patterns),

if $\beta_i at_i$ is not significant, then simplify any $fM_k \in F(M)$ that contain at_i in x of $x \rightarrow y$ to $x' \rightarrow y$, where $x' = x \setminus at_i$.

Denote the statistically reduced/simplified set of rules as *FS(A)*, *FS(B)*, *FS(M)* and *FS(C)*.

6. Apply *Minimum Improvement Redundancy Removal* to reduce *FS(A)*, *FS(B)*, *FS(M)* and *FS(C)* into *FR(A)*, *FR(B)*, *FR(M)* and *FR(C)*, respectively.

7. Apply *Contradictive Rule Constraints Removal* to reduce *FR(A)*, *FR(B)*, *FR(M)* and *FR(C)* into *FC(A), FC(B), FC(M)* and *FC(C)*, respectively.

8. Progressively filter *FC(A), FC(B), FC(M)* and *FC(C)* based on *confidence* threshold into *FCF(A), FCF(B), FCF(M) and FCF(C)*.

9. Calculate the AR and CR for each rule set *(F(A)*, *F(B)*, *F(M)* and *F(C))*, (*FS(A)*, *FS(B)*, *FS(M)* and *FS(C)), (FR(A)*, *FR(B)*, *FR(M)* and *FR(C))*, (*FC(A), FC(B), FC(M)* and *FC(C))* and *(FCF(A), FCF(B), FCF(M) and FCF(C))* towards the D_{tr} and D_{ts}.

return AR and CR for each of rule set *(F(A)*, *F(B)*, *F(M)* and *F(C))*; (*FS(A)*, *FS(B)*, *FS(M)* and *FS(C))*; *(FR(A)*, *FR(B)*, *FR(M)* and *FR(C))*; (*FC(A), FC(B), FC(M)* and *FC(C))*; and *(FCF(A), FCF(B), FCF(M)* and *FCF(C))*.

Figure 5.12: Pseudo code of the rules evaluation framework

References

Bayardo, R. J., Agrawal, R., & Gunopulos, D. (2000). Constraint-Based Rule Mining in Large, Dense Databases. *Data Mining and Knowledge Discovery, 4*, 217-240.

Blanchard, J., Guillet, F., Gras, R., & Briand, H. (2005). Using Information-Theoretic Measures to Assess Association Rule Interestingness. In *Proceedings of the 5th IEEE International Conference on Data Mining*: IEEE Computer Society.

Brijs, T., Vanhoof, K., & Wets, G. (2003). Defining interestingness for association rules. *International Journal of Information Theories and Applications, 10(4)*, 370-376.

Dillon, T. S., Hossain, T., Bloomer, W., & Witten, M. (1998). Improvements in Supervised BRAINNE: A Method for Symbolic Data Mining Using Neural Networks. In S. Spaccapietra & F. J. Maryanski (Eds.), *IFIP TC2/WG2.6 Seventh Conference on Database Semantics (DS-7)* (Vol. 124, pp. 67-88). Leysin, Switzerland: Chapman & Hall.

Gouda, K., & Zaki, M. J. (2001). Efficiently Mining Maximal Frequent Itemsets. In *First IEEE International Conference on Data Mining (ICDM'01)* (Vol. 0, pp. 163).

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

Ke, Y., Cheng, J., & Ng, W. (2008). An information-theoretic approach to quantitative association rule mining. *Knowl. Inf. Syst., 16*, 213-244.

Lakshminarayan, K., Harp, S. A., & Samad, T. (1999). Imputation of Missing Data in Industrial Databases. *Applied Intelligence, 11*, 259-275.

Refaat, M. (2007). *Data preparation for data mining using SAS*. San Francisco: Morgan Kaufmann Publishers.

Roiger, R., & Geatz, M. (2003). *Data mining: a tutorial-based primer*. Boston: Addison Wesley.

Tan, P.-N., Kumar, V., & Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Edmonton, Alberta, Canada: ACM.

Wang, J. (2003). *Data mining: opportunities and challenges*. Hershey, PA: Idea Group Pub.

Zaki, M. J., & Hsiao, C.-J. (2002). CHARM: An Efficient Algorithm for Closed Itemset Mining. In *2nd SIAM International Conference in Data Mining*.

Zhang, C., & Zhang, S. (2001). Collecting Quality Data for Database Mining. In *AI 2001: Advances in Artificial Intelligence* (pp. 131-142).

Zhou, X. J., & Dillon, T. S. (1991). A statistical-heuristic feature selection criterion for decision tree induction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*, 834-841.

CHAPTER 6 EVALUATION OF FRAMEWORK FOR RELATIONAL DATA

6.1 Introduction

This chapter describes several experiments performed in order to evaluate the developed framework for relational data. It also serves the other purpose of demonstrating several important issues mentioned throughout the thesis with respect to evaluating the interestingness of rules derived from relational data. This includes an experiment of selecting an appropriate and useful feature subset selection application and the applications of chi-squared test and logistic regression analysis. The existing frequent itemsets mining approaches include the Apriori, Maximal and Closed. In the developed framework, the frequent itemsets are first generated from the Apriori algorithm. Two variants of this algorithm are utilized in order to study the effect of applying the confidence threshold at different stages. Furthermore, the developed framework has been utilized to evaluate the rules from Maximal and Closed algorithms. In a later experiment, the observations are made in order to understand the implication and effect of difference confidence usage for each Apriori, Maximal and Closed approach in the developed framework.

6.2 Evaluation of Framework for Relational Data

The evaluation of the proposed unified framework is performed using the Wine, Mushroom, Iris and Adult datasets, which are real-world datasets of varying complexity obtained from the UCI Machine Learning Repository (Frank & Asuncion, 2010). All the datasets used here reflect classification problems in which, for supervised learning, the target variables have been chosen to be the right hand side/consequence of the association rules discovered during association rule mining analysis. The equal sized binning method is utilized for all continuous attributes in the Adult, Iris and Wine datasets and this will ensure that the data sizes are manageable by reducing the number of distinct values per attribute (Han & Kamber, 2001). Other discrete attributes in the Adult and Mushroom datasets were preserved in their original state.

In Section 6.2.1, the characteristics of the datasets used for generating the association rule and the initial number of rules discovered with each approach are presented (the

initial rules refer to the rules obtained by just simply applying the association rule mining without utilizing any of the steps defined within the proposed framework). Section 6.2.2 is devoted to feature subset selection experiments while Section 6.2.3 and Section 6.2.4 provide the experiments for chi-squared test and logistic regression analysis, respectively. The comparison between the rules discovered from the Apriori framework that consist of both minimum *support* and minimum *confidence* thresholds and the Apriori framework with only a minimum *support* threshold are shown in Section 6.2.5. The rules are then progressively verified based on statistical analysis, namely logistic regression and chi-squared test, redundancy and contradictive assessment methods. The comparison of the quality of the rule sets discovered from the Apriori, Maximal and Closed algorithms is presented in Section 6.2.6. In Section 6.2.7, a comparison is carried out to ascertain the effect of altering the minimum *confidence* values of the evaluation process for each algorithm.

6.2.1 Dataset Characteristics

Table 6.1 indicates the characteristics of the aforementioned datasets used in our evaluation. It shows the number of records, the number of attributes, and the number of selected attributes based on Symmetrical Tau (ST) features selection in each dataset.

Table 6.1: Dataset Characteristics

Dataset	**# Records**	**# Attr.***	**# Selected Attr. (Sym. Tau)**	**# of Rules with Target Variable**			
				Apriori (*min_sup* & *min_conf*)	**Apriori (*min_sup*)**	**Maximal**	**Closed**
Wine	178	14	12	234	272	103	242
Adult	45222	15	10	1680	2192	129	1866
Mushroom	8124	23	11	75237	77815	255	3009
Iris	150	5	4	51	58	13	41

(*): including the Class variable

The table also shows the number of initial rules generated by the Apriori, Maximal and Closed algorithms. The first Apriori algorithm (in Column 5) will act as the initial benchmark having both the minimum *support* and the minimum *confidence* in generating the rule set. The second variant (in Column 6) will discover only the rules based on the minimum *support* value. One reason for having only a minimum *support* in the second variant is to give a fair evaluation between the Apriori,

Maximal and Closed algorithms as both rule sets generated from the Maximal and Closed algorithms are constrained only by a minimum *support* threshold. Another reason is to evaluate the effect of applying the confidence measure for filtering after statistical and redundant and contradictive based filtering has been applied (rational of which as discussed in Chapter 5 (Section 5.1).

The selected attributes based on ST in Column 4 in Table 6.1 are measured according to their capabilities in predicting the values of attribute class in each dataset such as 'Adult : Income' (<=50K and >50K), 'Wine : Classes' (Low, Middle and High), 'Mushroom : Classes' (Edible and Poisonous) and 'Iris : Classes' (Setosa, Versicolour and Virginica). Columns 5 and 6 contain the number of rules discovered by each association rule mining technique for each dataset. In Table 6.1, there are two variants of Apriori utilized in this thesis. The first variant refers to the Apriori framework with a minimum *support* of 10% and a minimum *confidence* of 60%, while the second variant represents the Apriori framework using only a predefined minimum *support* of 10%. Columns 7 and 8 refer to the *Maximal* and Closed algorithms respectively in generating frequent itemsets with a minimum *support* of 10%.

6.2.2 Feature Subset Selection Process and Comparison of Symmetrical Tau (ST) and Mutual Information (MI)

Using the whole dataset as input would produce a large number of rules, many of which are created by the presence of irrelevant attributes. ST and MI are capable of measuring the relevance of attributes in predicting a class value, but they are different from each other in terms of their approach as aforementioned in Chapter 5. They can both be used as a means of selecting a feature subset to be used for rule generation, and in this section the two approaches are compared in terms of their general properties, and utilization for the feature subset selection process. At the end of the section, the feature subsets used for each of the datasets considered in the experimental evaluation is indicated.

The ST and MI measures for all the attributes in the Mushroom, Adult, Wine and Iris datasets are shown in Table 6.2, 6.3 6.4 and 6.5, respectively. The attributes were

ranked according to their decreasing ST and MI values. Based on the experiment with the Adult dataset, the MI approach seems to favor variables with more values. This can be observed in Table 6.2 for the Adult dataset as variables with more values have all been ranked in the top 7 based on the MI measure (i.e. Education(16), Occupation(14), Education Number(8), Age(10) and Hour PerWeek(10)), while each one of these is ranked lower based on ST, with attribute Capital Gain(6) occurring higher than all these attributes with more values.. Similarly, for the Mushroom dataset, variables with more values such as Gcolor(12), Scabovering(9), Scbelowring(9), are all ranked higher based on MI in contrast to ST ranking. For example, the ST measure has ranked the attribute Gsize with only 2 values as third in the ranking, higher than all these multi-valued attributes, whereas in the MI ranking the Gsize is seventh in the ranking after all those multi-valued attributes.

This observation of MI preference for multi-valued attributes is in accord with that of (Blanchard, Guillet, Gras, & Briand, 2005). In contrast, the procedure based on ST produces a more stable selection of variables which does not favor the multi-valued nature of attributes. This is in agreement with the claim by (Zhou & Dillon, 1991) that ST is fair in handling multi-valued variables. However, the question still remains of how the ST and MI methods compare when used for the purpose of feature subset selection.

Table 6.2: Comparison between ST and MI for Adult Dataset

# of Values	Variables	ST Values		# of Values	Variables	MI Values
7	Marital Status	0.1448		6	Relationships	0.1662
6	Relationship	0.1206		7	Marital Status	0.1575
6	Capital Gain	0.0706		16	Education	0.0934
8	Education Number	0.0688		14	Occupation	0.0932
16	Education	0.0528		8	Education Number	0.0900
2	Sex	0.0470		10	Age	0.0894
14	Occupation	0.0469		10	Hours Per Week	0.0545
10	Age	0.0432		6	Capital Gain	0.0475
5	Capital Loss	0.0361		2	Sex	0.0374
10	Hours Per Week	0.0354		5	Capital Loss	0.0238
7	Work Class	0.0166		7	Work Class	0.0171
5	Race	0.0085		41	Native Country	0.0093
41	Native Country	0.0077		5	Race	0.0083
10	FNLWGT	0.0002		10	FNLWGT	0.0002

Table 6.3: Comparison between ST and MI for Mushroom Dataset

Feature Subset Selection Based on ST				Feature Subset Selection Based o MI		
# of Values	Variables	ST Values		# of Values	Variables	MI Values
9	Odor	0.5872		9	Odor	0.9127
9	SporePrintColor	0.3246		9	SporePrintColor	0.4812
2	Gsize	0.2866		12	Gcolor	0.4078
5	Ringtype	0.2585		5	Ringtype	0.3172
2	Bruises	0.2487		9	Scabovering	0.251
12	Gcolor	0.2172		9	Scbelowring	0.2404
9	Scabovering	0.1462		2	Gsize	0.2271
6	Pop	0.1454		6	Pop	0.197
9	Scbelowring	0.1405		2	Bruises	0.1897
2	Gspacing	0.1298		7	Habitat	0.1578
7	Habitat	0.0980		2	Gspacing	0.1088
3	Ringnumber	0.0460		6	Cshape	0.0487
4	Sroot	0.0439		3	Ringnumber	0.0409
6	Cshape	0.0299		4	Sroot	0.0402
4	Csurface	0.0234		10	Ccolor	0.0356
10	Ccolor	0.0227		4	Csurface	0.0249
4	Veilcolor	0.0214		4	Veilcolor	0.0222
4	Ssabovering	0.0169		4	Ssbelowring	0.0166
4	Ssbelowring	0.0150		4	Ssabovering	0.0163
2	Sshape	0.0150		2	Gattachment	0.0122
2	Gattachment	0.0146		2	Sshape	0.0108
1	Veiltype	0.0000		1	Veiltype	0.0000

Table 6.4: Comparison between ST and MI for Wine Dataset

# of Values	Variables	ST Values		# of Values	Variables	MI Values
5	Flavanoids	0.4810		5	Flavanoids	0.8796
5	Color	0.4226		5	Diluted	0.8476
5	Diluted	0.3610		5	Color	0.7914
5	Proline	0.3543		5	Proline	0.7422
5	Hue	0.3019		5	Hue	0.6242
5	Alcohol	0.2367		5	Phenols	0.5591
5	Phenols	0.2312		5	Alcohol	0.5275
5	Magnesium	0.1840		5	Magnesium	0.3717
5	Alcalinity	0.1680		5	Proanthocyanins	0.3275
5	Proanthocyanins	0.1525		5	Alcalinity	0.3143
5	Malidacid	0.1403		5	Malidacid	0.2821
5	Nonflavanoids	0.1313		5	Nonflavanoids	0.2730
5	Ash	0.0499		5	Ash	0.0996

Table 6.5: Comparison between ST and MI for Iris Dataset

# of Values	Variables	ST Values		# of Values	Variables	MI Values
5	Petal Width	0.6738		5	Petal Width	1.311
5	Petal Length	0.6355		5	Petal Length	1.226
5	Sepal Length	0.2724		5	Sepal Length	0.618
5	Sepal Width	0.2301		5	Sepal Width	0.508

When using an attribute relevance measure for the feature subset selection problem, commonly a relevance cut-off point is chosen below which all attributes are removed. Hence, in the ranking of attributes according to their decreasing ST and MI values in Tables 6.2-6.5, a relevance cut-off needs to be set. Here, the cut-off point was selected based on the significant difference between the ST/MI values in decreasing order. The significant difference was considered to occur in the ranking at the position where that attribute's ST/MI value is less than half of the previous attribute's ST/MI value in the ranking, respectively. At this point and below in the ranking, all attributes are considered as irrelevant. In Tables 6.2-6.5, all the attributes that are considered as irrelevant based on this way of determining the cut-off value, are shaded gray. As one can see, the way in which feature subsets would be selected based on ST and MI measures, differs for the Adult dataset only. Hence, the performance of these two subsets when used for generating association rules for classification purposes will be evaluated next. Additionally, in the Iris dataset (Table 6.5), all input variables were considered in the experiments, as Iris dataset consists of only 4 attributes, and complexity problems would not occur.

For the Adult dataset, by ranking the attributes based on ST values, 10 input attributes are selected based on the aforementioned way of determining the cut-off value, while 13 input attributes are favored based on MI ranking. The cut-off point at and below which all attributes are considered as irrelevant, is shown in Table 6.2, where cells of attributes removed are shaded gray. Rules are then generated based on these 10 and 13 input variables and evaluated for their classification/predictive accuracy and coverage rate. As depicted in Table 6.6, for this dataset, the selection of 10 input attributes that were ranked based on ST resulted in 303 rules in comparison to 1726 rules when they were ranked by MI. This was not at the cost of a reduction in coverage rate; moreover, accuracy was slightly better for both the training and testing datasets.

Table 6.6: Rules Evaluation between attributes selected based on ST and MI for Adult dataset

	Data Partition	Symmetrical Tau			Mutual Information		
		# Of Rules	AR %	CR%	# Of Rules	AR %	CR %
Initial # of Rules	Training	2192	68.98	100.00	2192	68.98	100.00
	Testing		69.05	100.00		69.05	100.00
Rule # from feature subset	Training	303	67.46	100.00	1726	67.36	100.00
	Testing		67.45	100.00		67.38	100.00

As shown in the previous experiment for the Adult dataset, the ST has more advantageous properties in comparison with MI, as the feature subset selected according to the ST measure, resulted in many less rules which at the same time had a slightly higher accuracy and the same coverage rate of 100%. In addition, from the ranking of the different attributes relevance measures (i.e. Tables 6.2-6.5), it was shown that MI tends to favor multi-valued attributes in comparison to ST. Given these observation as well as others' claims (Zhou & Dillon, 1991) in regards to the advantageous properties of ST over other existing measures, the ST feature selection criterion was used within the framework as the first step in order to remove any irrelevant attributes. This would prevent the generation of rules that include any irrelevant attributes. Hence, in the experiments it is not necessary to use ST to further verify the rules as the rules were created from the attribute subset considered as relevant according to the measure, as was done in (Shaharanee, Hadzic, & Dillon, 2009; Shaharanee, Hadzic, & Dillon, 2011).

For example, the comparison results for the Adult dataset are shown in Table 6.2, where the capabilities of attributes in predicting the values of attribute 'Income' (<=50K and >50K) are measured. For the Adult dataset results presented in Table 6.2, the relevance cut-off value is 0.0166. This is due to the ST value of attribute 'Hours per week' being more than double the ST value for attribute 'Work class'. Thus, the subset of data now consists of 10 attributes: Marital status, Relationship, Capital gain, Education number, Education, Sex, Occupation, Age, Capital loss and Hours per week. Similarly for the Mushroom dataset in Table 6.3, the subset of data after the feature subsets selection process consists of 11 attributes: Odor, Spore Print Color, Gill Size, Ring Type, Bruises, Gill Color, Population, Stalk Color Above Ring, Stalk Color Below Ring, Gill Spacing and Habitat. For the Wine dataset (Table

6.4), only the *Ash* input variable has been discarded from further analysis. The next section discusses the verification of the extracted rules through statistical analysis.

6.2.3 Chi-Squared Test

A natural way to express the dependence between antecedent and the consequence of an association rule $x \rightarrow y$ is the correlation based on the chi-squared test for independence (Brijs, Vanhoof, & Wets, 2003). The chi-squared test results for the Mushroom, Iris and Wine datasets show that all variables passed the chi-squared basic requirement. The requirements are that all cells in the contingency table have expected values greater than 1 and at least 80% of the cells have expected values greater than 5.

However, for the Adult dataset, of the two categorical variables combination, one input variable namely *CapitalGain* has been discarded as the variable failed the basic chi-squared requirements. The variable is independent of the target variable *Income* and is considered not significant.

Thus, any rules that consist of the *Capital Gain* variable can be removed. Table 6.7 shows examples of rules that have been removed based on the chi-squared test for the Adult dataset. (A total of 151 numbers of rules are removed.)

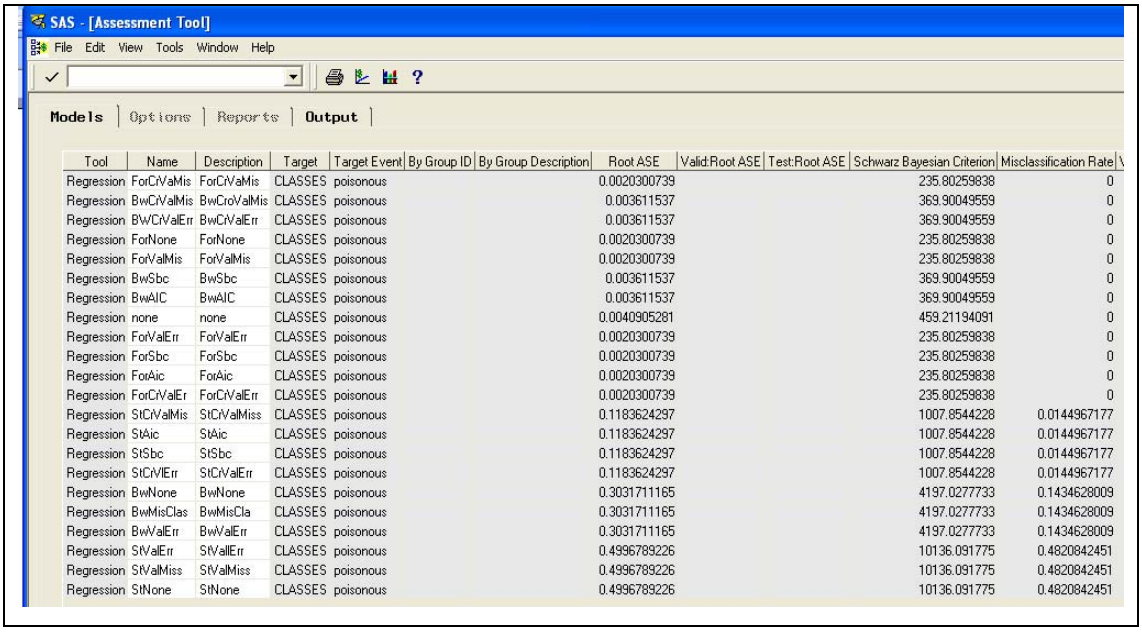
Table 6.7: Example of a pruned rule based on Chi-squared test for the Adult data

Set Size	Confidence (%)	Support (%)	Rules
2	76.94	75.07	lowCgain ==> lowIncome
3	78.40	72.86	lowCloss & lowCgain ==> lowIncome

6.2.4 Logistic Regression Analysis

The relationship between the antecedent and consequent in association rule mining can be presented as a relationship between a target variable and the input variables in logistic regression. As mentioned in Section 3.7.3.3 (Chapter 3), from logistic regression a number of models can be discovered. This result is possible because different variables may contain different/complementary information that contributes

to the prediction of the value of the target variable. Each model (second column of the assessment tool figures) will have a set of attributes with a specific misclassification rate. The model with the lowest misclassification rate (last column of the assessment tool figures) and the most parsimonious is selected. Figures 6.1 to 6.4 depict the assessment for each of the dataset.



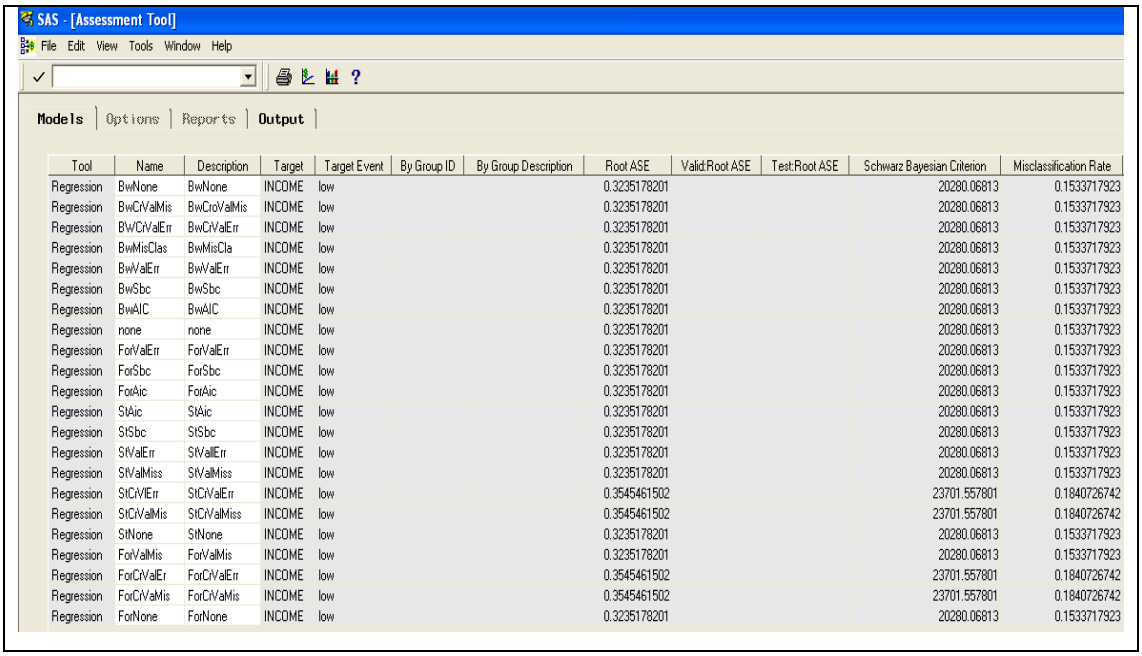
Tool	Name	Description	Target	Target Event	By Group ID	By Group Description	Root ASE	Valid:Root ASE	Test:Root ASE	Schwarz Bayesian Criterion	Misclassification Rate
Regression	ForCrValMis	ForCrValMis	CLASSES	poisonous			0.0020300739			235.80259838	0
Regression	BwCrValMis	BwCroValMis	CLASSES	poisonous			0.003611537			369.90049559	0
Regression	BWCrValErr	BwCrValErr	CLASSES	poisonous			0.003611537			369.90049559	0
Regression	ForNone	ForNone	CLASSES	poisonous			0.0020300739			235.80259838	0
Regression	ForValMis	ForValMis	CLASSES	poisonous			0.0020300739			235.80259838	0
Regression	BwSbc	BwSbc	CLASSES	poisonous			0.003611537			369.90049559	0
Regression	BwAIC	BwAIC	CLASSES	poisonous			0.003611537			369.90049559	0
Regression	none	none	CLASSES	poisonous			0.0040905281			459.21194091	0
Regression	ForValErr	ForValErr	CLASSES	poisonous			0.0020300739			235.80259838	0
Regression	ForSbc	ForSbc	CLASSES	poisonous			0.0020300739			235.80259838	0
Regression	ForAic	ForAic	CLASSES	poisonous			0.0020300739			235.80259838	0
Regression	ForCrValEr	ForCrValErr	CLASSES	poisonous			0.0020300739			235.80259838	0
Regression	StCrValMis	StCrValMiss	CLASSES	poisonous			0.1183624297			1007.8544228	0.0144967177
Regression	StAic	StAic	CLASSES	poisonous			0.1183624297			1007.8544228	0.0144967177
Regression	StSbc	StSbc	CLASSES	poisonous			0.1183624297			1007.8544228	0.0144967177
Regression	StCrVlErr	StCrValErr	CLASSES	poisonous			0.1183624297			1007.8544228	0.0144967177
Regression	BwNone	BwNone	CLASSES	poisonous			0.3031711165			4197.0277733	0.1434628009
Regression	BwMisClas	BwMisCla	CLASSES	poisonous			0.3031711165			4197.0277733	0.1434628009
Regression	BwValErr	BwValErr	CLASSES	poisonous			0.3031711165			4197.0277733	0.1434628009
Regression	StValErr	StValIErr	CLASSES	poisonous			0.4996789226			10136.091775	0.4820842451
Regression	StValMiss	StValMiss	CLASSES	poisonous			0.4996789226			10136.091775	0.4820842451
Regression	StNone	StNone	CLASSES	poisonous			0.4996789226			10136.091775	0.4820842451

Figure 6.1: Assessment for logistic regression model for Mushroom data

As for the Mushroom dataset, using the selected logistic regression model (i.e. Forward Selection Method variant), any rules that contain *Bruises*, *Gill Color*, *Population*, *Ring Type*, *Habitat*, *Stalk Color Below Ring* and *Stalk Color Above Ring* can be discarded, since they are not significant contributors. Table 6.8 depicts the example of pruned rules for Mushroom dataset.

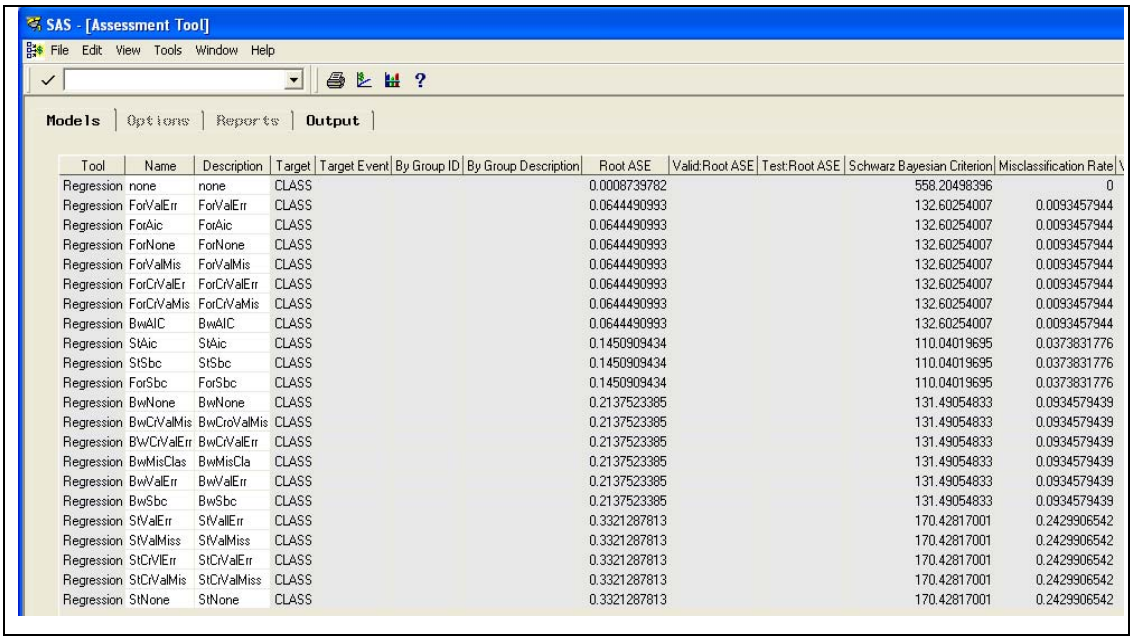
Table 6.8: Example of prunes rules based on logistic regression analysis for Mushroom data

Set Size	Confidence	Support	Rules
7	100 %	12.55 %	whiteSCBelowRing & whiteSCAboveRing & noneOdor & noBruises & crowdedGspacing & broadGSize => edibleClasses
8	100 %	10.80 %	whiteSporePrintColor & severalPop & pinkSCAboveRing & noBruises & narrowGSize & evanescentRingType & closeGspacing=> poisonousClasses



Tool	Name	Description	Target	Target Event	By Group ID	By Group Description	Root ASE	Valid:Root ASE	Test:Root ASE	Schwarz Bayesian Criterion	Misclassification Rate
Regression	BwNone	BwNone	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	BwCrValMis	BwCroValMis	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	BWCrValErr	BwCrValErr	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	BwMisClas	BwMisCla	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	BwValErr	BwValErr	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	BwSbc	BwSbc	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	BwAIC	BwAIC	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	none	none	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	ForValErr	ForValErr	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	ForSbc	ForSbc	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	ForAic	ForAic	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	StAic	StAic	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	StSbc	StSbc	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	StValErr	StValErr	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	StValMiss	StValMiss	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	StCrVlErr	StCrValErr	INCOME	low			0.3545461502			23701.557801	0.1840726742
Regression	StCrValMis	StCrValMiss	INCOME	low			0.3545461502			23701.557801	0.1840726742
Regression	StNone	StNone	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	ForValMis	ForValMis	INCOME	low			0.3235178201			20280.06813	0.1533717923
Regression	ForCrValEr	ForCrValErr	INCOME	low			0.3545461502			23701.557801	0.1840726742
Regression	ForCrVaMis	ForCrVaMis	INCOME	low			0.3545461502			23701.557801	0.1840726742
Regression	ForNone	ForNone	INCOME	low			0.3235178201			20280.06813	0.1533717923

Figure 6.2: Assessment for logistic regression model for Adult data



Tool	Name	Description	Target	Target Event	By Group ID	By Group Description	Root ASE	Valid:Root ASE	Test:Root ASE	Schwarz Bayesian Criterion	Misclassification Rate	\
Regression	none	none	CLASS				0.0008739782			558.20498396	0	
Regression	ForValErr	ForValErr	CLASS				0.0644490993			132.60254007	0.0093457944	
Regression	ForAic	ForAic	CLASS				0.0644490993			132.60254007	0.0093457944	
Regression	ForNone	ForNone	CLASS				0.0644490993			132.60254007	0.0093457944	
Regression	ForValMis	ForValMis	CLASS				0.0644490993			132.60254007	0.0093457944	
Regression	ForCrValEr	ForCrValErr	CLASS				0.0644490993			132.60254007	0.0093457944	
Regression	ForCrVaMis	ForCrVaMis	CLASS				0.0644490993			132.60254007	0.0093457944	
Regression	BwAIC	BwAIC	CLASS				0.0644490993			132.60254007	0.0093457944	
Regression	StAic	StAic	CLASS				0.1450909434			110.04019695	0.0373831776	
Regression	StSbc	StSbc	CLASS				0.1450909434			110.04019695	0.0373831776	
Regression	ForSbc	ForSbc	CLASS				0.1450909434			110.04019695	0.0373831776	
Regression	BwNone	BwNone	CLASS				0.2137523385			131.49054833	0.0934579439	
Regression	BwCrValMis	BwCroValMis	CLASS				0.2137523385			131.49054833	0.0934579439	
Regression	BWCrValErr	BwCrValErr	CLASS				0.2137523385			131.49054833	0.0934579439	
Regression	BwMisClas	BwMisCla	CLASS				0.2137523385			131.49054833	0.0934579439	
Regression	BwValErr	BwValErr	CLASS				0.2137523385			131.49054833	0.0934579439	
Regression	BwSbc	BwSbc	CLASS				0.2137523385			131.49054833	0.0934579439	
Regression	StValErr	StValErr	CLASS				0.3321287813			170.42817001	0.2429906542	
Regression	StValMiss	StValMiss	CLASS				0.3321287813			170.42817001	0.2429906542	
Regression	StCrVlErr	StCrValErr	CLASS				0.3321287813			170.42817001	0.2429906542	
Regression	StCrValMis	StCrValMiss	CLASS				0.3321287813			170.42817001	0.2429906542	
Regression	StNone	StNone	CLASS				0.3321287813			170.42817001	0.2429906542	

Figure 6.3: Assessment for logistic regression model for Wine data

For the Adult dataset, a logistic regression model (i.e. Backward Selection Method variant) is selected; any rules that contain *Education Number* have been removed. Finally, for the Wine dataset, a logistic regression model (i.e. Forward Selection Method variant) is selected, any rules that include variables *Diluted*, *Proline*, *Hue*, *Alcohol*, *Phenols*, *Alcalinity*, *Proanthocyanins*, *Malid Acid* and *Nonflavanoids* have

been removed. Tables 6.9 and 6.10 reveal the example of pruned rules from Adult and Wine datasets, respectively.

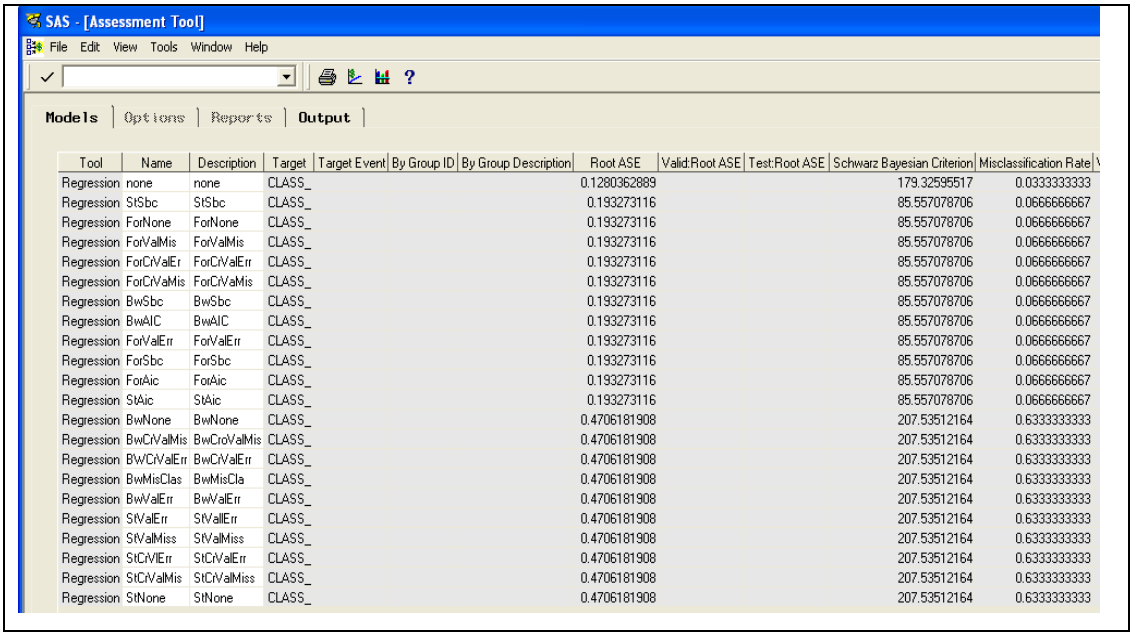
Table 6.9: Example of prunes rules based on logistic regression analysis for Adult data

Set Size	Confidence	Support	Rules
3	83.82 %	26.47 %	eEdunum & dHour ==> lowIncome
4	83.84 %	43.67 %	lowCloss & lowCgain & eEdunum ==> lowIncome

Table 6.10: Example of prunes rules based on logistic regression analysis for Wine data

Set Size	Confidence	Support	Rules
3	100.00 %	16.82 %	lowHue & lowDiluted ==> highClass
4	100.00 %	4.67 %	lowProline & lowPhenols & lowHue ==> highClass

None of the input attributes in the Iris dataset were discarded based on the statistical analysis approach either by the chi-squared or logistic regression. All inputs' attributes in the Iris dataset were statistically significant in predicting the target attributes.



Tool	Name	Description	Target	Target Event	By Group ID	By Group Description	Root ASE	Valid:Root ASE	Test:Root ASE	Schwarz Bayesian Criterion	Misclassification Rate
Regression	none	none	CLASS_				0.1280362889			179.32595517	0.0333333333
Regression	StSbc	StSbc	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	ForNone	ForNone	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	ForValMis	ForValMis	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	ForCrValEr	ForCrValErr	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	ForCrValMis	ForCrValMis	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	BwSbc	BwSbc	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	BwAIC	BwAIC	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	ForValErr	ForValErr	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	ForSbc	ForSbc	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	ForAic	ForAic	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	StAic	StAic	CLASS_				0.193273116			85.557078706	0.0666666667
Regression	BwNone	BwNone	CLASS_				0.4706181908			207.53512164	0.6333333333
Regression	BwCrValMis	BwCroValMis	CLASS_				0.4706181908			207.53512164	0.6333333333
Regression	BWCrValErr	BwCrValErr	CLASS_				0.4706181908			207.53512164	0.6333333333
Regression	BwMisClas	BwMisCla	CLASS_				0.4706181908			207.53512164	0.6333333333
Regression	BwValErr	BwValErr	CLASS_				0.4706181908			207.53512164	0.6333333333
Regression	StValErr	StValErr	CLASS_				0.4706181908			207.53512164	0.6333333333
Regression	StValMiss	StValMiss	CLASS_				0.4706181908			207.53512164	0.6333333333
Regression	StCrVlErr	StCrValErr	CLASS_				0.4706181908			207.53512164	0.6333333333
Regression	StCrValMis	StCrValMiss	CLASS_				0.4706181908			207.53512164	0.6333333333
Regression	StNone	StNone	CLASS_				0.4706181908			207.53512164	0.6333333333

Figure 6.4: Assessment for logistic regression model for Iris data

The use of statistical analysis helps to determine the usefulness and significance of input variables in predicting the target variables. Hence, this is an appropriate means of identifying and discarding rules that are not significant.

In this section, several examples of the application of statistical analysis were provided together with examples of rules being removed. Henceforth, the discussion will focus on comparing the difference between the various utilized approaches, and the quality of their resulting rules, and rules at different steps in the rule verification process. Hence, in the next section, an example of redundant/contradictive rules being removed throughout the process will be provided and the relationship between confidence measure and contradictive rule removal will be discussed.

6.2.5 Apriori *(Min_Sup* & *Min_Conf)* vs. Apriori *(Min_Sup)*

Apriori algorithms have demonstrated a good performance in generating frequent patterns (Zaki & Hsiao, 2002). However, the patterns generated need to be evaluated in order to arrive at significant and useful patterns. A unification framework for evaluating the interestingness of frequent itemsets obtained by the Apriori algorithm was previously developed in this thesis and reported in (Shaharanee et al., 2009; Shaharanee et al., 2011). It was found that the rules generated from the Apriori algorithm were large and contaminated with useless patterns. With appropriate statistical analysis, and redundancy and contradictive assessment methods, the unification framework managed to discard a large number of rules while still preserving high accuracy and coverage rate of the final reduced rule set.

In this section, the usefulness of the rules generated from both variants is compared. Table 6.11 reveals the progressive difference in the number of rules, the Accuracy Rate (AR) and the Coverage Rate (CR) values as the ST feature selection application, statistical analysis, redundancy and contradictive assessment methods are utilized.

For most of the discovered rules in Table 6.11, the AR in the training set (i.e. classification accuracy) was consistently higher than the testing set (predictive accuracy). This is due to the fact that the discovered rules were generated from the

training set, and as a consequence, the rules mostly fit well the characteristics of the data objects that exist predominantly in the training set.

Table 6.11: Comparison between Apriori (*Min_Sup & Min Conf*) and Apriori (*Min_Sup*) in Wine Dataset

Type of analysis	**Data Partition**	**Apriori (*Min_Sup & Min Conf*)**			**Apriori (*Min_Sup*)**		
		# Of Rules	**AR %**	**CR%**	**# Of Rules**	**AR %**	**CR %**
Initial # of Rules	Training	234	87.58	100.00	272	76.83	100.00
	Testing		79.84	100.00		69.68	100.00
# of Rules after ST	Training	195	87.53	100.00	217	74.26	100.00
	Testing		79.44	100.00		68.00	100.00
Statistics Analysis	Training	17	85.07	100.00	24	64.16	100.00
	Testing		81.98	100.00		60.46	100.00
Redundant Removal	Training	16	85.07	100.00	23	63.52	100.00
	Testing		81.98	100.00		60.05	100.00
Contradictive Removal	Training	16	85.07	100.00	16	85.63	100.00
	Testing		81.98	100.00		81.94	100.00
Conf. 60%	Training				15	87.84	100.00
	Testing					84.77	100.00

The initial number of rules from Apriori constrained with *min_sup* is larger compared to the initial number of rules in Apriori constrained with both *min_sup* and *min_conf* due to the removal of the minimum *confidence* threshold. As application of the Symmetrical Tau, statistical analysis and redundancy assessment were progressively applied to the initial set of rules; at least 90% of the rules in the rule set have been discarded. Both AR values for the testing dataset in Apriori (with *min_sup* and *min_conf)* and Apriori (with *min_sup*) increased while the CR of the rules was still preserved at 100%. The rule sets at this stage are shown in Table 6.12.

Table 6.12: Apriori (*Min_Sup & Min Conf*) and Apriori (*Min_Sup*) rules for Wine dataset after filtering according to statistical analysis and redundancy assessment

Apriori (*Min_Sup & Min Conf*) – 16 # of Rules			**Apriori (*Min_Sup*) – 23 # of Rules**		
Conf. (%)	**Sup. (%)**	**Rules**	**Conf. (%)**	**Sup. (%)**	**Rules**
92.31	11.21	Flavanoids(3.18 - 4.13) ==> Class(Low)	97.06	30.84	ColorIntensity(low - 3.62) ==> Class(Middle)
97.06	30.84	ColorIntensity(low - 3.62) ==> Class(Middle)	87.88	27.10	Flavanoids(low - 1.29) ==> Class(High)
65	12.15	ColorIntensity(5.97 - 8.31) ==> Class(Low)	64.10	23.36	Flavanoids(2.24 - 3.18)==> Class(Low)
86.36	17.76	Flavanoids(1.29 - 2.24) ==> Class(Middle)	57.50	21.50	ColorIntensity(3.62 - 5.97) ==> Class(Low)
87.88	27.1	Flavanoids(low - 1.29) ==> Class(High)	78.57	20.56	Magnesium(low - 88.4) ==> Class(Middle)
78.57	20.56	Magnesium(low - 88.4) ==> Class(Middle)	86.36	17.76	Flavanoids(1.29 - 2.24) ==> Class(Middle)
64.1	23.36	Flavanoids(2.24 - 3.18)==> Class(Low)	38.78	17.76	Magnesium(88.4 - 106.8) ==> Class(Low)
64	14.95	Magnesium(106.8- 125.2) ==> Class(Low)	34.69	15.89	Magnesium(88.4 - 106.8) ==> Class(High)
100	17.76	Magnesium(low - 88.4) & ColorIntensity(low - 3.62) ==> Class(Middle)	64.00	14.95	Magnesium(106.8- 125.2) ==> Class(Low)
100	15.89	Flavanoids(low - 1.29) & Magnesium(88.4 - 106.8) ==> Class(High)	35.90	13.08	Flavanoids(2.24 - 3.18)==> Class(Middle)
100	10.28	Magnesium(106.8- 125.2) & Flavanoids(2.24 - 3.18)==> Class(Low)	26.53	12.15	Magnesium(88.4 - 106.8) ==> Class(Middle)
100	14.95	ColorIntensity(low - 3.62) & Flavanoids(1.29 - 2.24) ==> Class(Middle)	65.00	12.15	ColorIntensity(5.97 - 8.31) ==> Class(Low)
100	10.28	ColorIntensity(low - 3.62) & Magnesium(88.4 - 106.8) ==> Class(Middle)	92.31	11.21	Flavanoids(3.18-4.13) ==> Class(Low)
95	17.76	Flavanoids(2.24 - 3.18)& ColorIntensity(3.62 - 5.97) ==> Class(Low)	30.00	11.21	ColorIntensity(3.62 - 5.97) ==> Class(High)
73.33	10.28	Flavanoids(2.24 - 3.18)& Magnesium(88.4 - 106.8) ==> Class(Low)	100	17.76	Magnesium(low - 88.4) & ColorIntensity(low - 3.62) ==> Class(Middle)
100	11.21	Flavanoids(low - 1.29) & ColorIntensity(3.62 - 5.97) ==> Class(High)	95.00	17.76	Flavanoids(2.24 - 3.18)& ColorIntensity(3.62 - 5.97) ==> Class(Low)
			100	15.89	Flavanoids(low - 1.29) & Magnesium(88.4 - 106.8) ==> Class(High)
			100	14.95	ColorIntensity(low - 3.62) & Flavanoids(1.29 - 2.24) ==> Class(Middle)
			58.33	13.08	Magnesium(88.4 - 106.8) & ColorIntensity(3.62 - 5.97)

			==> Class(Low)
	100	11.21	Flavanoids(low - 1.29) & ColorIntensity(3.62 - 5.97) ==> Class(High)
	100	10.28	ColorIntensity(low - 3.62) & Magnesium(88.4 - 106.8) ==> Class(Middle)
	100	10.28	Magnesium(106.8- 125.2) & Flavanoids(2.24 - 3.18)==> Class(Low)
	73.33	10.28	Flavanoids(2.24 - 3.18)& Magnesium(88.4 - 106.8) ==> Class(Low)

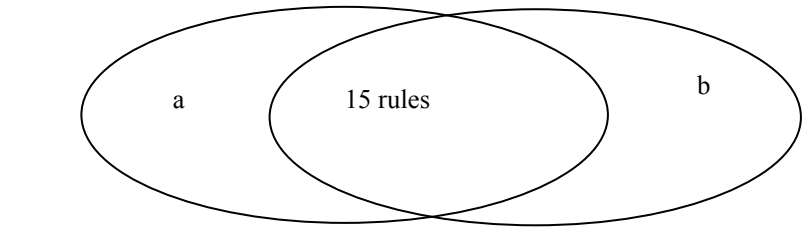
As an extension of our previous work in (Shaharanee, Dillon, & Hadzic, 2009), another method of analysis to discard contradictive rules (Zhang & Zhang, 2001) was included. Contradictive rules exist in Apriori (with *min_sup*) because they are constrained by only a minimum *support* threshold, because at the set *confidence* threshold of 60% in Apriori (with *min_sup & min_conf*), they do not exist. However, this also points to the important difference. The rules with confidence higher than 60% that are contradictive to other frequent rules in the data, which cannot be present in the rule set as they cannot have 60% confidence at the same time, will remain in the rule set, but will have a higher misclassification rate. Hence, their contradictive nature would not be captured, which essentially would negatively affect the accuracy of the rule set as a whole. An example of this scenario is provided later. The contradictive rules detected in Apriori (with *min_sup*) rule set are shown in Table 6.13.

Table 6.13: List of contradictive rules in Wine dataset for Apriori (*Min_Sup*)

Conf. (%)	**Sup. (%)**	**Rules**
64.10	23.36	Flavanoids(2.24 - 3.18) ==> Class(Low)
35.90	13.08	Flavanoids(2.24 - 3.18) ==> Class(Middle)
57.50	21.50	ColorIntensity(3.62 - 5.97) ==> Class(Low)
30.00	11.21	ColorIntensity(3.62 - 5.97) ==> Class(High)
38.78	17.76	Magnesium(88.4 - 106.8) ==> Class(Low)
34.69	15.89	Magnesium(88.4 - 106.8) ==> Class(High)
26.53	12.15	Magnesium(88.4 - 106.8) ==> Class(Middle)

With the removal of the contradictive rules in Apriori (with *min_sup*), both approaches now contain the same number of rules (16) with only a modest difference in AR% as shown in Table 6.11. Even though both contain the same number of rules,

there are still differences as shown in Figure 6.5. These differences are due to the sequence of the evaluation process in both approaches. Rule (b) does not appear in Apriori (with *min_sup & min_conf*) due to the *confidence* value being lower than the minimum threshold of 60%, while rule (a) does not exist in Apriori (with *min_sup*) because the rule contradicts another rule (see Table 6.13 row 3).



	Conf. (%)	Sup. (%)	Rules
a	64.10	23.36	Flavanoids(2.24 - 3.18) ==> Class(Low)
b	58.33	13.08	Magnesium(88.4 - 106.8) & ColorIntensity(3.62 - 5.97) ==> Class(Low)

Figure 6.5: Rule differences between Apriori (*Min_Sup & Min Conf*) and Apriori (*Min_Sup*) after contradictive rule removal

Finally, the minimum *confidence* constraint was utilized on the Apriori (with *min*_sup) rule set and 15 rules were obtained as our final significant rule set (i.e. Rule (b) from Figure 6.5 was removed). As for the final 15 rules, the AR value in Apriori (with *min*_sup) is higher than Apriori (with *min_sup & min_conf*), while the CR value remained the same (see Table 6.11). When the individual accuracy of each rule was checked, it was exactly the rules (a) and (b) (Figure 6.5) causing lower AR in the rules from Apriori (with *min_sup & min_conf*) and Apriori (with *min_sup*), respectively. The rule (a) was discarded in Apriori (with *min_sup*) because it contradicted another rule as shown in Table 6.13.

This knowledge of the rule (a) being contradictive to another rule (frequent association to another class value) was not available in Apriori (with *min_sup & min_conf*) because the minimum *confidence* constraint was applied at the start. This approach missed the fact that association "Flavanoids(2.24 - 3.18) ==> Class(Middle)" occurred frequently enough to know that the rule "Flavanoids(2.24 -

3.18) ==> Class(Low)" is not reliable enough to be used for prediction. This is supported by the fact that the AR of the final 15 rules is higher than the AR of the 16 rules from Apriori (with *min_sup & min_conf*) containing the contradictive rule. In Apriori (with *min_sup & min_conf*), the contradictive rule "Flavanoids(2.24 - 3.18) ==> Class(Low)" has misclassified 14 instances from the training set and 10 instances from the testing set. By removing this rule, a portion of the misclassified instances is captured by other rule(s) that are based on different attribute constraints, and there is an increase in accuracy as seen in Table 6.11.

These results suggest that it may be advantageous to not apply the *confidence* constraints at the start of the process but rather at the end or after any contradictive frequent rules/patterns have been removed. Another option would be to start with a lower *confidence* threshold to still discard those patterns where the *confidence* is not high enough for them to be considered as a significant contradiction to another rule with much higher *confidence*. One can then increase the threshold, and the effects of progressively increasing the *confidence* threshold are shown in Section 6.2.6. This relationship between contradictive rules and the application of a *confidence* threshold was not discussed in (Zhang & Zhang, 2001) where the contradictive assessment was introduced.

The comparison of the rules generated from the Apriori (with *min_sup & min_conf*) and Apriori (with *min_sup*) of the Iris, Mushroom and Adult datasets is fairly similar to the rules extracted from the Wine dataset. The initial rule set from the Apriori (with *min_sup*) algorithm is naturally always larger than the rule set of the Apriori (with *min_sup & min_conf*) algorithm as depicted in Tables 6.14, 6.15 and 6.16. The ST application, statistical analysis, redundancy and contradictive assessment methods, and a specific minimum *confidence* threshold (for Apriori (with *min_sup*)) are progressively applied to each rule set. As the number of rules for each dataset and each variant was reduced dramatically, the AR for the training and the testing dataset increased gradually except for the rule set from Apriori (with *min_sup & min_conf*) for Iris and Mushroom dataset as there are slight decreases in their AR. While the CR for each of the Mushroom and Iris datasets was well preserved at 100%, the CR in Adult marginally decreased. The Adult dataset is characterized by imbalanced target data, as discussed in (Liu, Ma, & Wong, 2000; Shaharanee et al., 2011), and many

rules were discarded so there were no rules left to cover the rarely occurring class value '>50K'.

The differences between the final number of rules for both Apriori (with *min_sup & min_conf*) and Apriori (with *min_sup*), in each of the Iris, Mushroom and Adult datasets are due to the sequence of the evaluation process as mentioned earlier. For the final rule sets obtained from the Iris, Mushroom and Adult datasets, the Apriori (with *min_sup*) approach achieved higher accuracy, which again confirms our earlier suggestion to apply the *confidence* constraint after the contradictive rules have been removed. In all cases, the Apriori (with *min_sup*) approach removed a contradictive rule that remained in Apriori (with *min_sup & min_conf*). In the Iris dataset the contradictive rule removed by Apriori (with *min_sup*) during the contradictive assessment was "Petal_length(4.54 - 5.72) ==> Class(Virginica)" (*confidence*: 70%, *support*: 23.33%), because it contradicted another rule "Petal_length(4.54 - 5.72) ==> Class(Versicolour)" (*confidence*: 30%, *support*: 10%). In the Apriori (with *min_sup & min_conf*) approach the contradictive rule "Petal_length(4.54 - 5.72) ==> Class(Virginica)" has a classification accuracy of 70% and misclassifies 9 instances from the training set, while it has a predictive accuracy of 70.59% and misclassifies 5 instances from the testing set.

Similarly, in the Mushroom dataset the contradictive rule removed by Apriori (with *min_sup*) was "Gillsize(Broad) ==> Class(Edible)" (*confidence*: 69.16%, *support*: 48.07%), because it contradicted another rule "Gillsize(Broad) ==> Class(Poisonous)" (*confidence*: 30.84%, *support*: 21.44%). In the Apriori (with *min_sup & min_conf*) approach the contradictive rule "Gillsize(Broad) ==> Class(Edible)" has a classification accuracy of 69.16% and misclassifies 1045 instances from the training set, while it has a predictive accuracy of 70.91% and misclassifies 647 instances from the testing set.

For the Adult dataset, there are 3 extra rules in the final rule set for Apriori (with *min_sup & min_conf*) compared to the final rule set for Apriori (with *min_sup*). These 3 rules have been discarded in Apriori (with *min_sup*) due to the contradictive assessment conducted. For example; the contradictive rule "Sex(Male) ==> Class(Income<=50K)" (*confidence*: 68.62%, *support*: 46.32%), has been removed

from Apriori (with *min_sup*) because it contradicted another rule "Sex(Male) ==> Class(Income>50K): (*confidence*: 31.38%, *support*: 21.21%). In the Apriori (with *min_sup & min_conf*) approach, the contradictive rule "Sex(Male) ==> Class(Income<=50K)" has a classification accuracy of 68.62% and misclassifies 6396 instances from the training set, while it has a predictive accuracy of 69.03% and misclassifies 3134 instances from the testing set.

Table 6.14: Comparison between Apriori (*Min_Sup & Min Conf*) and Apriori (*Min_Sup*) in Iris Dataset

Type of analysis	**Data Partition**	**Apriori (*Min_Sup & Min Conf*)**			**Apriori (*Min_Sup*)**		
		# Of Rules	**AR %**	**CR%**	**# Of Rules**	**AR %**	**CR %**
Initial # of Rules	Training	51	92.86	100.00	58	81.77	100.00
	Testing		90.99	100.00		78.46	100.00
# of Rules after ST	Training	51	92.86	100.00	58	81.77	100.00
	Testing		90.99	100.00		78.46	100.00
Statistics Analysis	Training	22	88.15	100.00	29	71.60	100.00
	Testing		85.29	100.00		68.07	100.00
Redundancy Removal	Training	22	88.15	100.00	29	71.60	100.00
	Testing		85.29	100.00		68.07	100.00
Contradictive Removal	Training	22	88.15	100.00	21	89.79	100.00
	Testing		85.29	100.00		86.43	100.00
Conf. 60%	Training				21	89.79	100.00
	Testing					86.43	100.00

Table 6.15: Comparison between Apriori (*Min_Sup & Min Conf*) and Apriori (*Min_Sup*) in Mushroom Dataset

Type of analysis	**Data Partition**	**Apriori (*Min_Sup & Min Conf*)**			**Apriori (*Min_Sup*)**		
		# Of Rules	**AR %**	**CR%**	**# Of Rules**	**AR %**	**CR %**
Initial # of Rules	Training	75237	94.27	100.00	77815	91.79	100.00
	Testing		94.34	100.00		83.20	100.00
# of Rules after ST	Training	653	91.63	100.00	669	89.97	100.00
	Testing		91.75	100.00		90.08	100.00
Statistics Analysis	Training	44	92.43	100.00	48	81.20	100.00
	Testing		92.51	100.00		81.06	100.00
Redundancy Removal	Training	21	91.33	100.00	24	76.97	100.00
	Testing		91.28	100.00		76.88	100.00
Contradictive Removal	Training	21	91.33	100.00	20	94.62	100.00
	Testing		91.28	100.00		94.24	100.00
Conf. 60%	Training				20	94.62	100.00
	Testing					94.24	100.00

Table 6.16: Comparison between Apriori (*Min_Sup & Min Conf*) and Apriori (*Min_Sup*) in Adult Dataset

Type of analysis	**Data Partition**	**Apriori (*Min_Sup & Min Conf*)**			**Apriori (*Min_Sup*)**		
		# Of Rules	**AR %**	**CR%**	**# Of Rules**	**AR %**	**CR %**
Initial # of Rules	Training	1680	81.23	100.00	2192	68.98	100.00
	Testing		81.35	100.00		69.05	100.00
# of Rules after ST	Training	233	80.46	100.00	303	67.46	100.00
	Testing		80.50	100.00		67.45	100.00
Statistics Analysis	Training	71	81.49	100.00	107	63.83	100.00
	Testing		81.65	100.00		63.87	100.00
Redundancy Removal	Training	46	85.46	100.00	58	69.65	100.00
	Testing		85.61	100.00		69.72	100.00
Contradictive Removal	Training	46	85.46	100.00	48	81.79	99.98
	Testing		85.61	100.00		81.91	99.95
Conf. 60%	Training				43	88.31	96.38
	Testing					88.41	96.12

6.2.6 Apriori vs. Maximal vs. Closed

From the previous section, the Apriori with *min_sup* constraint approach demonstrated that, in general, it achieves a better rule set than Apriori with both *min_sup* and *min_conf* constraint because it applies a *confidence* constraint after any contradictive rules have been removed. **Thus, here and in later sections, the term 'Apriori' is used when referring to the Apriori with *min_sup***. In this section, the Apriori approach is compared with the results obtained using the Closed and Maximal frequent pattern mining algorithms. As discussed earlier, the closed patterns are those for which no other super-pattern has the same *support*, while a maximal pattern has no super-pattern that is frequent. Hence, the Closed and Maximal rule sets obtained are a subset of rule sets obtained from Apriori, while the Maximal rule set is a subset of the Closed rule set, i.e. $\text{Maximal} \subseteq \text{Closed} \subseteq \text{Apriori}$. Hence, in terms of the characteristics of the rule sets as a whole, Apriori rule set will be most specific, followed by Closed and then Maximal. However, at the individual rule level, another important difference between the rule sets corresponds to the method's preference for longer/shorter rules. A longer rule will consist of more constraints in the precedent of a rule, and in this context we will refer to it as being more specific, while a shorter rule (fewer constraints in precedent) is then considered more general. The Apriori approach contains a complete set of possible rules, i.e., both general and more specific rules, while both Closed and Maximal, will by

definition contain fewer general (shorter) rules in comparison to Apriori. One would expect that the difference between the Maximal and Closed patterns, when used for classification tasks, is that because of their characteristics, the Maximal rule set will also contain fewer general rules in comparison to the Closed rule set which may contain more subsets of the maximal pattern based rules as long as they have a different *support* value.

However, as previously mentioned, the large volume of rules initially detected by all of the three approaches can still be insignificant and uninteresting. With the application of feature subset selection application, a proper statistical evaluation, and redundancy and contradictive assessment methods, this large volume of rules can be reduced and at the same time the quality of the reduced rule set is preserved well. Table 6.17 shows the comparison between the three association rule mining algorithms using the Wine dataset. The comparisons are based on the evaluation process of the rules with the applications of Symmetrical Tau, statistical analysis, redundant and contradictive assessment methods; and a specific minimum *confidence* threshold. The AR and CR values are revealed to evaluate the performance of the reduced rule sets (reflecting significant rules in the proposed framework).

Table 6.17: Comparison between Apriori, Maximal and Closed for Wine dataset

Type of analysis	**Data Partition**	**Apriori**			**Maximal**			**Closed**		
		# Rules	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**
Initial # of Rules	Training	272	76.83	100.00	103	81.09	100.00	242	73.59	100.00
	Testing		69.68	100.00		76.85	100.00		68.16	100.00
# of Rules after ST	Training	217	74.26	100.00	84	83.39	100.00	196	75.21	100.00
	Testing		68.00	100.00		79.14	100.00		69.54	100.00
Statistics Analysis	Training	24	64.16	100.00	20	76.62	100.00	21	69.16	100.00
	Testing		60.46	100.00		72.12	100.00		65.18	100.00
Redundancy Removal	Training	23	63.52	100.00	19	76.12	100.00	20	68.52	100.00
	Testing		60.05	100.00		71.80	100.00		64.77	100.00
Contradictive Removal	Training	16	85.63	100.00	19	76.12	100.00	16	82.90	100.00
	Testing		81.94	100.00		71.80	100.00		77.35	100.00
Conf. 60%	Training	15	87.84	100.00	16	85.07	100.00	14	88.61	100.00
	Testing		84.77	100.00		81.98	100.00		84.86	100.00

Let us compare the progressive rules reduction for the different algorithms with their AR and CR values. The Maximal algorithm offers a smaller volume in the initial rule set to be evaluated. This situation occurs because the algorithm itself eliminates a non-maximal itemset. The Apriori algorithm produced the largest volume of rules to be evaluated. This is due to the enumeration of every single frequent itemset

.

The experiment in Section 6.2.5 (refer to Table 6.11 for Wine dataset) demonstrated that the final 15 rules from Apriori with *min_sup* are the significant rules based on the highest AR and CR. However, as the Maximal and Closed frequent itemsets (refer to Table 6.17) were included in the experiment, the results reveal that the AR value for the Closed rule set is higher than the Apriori rule set, while at the same time the CR value settled at 100% and the rule set reduced from 15 to 14 rules.

On the final 15 rules derived from the Apriori algorithm and 16 rules from the Maximal algorithm in Table 6.17, it was found that 14 of the rules are the same to the final rules in the Closed rule set. Table 6.18 exhibits one rule that appeared in the Apriori and Maximal rule sets but not in the Closed rule set. The rule in Table 6.18 did not appear in the Closed rule set because initially there exists a superset rule that has the same frequency of 10.28%, i.e. "Flavanoids(2.24 - 3.18) & Nonflavanoids(0.24 – 3.18) & Magnesium(88.4 - 106.8) ==> Class(Low)" (*confidence* = 84.6%). However, this superset rule has been removed from the Closed rule set after the statistical analysis was applied, because the 'Nonflavanoids' attribute was determined as insignificant. This explains the slightly higher AR for the final Closed rule set in comparison to the Apriori and Maximal rule sets, since the rule from Table 6.18 has a classification accuracy of 64.10% as it misclassifies 14 instances in the training dataset and has a predictive accuracy of 60% as it misclassifies 10 instances from the testing dataset.

Table 6.18: Rule Discovered in Apriori and Maximal but Not in Closed

Conf.	**Support**	**Rule**
73.33 %	10.28 %	Flavanoids(2.24 - 3.18) & Magnesium(88.4 - 106.8) ==> Class(Low)

Additionally, another rule (see Table 6.19) that remained in Maximal rule set is discarded from the Closed rule set because the rule failed the contradictive

assessment in the Closed rule set, as it contradicted another rule “Flavanoids(2.24 - 3.18) ==> Class(Middle)” (*confidence*: 35.90, *support*: 13.08). These are the same contradictive rules that were discarded by the Apriori (*min_sup*) approach but not the Apriori (*min_sup &min_conf*) approach because it did not have the rule with smaller *confidence*, as discussed in the previous section. All of the approaches compared in this section do not initially utilize the *confidence* constraint.

It is because of the properties of the Maximal patterns that the rule “Flavanoids(2.24 - 3.18) ==> Class(Middle)” did not appear in the Maximal rule set for the contradiction to be detected. The initial rule set detected by the *Maximal* approach contained the superset of this rule, namely “Flavanoids(2.24 - 3.18) & ColorIntensity(low - 3.62) ==> Class(Middle)” (*confidence*: 92.86%, *support*: 12.15%). This rule “Flavanoids(2.24 - 3.18) & ColorIntensity(low - 3.62) ==> Class(Middle)” has been removed from the Maximal rule set after the redundancy assessment as another subset of the rule had a higher *confidence* value, i.e. “ColorIntensity(low - 3.62) ==> Class(Middle)” (*confidence*: 97.06%, *support*: 30.84%). Note that one would not expect that both “Flavanoids(2.24 - 3.18) & ColorIntensity(low - 3.62) ==> Class(Middle)” and. “ColorIntensity(low - 3.62) ==> Class(Middle)” rules exist together within the Maximal pattern, as the second rule is the subset of the first. However, this was not initially the case as the second rule “ColorIntensity(low - 3.62) ==> Class(Middle)” is a simplified form of the initial rule “Hue(0.86 – 1.25) & ColorIntensity(low - 3.62) ==> Class(Middle)” as the variable (Hue) was detected insignificant based on the statistical analysis and removed from the precedent of the rule.

Table 6.19: Rule Discovered in Maximal but Not in Closed and Apriori

Conf.	Support	Rule
64.10 %	23.36 %	Flavanoids(2.24 - 3.18)==> Class(Low)

Given the unique criteria for each algorithm and based on the unification techniques in evaluating the rules, the results show that the rule set from the Maximal algorithm contains only the most specific rules (i.e. contain more attribute constraints as they prefer longest (Maximal) patterns) while it does not have the more general rules (i.e. subsets of longer rules with less attribute constraints) as do the rule sets from Closed

and Apriori algorithms. Consequently, the AR values in the Maximal rule set are lower compared to those of both the Apriori and Closed rule sets, (except for Mushroom dataset where the accuracy is slightly higher in comparison to Closed). This is especially the case for the testing set as less specific (contains fewer attribute constraints) rules usually perform better on unseen data than do the more specific rules.

As previously mentioned, the large volume of rules in the Wine dataset initially detected by all of the three approaches can still be insignificant and uninteresting. As seen from the results, with a proper feature subset selection application, statistical evaluation, and redundancy and contradictive assessment methods, this large volume of rules can be reduced and at the same time the quality of the reduced rule set is well preserved. Thus finally, an optimal rule set of 15, 16 and 14 are discovered using the Apriori, Maximal and Closed algorithms, respectively as shown in Table 6.20.

Table 6.20: Rules for Wine dataset based Apriori, Maximal and Closed

Apriori	**Maximal**	**Closed**
Flavanoids(low - 1.29) ==> Class(High)	ColorIntensity(low - 3.62) ==> Class(Middle)	Flavanoids(3.18-4.13) ==> Class(Low)
Magnesium(106.8- 125.2) ==> Class(Low)	Flavanoids(3.18-4.13) ==> Class(Low)	ColorIntensity(5.97 - 8.31) ==> Class(Low)
ColorIntensity(5.97 - 8.31) ==> Class(Low)	Flavanoids(low - 1.29) ==> Class(High)	ColorIntensity(low - 3.62) ==> Class(Middle)
Flavanoids(3.18-4.13) ==> Class(Low)	Flavanoids(1.29 - 2.24) ==> Class(middle)	Flavanoids(low - 1.29) ==> Class(High)
ColorIntensity(low - 3.62) ==> Class(Middle)	Magnesium(low - 88.4) ==> Class(Middle)	Flavanoids(1.29 - 2.24) ==> Class(Middle)
Magnesium(low - 88.4) ==> Class(Middle)	ColorIntensity(5.97 - 8.31) ==> Class(Low)	Magnesium(low - 88.4) ==> Class(Middle)
Flavanoids(1.29 - 2.24) ==> Class(Middle)	Flavanoids(2.24 - 3.18)==> Class(Low)	Magnesium(106.8- 125.2) ==> Class(Low)
Flavanoids(low - 1.29) & Magnesium(88.4 - 106.8) ==> Class(High)	Magnesium(106.8- 125.2) ==> Class(Low)	Magnesium(106.8- 125.2) & Flavanoids(2.24 - 3.18)==> Class(Low)
Flavanoids(low - 1.29) & ColorIntensity(3.62 - 5.97) ==> Class(High)	Magnesium(106.8- 125.2) & Flavanoids(2.24 - 3.18)==> Class(Low)	Flavanoids(2.24 - 3.18)& ColorIntensity(3.62 - 5.97) ==> Class(Low)
Flavanoids(2.24 - 3.18)& ColorIntensity(3.62 - 5.97) ==> Class(Low)	Flavanoids(low - 1.29) & Magnesium(88.4 - 106.8) ==> Class(High)	ColorIntensity(low - 3.62) & Flavanoids(1.29 - 2.24) ==> Class(Middle)
Magnesium(106.8- 125.2) & Flavanoids(2.24 - 3.18)==> Class(Low)	ColorIntensity(low - 3.62) & Flavanoids(1.29 - 2.24) ==> Class(Middle)	Magnesium(low - 88.4) & ColorIntensity(low - 3.62) ==> Class(Middle)
Flavanoids(2.24 - 3.18)&	Magnesium(low - 88.4) &	ColorIntensity(low - 3.62) &

Magnesium(88.4 - 106.8) ==> Class(Low)	ColorIntensity(low - 3.62) ==> Class(Middle)	Magnesium(88.4 - 106.8) ==> Class(Middle)
Magnesium(low - 88.4) & ColorIntensity(low - 3.62) ==> Class(Middle)	ColorIntensity(low - 3.62) & Magnesium(88.4 - 106.8) ==> Class(Middle)	Flavanoids(low - 1.29) & ColorIntensity(3.62 - 5.97) ==> Class(High)
ColorIntensity(low - 3.62) & Flavanoids(1.29 - 2.24) ==> Class(Middle)	Flavanoids(low - 1.29) & ColorIntensity(3.62 - 5.97) ==> Class(High)	Flavanoids(low - 1.29) & Magnesium(88.4 - 106.8) ==> Class(High)
ColorIntensity(low - 3.62) & Magnesium(88.4 - 106.8) ==> Class(Middle)	Flavanoids(2.24 - 3.18)& ColorIntensity(3.62 - 5.97) ==> Class(Low)	
	Flavanoids(2.24 - 3.18)& Magnesium(88.4 - 106.8) ==> Class(Low)	

The overall comparison between the Apriori, Maximal and Closed algorithms using the Iris, Mushroom and Adult datasets are shown in Tables 6.21 to 6.23. With a proper combination of the evaluation process on rules discovered from the Apriori, Maximal and Closed algorithms for each datasets, an average of 89% in the number of rules have been discarded. While reducing the rules, the increase in the AR ranged from 7% to 20% in the Apriori rule set, from 1.6% to 20% in the Maximal rule set and from 10% to 22% in the Closed rule set (on both the training and the testing sets). As for the CR, the result reveals that the final set of rules covered almost all the instances in the dataset for each algorithm with the highest CR of 100% and the lowest CR of 76% in both the training and the testing sets.

From these results, one can see that the Closed algorithm achieves better AR values compared to Apriori and Maximal for the final rule set, for the Wine, Iris and Adult datasets. Moreover, in comparing the AR values for the final rule sets for the aforementioned datasets between Maximal and Closed, the result revealed that the AR from the final Maximal rule set are lower compared to the final Closed rule sets. For the Mushroom dataset, the Apriori approach achieves the best results followed by the Maximal approach. However, for the Maximal approach, the CR is not 100% and this can be explained by the fact that several more general rules with fewer attribute constraints are typically removed in the Maximal approach, when they are a subset of a more specific rule. The Closed approach still achieves 100% CR in its final rule set but has a smaller AR. The Apriori approach has 20 rules in its final rule set and the additional rules are what achieves the better AR overall.

From these results, one cannot draw a firm conclusion about using closed patterns for classification, as even though they performed better in three datasets, the Apriori approach which discovered all patterns had a better accuracy for the Mushroom dataset. However, one could infer that the Closed approach is a better approach than the Maximal algorithm for classification tasks because even when the Maximal approach achieved a slightly better accuracy for the Mushroom dataset, it was at the cost of a small reduction in the CR. While the Maximal approach removes a pattern if a frequent superset of that pattern exists, the Closed approach patterns will retain a pattern as long as it has a different *support* to that of another super-pattern. In the statistical analysis, redundancy and contradictive assessment methods approaches for evaluating the significance of rules, having these additional patterns can be preferable as, for example, a contradictive rule will be detected. This was the case for the rule in Table 6.19 which remained in the final rule set of the Maximal approach, but was detected as contradictive by both the Apriori and Closed approaches.

Table 6.21: Comparison between Apriori, Maximal and Closed for Iris dataset

Type of analysis	**Data Partition**	**Apriori**			**Maximal**			**Closed**		
		# Rules	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**
Initial # of Rules	Training	58	81.77	100.00	13	79.63	86.67	41	80.24	97.78
	Testing		78.46	100.00		73.75	76.67		76.21	100.00
# of Rules after ST	Training	58	81.77	100.00	13	79.63	86.67	41	80.24	97.78
	Testing		78.46	100.00		73.75	76.67		76.21	100.00
Statistics Analysis	Training	29	71.60	100.00	13	79.63	86.67	28	75.40	97.78
	Testing		68.07	100.00		73.75	76.67		71.03	100.00
Redundancy Removal	Training	21	89.79	100.00	13	79.63	86.67	24	83.16	97.78
	Testing		86.43	100.00		73.75	76.67		77.08	100.00
Contradictive Removal	Training	21	89.79	100.00	12	90.91	86.67	22	90.97	97.78
	Testing		86.43	100.00		85.71	76.67		86.54	100.00
Conf. 60%	Training	21	89.79	100.00	12	90.91	86.67	22	90.97	97.78
	Testing		86.43	100.00		85.71	76.67		86.54	100.00

Table 6.22: Comparison between Apriori, Maximal and Closed for Mushroom dataset

Type of analysis	Data Partition	Apriori			Maximal			Closed		
		# Rules	AR %	CR %	# Rules	AR %	CR %	# Rules	AR %	CR %
Initial # of Rules	Training	77815	91.79	100.00	255	80.56	100.00	3009	75.00	100.00
	Testing		83.20	100.00		80.37	99.94		75.30	100.00
# of Rules after ST	Training	669	89.97	100.00	31	88.94	98.26	260	79.61	100.00
	Testing		90.08	100.00		88.56	98.12		79.69	100.00
Statistics Analysis	Training	48	81.20	100.00	16	88.42	98.89	24	77.48	100.00
	Testing		81.06	100.00		89.02	98.71		77.54	100.00
Redundancy Removal	Training	24	76.97	100.00	13	91.61	98.89	16	78.21	100.00
	Testing		76.88	100.00		91.85	98.71		78.34	100.00
Contradictive Removal	Training	20	94.62	100.00	13	91.61	98.89	14	89.79	100.00
	Testing		94.24	100.00		91.85	98.71		89.88	100.00
Conf. 60%	Training	20	94.62	100.00	13	91.61	98.89	14	89.79	100.00
	Testing		94.24	100.00		91.85	98.71		89.88	100.00

Table 6.23: Comparison between Apriori, Maximal and Closed for Adult dataset

Type of analysis	Data Partition	Apriori			Maximal			Closed		
		# Rules	AR %	CR %	# Rules	AR %	CR %	# Rules	AR %	CR %
Initial # of Rules	Training	2192	68.98	100.00	129	62.98	99.87	1866	68.54	100.00
	Testing		69.05	100.00		63.05	99.87		68.60	100.00
# of Rules after ST	Training	303	67.46	100.00	34	67.23	98.34	257	67.06	100.00
	Testing		67.45	100.00		67.18	98.25		67.02	100.00
Statistics Analysis	Training	107	63.83	100.00	24	81.88	97.39	77	61.57	100.00
	Testing		63.87	100.00		82.18	97.28		61.57	100.00
Redundancy Removal	Training	58	69.65	100.00	21	80.92	97.39	41	67.45	100.00
	Testing		69.72	100.00		81.16	97.28		67.51	100.00
Contradictive Removal	Training	48	81.79	99.98	21	80.92	97.39	33	79.29	99.99
	Testing		81.91	99.95		81.16	97.28		79.42	99.97
Conf. 60%	Training	43	88.31	96.38	19	87.28	96.37	27	89.09	98.01
	Testing		88.41	96.12		87.48	96.10		89.25	98.08

6.2.7 Minimum Confidence Effects

This section presents a set of experiments conducted to show that the performance of the AR and CR can vary by altering the value of minimum *confidence*. This was done for the three different algorithms on the Wine dataset (Refer to Table 6.24). By increasing the minimum *confidence* from 60% to 70%, the remainder of the rules in the Apriori and Maximal rule sets were 13 identical rules. The CR values in the training set remained stable at 100%, while the AR increased to 92.03. For the testing dataset, while there was an increase in the AR, the CR values decreased. This occurs because the 13 rules in the Apriori and Maximal rule sets failed to capture all of the

instances in this dataset. For the Closed algorithm, another 2 rules were discarded from the 14 rules by increasing the minimum *confidence* to 70%. As the AR values in the training and testing sets increased, the CR values in both partitions decreased by about 1% and 3 % respectively. As the *confidence* thresholds gradually increased to 70%, 80%, 90% and 100%, the number of rules in the rule sets became smaller and identical, which led to the increase in AR but at the cost of decreasing the number of instances covered by the rules.

The changes in *confidence* values have a direct impact on the size of the rule set, AR and CR values. For example, the initial rules in Table 6.17 (Refer to Section 6.2.6) were too general and large, and may lack specificity. Hence the AR was low, but as they were reduced to only those rules in Table 6.20, the AR improved and the CR remained at 100%. Progressively increasing the minimum *confidence* threshold results in an even smaller set of rules which are more accurate but the CR suffers (Table 6.24). Thus, it is essential to determine the trade-off between finding a rule set with optimal values of AR and CR. This agrees with (Wang, Dillon, & Chang, 2002) who assert the need for balancing these conflicting regularization parameters.

Table 6.24: Minimum *Confidence* Effect on Rule Discovered in Apriori, Maximal and Closed for Wine dataset

Type of analysis	**Data Partition**	**Apriori**			**Maximal**			**Closed**		
		# Rules	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**
Conf. 60%	Training	15	87.84	100.00	16	85.07	100.00	14	88.61	100.00
	Testing		84.77	100.00		81.98	100.00		84.86	100.00
Conf. 70%	Training	13	92.03	100.00	13	92.03	100.00	12	93.22	99.07
	Testing		89.60	98.59		89.60	98.59		90.06	97.18
Conf. 80%	Training	11	95.19	99.07	11	95.19	99.07	11	95.19	99.07
	Testing		90.14	97.18		90.14	97.18		90.14	97.18
Conf. 90%	Training	9	98.04	85.98	9	98.04	85.98	9	98.04	85.98
	Testing		91.26	83.10		91.26	83.10		91.26	83.10
Conf. 100%	Training	6	100.00	58.88	6	100.00	58.88	6	100.00	58.88
	Testing		92.98	53.52		92.98	53.52		92.98	53.52

Tables 6.25 to 6.27 show the effect of altering the minimum *confidence* of rules obtained from other datasets. Such results are in agreement with those of (Do, Hui, & Fong, 2005) who stated that a rule with a high *confidence* value implies an accurate prediction. However, as shown in Tables 6.24 to 6.27, even though the AR increased

simultaneously with the increment of minimum *confidence* values, the CR values decreased as a result. This depicted the trade-off in choosing the suitable minimum *confidence* threshold for each dataset or domain considered. For example, in the Mushroom dataset (see Table 6.26), it appears that for the best results of the Apriori algorithm, the *confidence* could have been safely set up to 80%, without a loss in coverage rate.

Table 6.25: Minimum *Confidence* Effect on Rule Discovered in Apriori, Maximal and Closed for Iris dataset

Type of analysis	**Data Partition**	**Apriori**			**Maximal**			**Closed**		
		# Rules	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**
Conf. 60%	Training	21	89.79	100.00	12	90.91	86.67	22	90.97	97.78
	Testing		86.43	100.00		85.71	76.67		86.54	100.00
Conf. 70%	Training	19	92.91	100.00	11	93.28	84.44	20	94.37	97.78
	Testing		93.23	100.00		89.09	75.00		93.85	100.00
Conf. 80%	Training	17	94.76	100.00	9	98.89	70.00	18	96.47	97.78
	Testing		95.98	100.00		100.00	55.00		96.89	100.00
Conf. 90%	Training	14	97.25	94.44	9	98.89	70.00	17	97.50	94.44
	Testing		97.89	88.33		100.00	55.00		97.96	88.33
Conf. 100%	Training	9	100.00	74.44	8	100.00	62.22	12	100.00	74.44
	Testing		100.00	71.67		100.00	48.33		100.00	71.67

Table 6.26: Minimum *Confidence* Effect on Rule Discovered in Apriori, Maximal and Closed for Mushroom dataset

Type of analysis	**Data Partition**	**Apriori**			**Maximal**			**Closed**		
		# Rules	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**
Conf. 60%	Training	20	94.62	100.00	13	91.61	98.89	14	89.79	100.00
	Testing		94.24	100.00		91.85	98.71		89.88	100.00
Conf. 70%	Training	20	94.62	100.00	12	97.51	98.36	13	94.01	100.00
	Testing		94.24	100.00		97.23	98.40		93.66	100.00
Conf. 80%	Training	19	95.84	100.00	12	97.51	98.36	12	95.69	99.47
	Testing		95.51	100.00		97.23	98.40		95.40	99.69
Conf. 90%	Training	15	98.15	99.47	11	98.43	97.23	9	98.15	97.23
	Testing		97.67	99.69		98.06	97.14		97.71	97.14
Conf. 100%	Training	8	100.00	85.86	7	100.00	85.35	5	100.00	68.20
	Testing		100.00	85.88		100.00	85.17		100.00	67.08

Table 6.27: Minimum *Confidence* Effect on Rule Discovered in Apriori, Maximal and Closed for Adult dataset

Type of analysis	Data Partition	Apriori			Maximal			Closed		
		# Rules	AR %	CR %	# Rules	AR %	CR %	# Rules	AR %	CR %
Conf. 60%	Training	43	88.31	96.38	19	87.28	96.37	27	89.09	98.01
	Testing		88.41	96.12		87.48	96.10		89.25	98.08
Conf. 70%	Training	41	89.63	93.78	19	87.28	96.37	27	89.09	98.01
	Testing		89.75	93.44		87.48	96.23		89.25	98.08
Conf. 80%	Training	38	90.61	90.45	17	91.55	75.58	25	91.84	82.35
	Testing		90.72	90.05		91.71	74.73		91.97	81.84
Conf. 90%	Training	21	96.16	53.50	10	95.66	51.24	15	96.83	49.23
	Testing		96.00	53.90		95.59	51.57		96.71	49.20
Conf. 100%	Training	0	-	-	0	-	-	0	-	-
	Testing		-	-		-	-		-	-

6.2.7.1 Choosing the Confidence Threshold based on the AR and CR

As just discussed and illustrated in Tables 6.24-6.27, restricting the rule sets according to the minimum *confidence* values impacts on the trade-off between accuracy and coverage rates. Experiments show that, the AR increase simultaneously with the increase of the confidence values. However at some stages, too many rules will be discarded which significantly make the coverage rate suffer. In the previous experiment in Section 6.2.6, when the statistical analysis, and redundancy and contradictive assessment methods were utilized for the rules, there is no significant reduction of CR with the reduced number of rules until the confidence parameter is included. It is important in this framework to monitor the CR in reducing the number of rules and to identify the break point/right time at which to stop reducing the number of rules (increasing the confidence values). Thus, in this experiment, the stopping points are chosen at which a dramatic reduction of CR occurred. Choosing an optimal stopping point based on the confidence threshold will ensure that the reduction of rules can be stopped before reaching an unacceptable level of coverage.

For example in Figure 6.6 and Figure 6.7, for both Wine and Adult datasets for each of the Apriori, Maximal and Closed rule sets, the cut-off value to stop reducing the number of rules is at 90% confidence, as it was found that the CR drops dramatically when the confidence threshold is increased to 90%. Additionally, at 100% confidence threshold for Adult data (Figure 6.7), there are no rules that satisfy the given threshold.

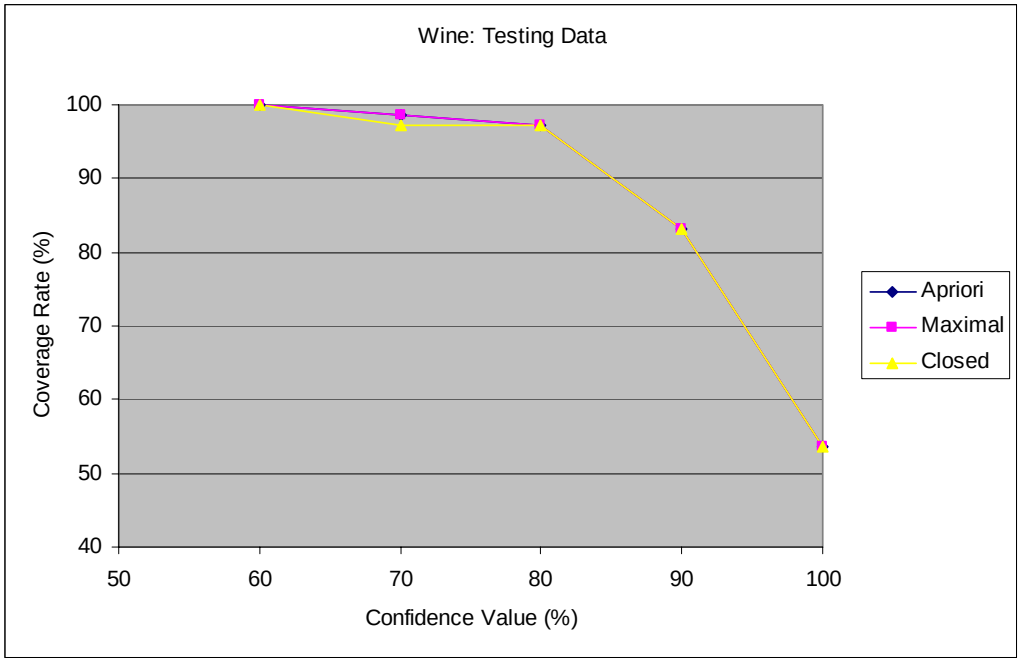


Figure 6.6: Confidence Value vs. Coverage Rate on Wine for Testing Dataset

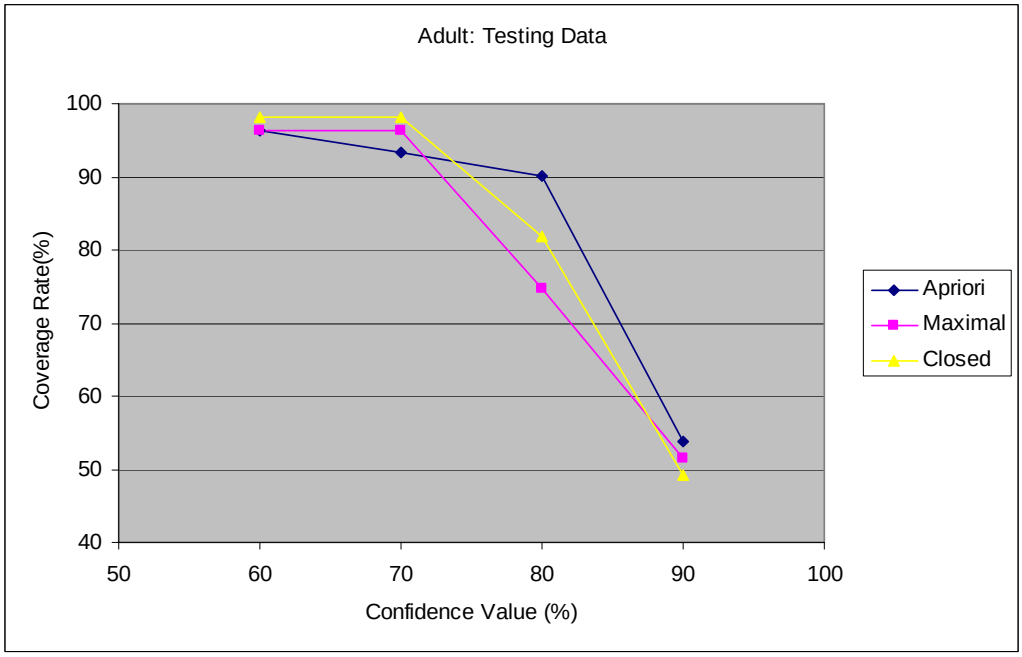


Figure 6.7: Confidence Value vs. Coverage Rate on Adult for Testing Dataset

As for the Mushroom data in Figure 6.8, the graph demonstrated that there is a significant drop of coverage rate as the confidence value is increased to 100%. This occurred for each of the Apriori, Maximal and Closed frequent itemsets rules. Finally, for the Iris dataset (Figure 6.9), the confidence break point at which the

coverage rate dropped significantly is 90% for both Apriori and Closed, and 80% for Maximal.

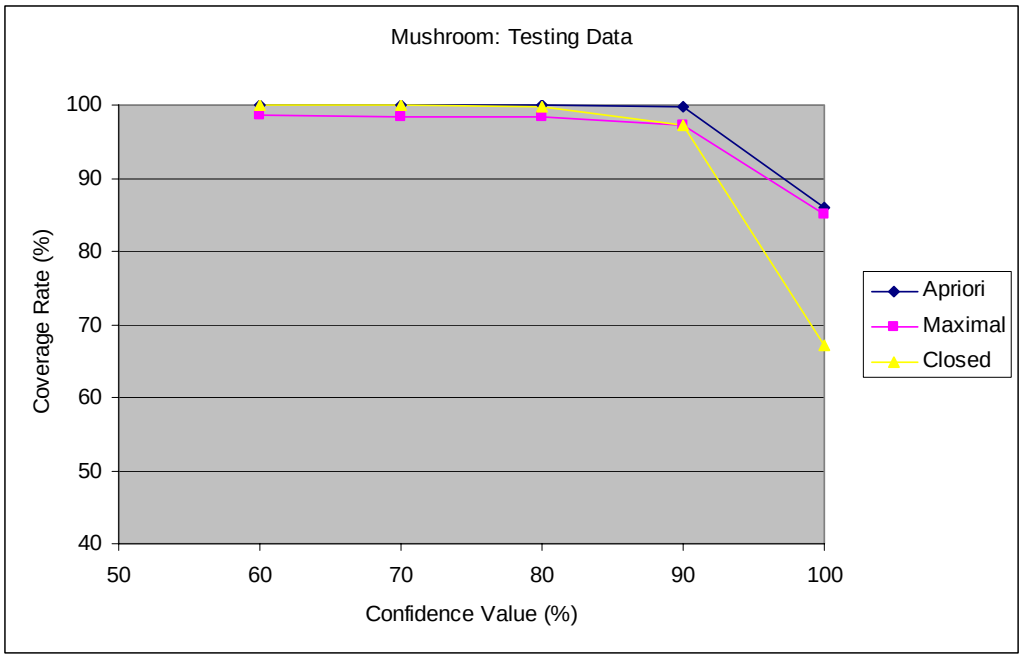


Figure 6.8: Confidence Value vs. Coverage Rate on Mushroom for Testing Dataset

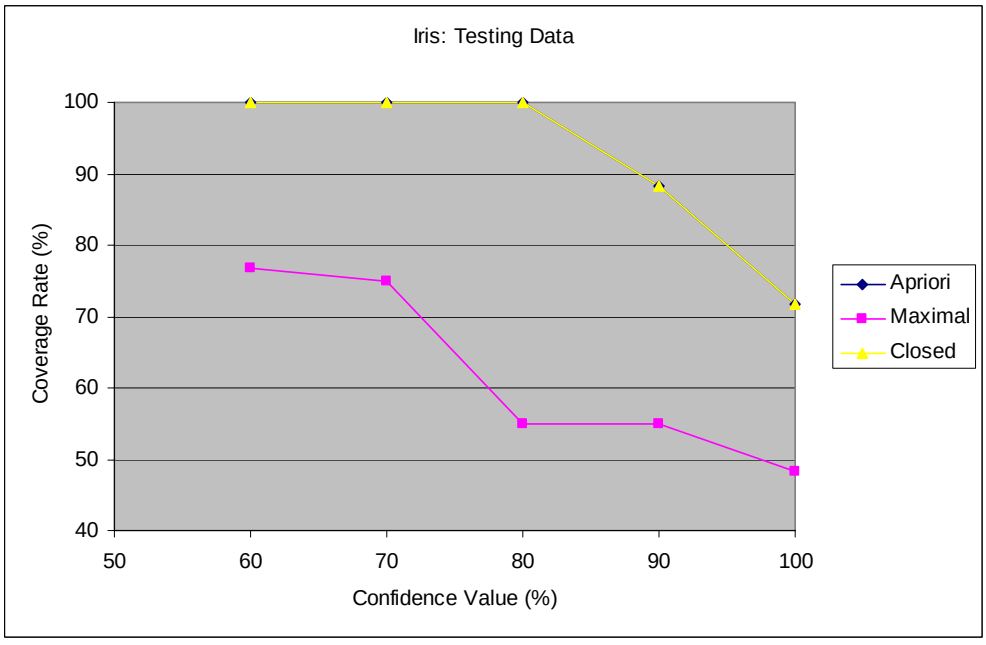


Figure 6.9: Confidence Value vs. Coverage Rate on Iris for Testing Dataset

While the confidence cut-off value may differ for each dataset and for each Apriori, *Maximal* and *Closed* frequent itemsets, this may indicate that the choice of optimal

confidence break points will depend on the type of data, application and business involved. In some cases, such as in a critical domain, one may prefer to have a strict cut-off value or higher confidence rate to guarantee an accurate assessment and results. For example, they may prefer to use only high confidence rules for decision support in the cases covered by the rule set, and would prefer not to use lower confidence rules for decision support in the remaining cases.

6.3 Conclusion

This chapter presented a framework for evaluating the rules discovered from three association rule mining algorithms, namely the Apriori, Maximal and Closed algorithms. The quality of the rules discovered was measured using statistical analysis, and redundancy and contradictive assessment methods.

Initially, two variants of the Apriori algorithm were evaluated. The first variant corresponded to the standard Apriori algorithm with both *support* and *confidence* thresholds, while the second variant was constrained using only the minimum *support* threshold. Rules were then verified in order to determine their validity and interestingness. The results show that it is more advantageous to remove the rules that failed the statistical test, the redundant rules, and the contradictive rules in the initial evaluating process and utilize the *confidence* constraint only at the end of the process. This will result in a relatively small number of rules and at the same time allow for detection of contradictive rules as all lower confidence rules are initially considered. As demonstrated in the experiments, a drawback of applying the minimum *confidence* threshold at the start of the process is that the existence of a contradictive rule that has relatively low *confidence* will not be known. This lack of knowledge can cause an unreliable association rule to become part of the final rule set which, as demonstrated, reduces the accuracy of the rule set in comparison to when the rule was removed. Alternatively, within the rules generated from Apriori with *min_sup*, initially the two or more contradictive rules exist so all of the contradictive rules will be discarded, as the contradiction implies that they are unreliable for prediction purposes. An alternative approach would be to start with a lower *confidence* threshold to still discard those patterns where the *confidence* is not high enough for them to be considered as a significant contradiction to another rule

with much higher *confidence*. One can then progressively increase the threshold after the statistical rule validation techniques have been applied. Based on the proper rule evaluating steps in the proposed framework, the final rules from the Wine, Iris, Mushroom and Adult datasets generated using the Apriori with *min_sup* constraint are fewer in number and achieve a better classification and prediction accuracy for both the training and the testing datasets.

In the second experiment set, the rules generated from the Apriori with *min_sup*, Closed and Maximal approaches are evaluated according to the proposed framework. Experimental results show that the rules from the Closed approach generated from the Wine, Iris and Adult datasets offer a better final rule set in terms of classification and prediction accuracy, while for the Mushroom dataset, the Apriori obtains the best results. The Maximal approach, due to its preference for longer patterns, generally produces fewer general rules and only the most specific rules that satisfy the minimum support threshold. On the other hand these specific rules (contain more attributes constraints) are commonly removed in Closed and Apriori, because more general rules (less attribute constraints) existed, which through statistical/redundant/contradictive check, determined the redundancy of those specific rules. This is explained by the lowest coverage rate of the final rule sets from every dataset, especially for the testing set. Generally speaking, the coverage rate of long and specific rules (contains more attributes constrain) is expected to be weaker for unseen datasets.

In the final experiment set, the effects of the minimum confidence on the proposed framework in evaluating the rules from each of the datasets generated from all three approaches are revealed. Increasing the *confidence* threshold will gradually reduce the number of rules to those that have very high accuracy because of large *confidence*. However, as the rule sets have been reduced, more instances will not be captured by the rule set; hence, typically there is deterioration in the CR. Choosing smaller *confidence* thresholds will result in larger sets of rules that may lack generalization power, thereby weakening the AR performance, but are capable of covering more instances. Alternatively, choosing relatively high *confidence* thresholds will result in a smaller set of rules, thereby achieving higher AR with the trade-off of capturing fewer instances. Hence, one needs to carefully monitor the

effect of increasing the confidence thresholds. As indicated by the experiment, while the AR improved with the increase of the confidence threshold, conversely there is significant loss of CR. Thus, by properly defining and providing a certain optimal break point of the confidence threshold, at some point, an acceptable level of AR and CR will be achieved. Thus, it is important to balance the trade-off between AR and CR in order to determine the optimal value for the minimum *confidence* threshold, which may differ depending on the sensitivity of the domain at hand.

Generally speaking, the experimental results have demonstrated that the proposed framework is capable of reducing a large number of non-significant and redundant rules while simultaneously preserving a relatively high level of accuracy. The experiments have also revealed the important differences between frequent, closed and maximal patterns, when used for classification tasks, and the effect of the *confidence* threshold and the difference when utilized at different stages of the rule filtering process.

References

Blanchard, J., Guillet, F., Gras, R., & Briand, H. (2005). Using Information-Theoretic Measures to Assess Association Rule Interestingness. In *Proceedings of the 5th IEEE International Conference on Data Mining*: IEEE Computer Society.

Brijs, T., Vanhoof, K., & Wets, G. (2003). Defining interestingness for association rules. *International Journal of Information Theories and Applications, 10(4)*, 370-376.

Do, T. D., Hui, S. C., & Fong, A. C. M. (2005). Prediction confidence for associative classification. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on* (Vol. 4, pp. 1993-1998 Vol. 1994).

Frank, A., & Asuncion, A. (2010). {UCI} Machine Learning Repository. In: University of California, Irvine, School of Information and Computer Sciences.

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

Liu, B., Ma, Y., & Wong, C. (2000). Improving an Association Rule Based Classifier. In D. Zighed, J. Komorowski & J. Zytkow (Eds.), *Principles of Data Mining and Knowledge Discovery* (Vol. 1910, pp. 293-317): Springer Berlin / Heidelberg.

Shaharanee, I. N. M., Dillon, T. S., & Hadzic, F. (2009). Ascertaining Association Rules Using Statistical Analysis. In P. S. Sandhu (Ed.), *2009 International Symposium on Computing, Communication and Control (ISCCC 2009)* (pp. 180-188). Singapore: IACSIT

Shaharanee, I. N. M., Hadzic, F., & Dillon, T. (2009). Interestingness of Association Rules Using Symmetrical Tau and Logistic Regression. In A. Nicholson & X. Li (Eds.), *AI 2009* (Vol. 5866, pp. 442-431): LNAI.

Shaharanee, I. N. M., Hadzic, F., & Dillon, T. S. (2011). Interestingness measures for association rules based on statistical validity. *Knowledge-Based Systems, 24*, 386-392.

Wang, D., Dillon, T., & Chang, E. (2002). Trading off between Misclassification, Recognition and Generalization in Data Mining with Continuous Features. In *Developments in Applied Artificial Intelligence* (Vol. 2358, pp. 121-130): Springer Berlin / Heidelberg.

Zaki, M. J., & Hsiao, C.-J. (2002). CHARM: An Efficient Algorithm for Closed Itemset Mining. In *2nd SIAM International Conference in Data Mining*.

Zhang, C., & Zhang, S. (2001). Collecting Quality Data for Database Mining. In *AI 2001: Advances in Artificial Intelligence* (pp. 131-142).

Zhou, X. J., & Dillon, T. S. (1991). A statistical-heuristic feature selection criterion for decision tree induction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*, 834-841.

CHAPTER 7 DETAILS OF SOLUTION AND EVALUATION FOR TREE-STRUCTURED DATA

7.1 Introduction

This chapter provides the details of a framework developed for evaluating the association rules discovered from XML documents or tree-structured data in general; the evaluation and validation of the proposed framework is also presented. As described earlier in Chapter 4, the work on evaluating the association rules derived from XML documents was extended from the initial framework intended to evaluate the association rules from relational data. The main motivation of this thesis is to examine how statistical analysis, redundancy and contradictive assessment methods can be employed, and to develop a proper sequence of use of these techniques to arrive at a more reliable and interesting set of rules. However, the focus here is on the association rules discovered from semi-structured or tree-structured data such as XML. This chapter starts with an overview of tree-structured data, and the issues surrounding the modelling of tree structured-data are discussed in Section 7.2.

A general description of frequent subtree mining and the IMB3 algorithm for frequent subtree discovery are provided in Section 7.3. In this chapter, the initial focus is on evaluating frequent subtrees generated using the IMB3 Miner algorithm (Tan, Dillon, Hadzic, Chang, & Feng, 2006). This frequent subtree mining algorithm is characterized by adopting a Tree Model Guided (TMG) candidate generation (Tan, Hadzic, Dillon, Chang, & Feng, 2008).

As the experiment using the IMB3 algorithm was expanded to include more complex tree-structured data, the statistical analysis to determine irrelevant rules was difficult to be applied directly as the measures themselves do not take structural aspects of data into account. As demonstrated in evaluating association rules from relational data (Chapter 6), a flat representation allows application of standard statistical analysis, and redundancy and contradictive assessment methods. Thus, by having a tree structured data effectively represented in a flat format while preserving the structural characteristics, this will allow the direct application of statistical analysis, and redundancy and contradictive assessment methods to evaluate the frequent-

subtree-based rules. (Hadzic, 2011) developed a method to represent and preserve tree-structured data in a flat format by using a *Database Structure Model* (*DSM*). This method was developed to enable a wider range of data mining/analysis techniques to be directly applied to tree-structured data, as well as to handle complexity issues that arise when dealing with complex tree structrues. Section 7.4 provides a detailed explanation and examples of tree-structured data represented in such a structure-preserving flat format.

Section 7.5 provides a general description of the evaluation process for the proposed framework, while Subsection 7.5.1 describes in detail the characteristics of each dataset used in each experiment. The evaluation of the proposed framework is undertaken in two main sections. In the first section, the interestingness of frequent subtrees discovered using the IMB3 algorithm will be evaluated. This acts as the traditional frequent subtree mining application in discovering association rules from tree-structured data. This can be found in Section 7.6. Section 7.7 provides a new direction for the way that the frequent subtree will be evaluated. The tree-structured data format is converted into the structure-preserving flat table format and from here the evaluation is done of the frequent rules discovered from the commonly-used itemset format representation. In Section 7.7.1, the frequent rules derived with the *DSM* approach are progressively verified with statistical analysis, and redundancy and contradictive assessment methods. The effect of the inclusion/exclusion of backtrack attributes/nodes for *DSM* approach is discussed in Section 7.7.2. Moreover, the evaluation extends to the rules based on embedded subtrees and rules based on induced subtrees in Section 7.7.3. In addition to that, a comparison with a well-known structural classification approach based on frequent subtrees, namely the *XRules* classifier (Zaki & Aggarwal, 2003) is undertaken in Section 7.7.4 to measure and compare the performance of the proposed framework. The evaluation of the interestingness of frequent-subtree-based rules extracted using the *DSM* approach using a real-world dataset is presented in Section 7.7.5. The results are summarized and the chapter is concluded in Section 7.8.

7.2 Tree-Structured Data

The XML-enabled association rule as explained in Chapter 3 enables the discovery of interesting relationships in a given set of semi-structured data. An XML document has a hierarchical structure, where an element may contain further embedded elements, and a number of attributes can be attached to each element. It is therefore commonly represented as a rooted ordered labelled tree and such representation is utilized in this thesis.

String-like representation of rooted ordered labelled trees has become a well-accepted representation in the frequent subtree mining field (Chi, Muntz, Nijssen, & Kok, 2005; Tan, Dillon, Hadzic, Chang, & Feng, 2006; Tan, Hadzic, Dillon, Chang, & Feng, 2008; Zaki, 2005). This representation as reviewed by (Chi, Muntz, Nijssen, & Kok, 2005), is more compact and easy to manipulate compared to standard data structures, thus creating space efficiency.

The pre-order (depth-first) string encoding as described in (Chi et al., 2005; Zaki, 2005) is utilized in this thesis for frequent subtree mining tasks. The definition of *pre-order traversal* is taken from (Hadzic, Tan, & Dillon, 2011): If ordered tree T consists only of a root node r, then r is the pre-order traversal of T. Otherwise let T_1, T_2, ..., T_n be the subtrees occurring at r from left to right in T. The pre-order traversal begins by visiting r and then traversing all the remaining subtrees in pre-order starting from T_1 and finishing with T_n. The pre-order string encoding (Chi, et al., 2005) can be generated by adding vertex labels in a pre-order traversal of a tree, and appending a backtrack symbol (for example '-1', and '-1'$\notin$L) whenever there is backtrack from a child node to its parent node.

Figure 7.1 and Table 7.1 depict a tree database consisting of 7 tree instances (or transactions) and the string encoding for tree database, respectively:

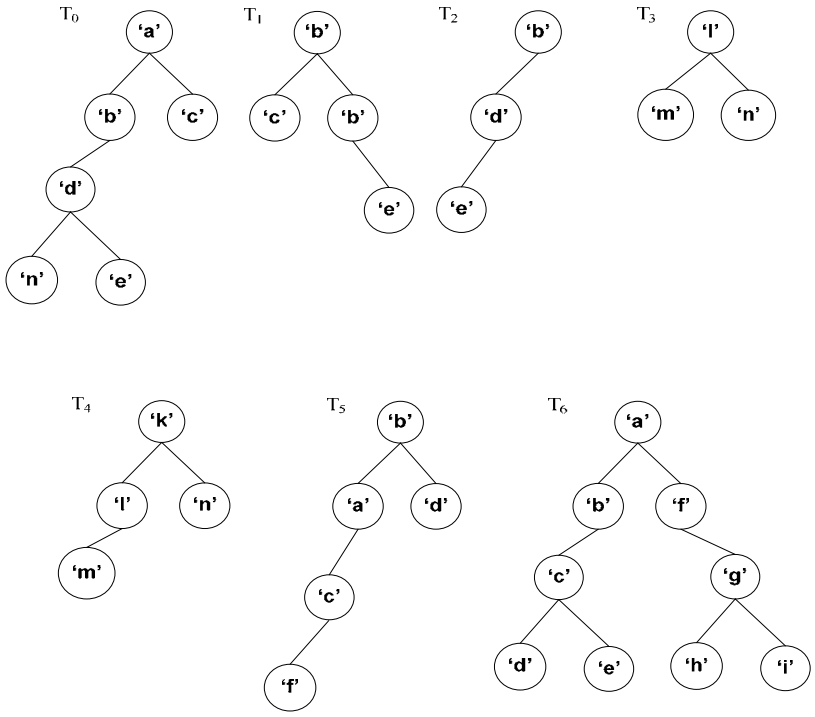


Figure 7.1: Example of a tree-structured database (*Tdb*) consisting of 7 ($T_0 - T_6$ transactions)

Table 7.1: Example of Tree Transactions

Tree Database (*Tdb*)	**Pre-order String Encoding**
T_0	'a b d n -1 e -1 -1 -1 c -1'
T_1	'b c -1 b e -1 -1'
T_2	'b d e -1 -1'
T_3	'l m -1 n -1'
T_4	'k l m -1 -1 n -1'
T_5	'b a c f -1 -1 -1 d -1'
T_6	'a b c d -1 e -1 -1 -1 f g h -1 i -1 -1 -1'

7.2.1 Modelling Tree-Structured Data

An example of three user sessions logged into the DEBII website server is depicted here to represent the process of modelling the XML documents in a tree structure format.

Session 1:
/
/research.html
/research/topics.html
/research/topics/51-business-intelligence.html
/research/topics/55-e-education-ecosystems.html
/research/seminars.html
/research/seminars/413-presentation-by-eric-feinberg.html
/phd-a-msc.html
/phd-a-msc/scholarships.html
/phd-a-msc/scholarships.html#debii
/about.html
/about/objectives.html
/about/mission-and-vision.html

Session 2:
/
/research.html
/research/centres-and-labs.html
/centres-and-labs/217-anti-spam-research-lab-asrl.html
/centres-and-labs/214-centre-for-stringology-a-applications-csa-.html
/research/jobs.html
/contact-us.html

Session3:
/
/research.html
/research/publications.html
/research/publications/conf-a-journal-papers.html
/allstaff.html
/allstaff/Research Professors & Fellows.html
/exchange-students.html
/phd-a-msc.html
/phd-a-msc/research-training.html

Figure 7.2: Example of user sessions

Table 7.2: Integer mapping for web pages from Figure7.2

ID	Web page
0	Home page
1	Research
2	Topics
3	51-business-intelligence
4	55-e-education-ecosystems
5	Seminars
6	413-presentation-by-eric-feinberg
7	phd-a-msc
8	Scholarships
9	scholarships.html#debii
10	About
11	Objectives
12	mission-and-vision
13	centres-and-labs
14	217-anti-spam-research-lab-asrl
15	214-centre-for-stringoLogsy-a-applications-csa-
16	Jobs
17	contact-us
18	Publications
19	conf-a-journal-papers
20	Allstaff
21	Research Professors & Fellows
22	research-training

In Section 7.2, a formal representation of modeling an XML document to a tree structure is provided. Table 7.2 is an example of an XML string index computed from the XML session in Figure 7.2. The mapping process from string index to integer index can be done with a hash function as discussed in (Zaki, 2005). Representing a label as an integer instead of a string label has considerable performance and space advantages (Tan et al., 2006).

As mentioned earlier, a common way of representing trees is to use the pre-order (depth-first) string encoding (φ) as described in (Zaki, 2005). For example, the pre-order string encoding representation of the underlying tree structure of the user navigation of Figure 7.2 is transformed to φ(session 1) = ‘0 1 2 3 -1 4 -1 -1 5 6 -1 -1 -1 7 8 9 -1 -1 -1 10 11 -1 12 -1 -1 ’ and φ(session 2) = ‘0 1 13 14 -1 15 -1 -1 16 -1 -1 17 -1’ and φ(session 3) = ‘0 1 18 19 -1 -1 -1 20 21 -1 -1 7 22 -1 -1’. The access sequence of web pages from Figure 7.2 can be represented in a tree-structured way as shown in Figure 7.3. The order of pages accessed is reflected by the pre-order traversal of the tree. The corresponding tree structure is more informative than just a

sequence of pages accessed as it captures the structure of the web site, and navigational patterns over this website. With this approach, specific pages can be considered within the same context.

An example of this is the two pages being grouped under the ‘centres and labs’ parent node with label 13 in the tree of session 2, and 2 pages under the ‘research’ parent node with label 1 in the tree of session 1. Session 1 has come from an IP within the university and is most likely an example of a student acquiring some general information about the institute and then seeking information related to postgraduate study. The second session came from an IP internal to the university (most likely from another university in Perth), where the user was interested in looking for jobs by browsing DEBII centres and labs, and contacted the institute for more information. While session three may come from a potential external student who is searching for a potential supervisor by browsing some related conference papers and is interested in finding a research training program.

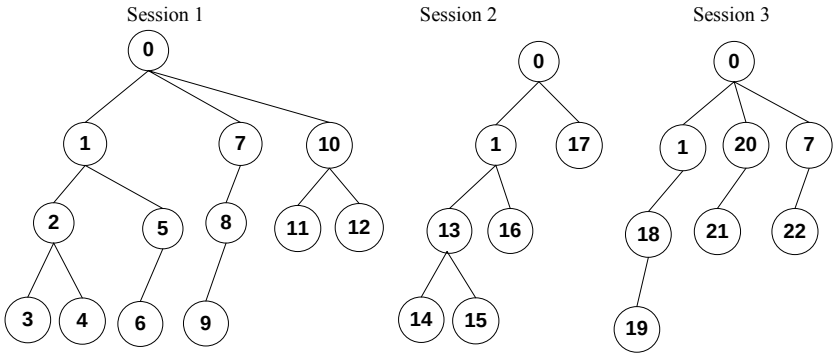


Figure 7.3: Integer-indexed tree of XML tree in Figure 7.2

The integer–indexed tree is then formatted as shown in Table 7.3. This dataset format representation was proposed by (Zaki, 2005). Please note that the second column (*cid*) could be used to refer to a specific entity which the record describes (eg. User id). However, in many domains such information is often unavailable, or it has been intentionally omitted or related through the transaction id (*tid*). Hence, in most of the tree databases represented in this format, the *cid* column will simple be a repetition of the *tid* column. This is the common format used in the frequent subtree mining field (Hadzic, Tan et al., 2011).

Table 7.3: An Integer-Indexed tree in Figure 7.3 formatted as a string-like representation as used in (Zaki, 2005). *tid*: transaction-id; *cid*: omitted (i.e. equal to *tid*); |S|: size of string

tid	***cid***	**|S|**	**Pre-order(depth-first) encoding**
0	0	25	0 1 2 3 -1 4 -1 -1 5 6 -1 -1 -1 7 8 9 -1 -1 -1 10 11 -1 12 -1 -1
1	1	13	0 1 13 14 -1 15 -1 -1 16 -1 -1 17 -1
2	2	15	0 1 18 19 -1 -1 -1 20 21 -1 -1 7 22 -1 -1
…	...	…	…
…	...	…	…

The dataset format in Table 7.3 is a comprehensive approach for frequent subtree mining algorithms. As proven by (Zaki, 2005), it involves a simple process of scanning the tree database to discover frequent associations among tree structured data objects.

7.3 Frequent Subtree Mining

As mentioned in Chapter 3 and Chapter 4, the aim in this thesis with respect to the tree- structured data is to evaluate the interestingness of frequent subtrees. In this section, there is a discussion of general issues surrounding frequent subtree mining including the candidate generation and counting, and the algorithm used in this thesis in generating the frequent subtrees. These development issues are not part of the thesis; however, there are important issues worth discussing in order to develop an understanding of how frequent subtrees are generated. As stated earlier in Chapter 4, the problem of frequent subtree mining is to find all subtrees that occur at least as many times as the user-specified support threshold. The generation of frequent subtrees, involves two steps: **candidate subtree enumeration** and **frequency subtree counting**. The candidate subtree enumeration refers to the task of enumerating all possible subtrees of the given document tree in a complete, non-redundant and efficient manner. While there are many techniques for enumerating the candidate subtrees (Chi et al., 2005; Hadzic, Tan et al., 2011), in this thesis the candidate subtrees are enumerated based on the tree model guided (TMG) approach as proposed by (Tan et al., 2006; Tan, Hadzic, Dillon, Chang, et al., 2008). This approach offers a non-redundant systematic enumeration model that will ensure that only those valid candidates are generated which conform to the actual tree structure of the data. The application of TMG candidate generation utilized the tree representation identified as an *embedding list* to present the structural aspect of XML

documents in an efficient way. This technique represents the structural node information in the tree that enables efficient implementation of TMG generation.

With this approach, the XML documents are scanned in order to create the global pre-order sequence in memory identified as *dictionary*. The *dictionary* acts as the shared global nodes' related information that can be directly accessed. The advantages of having shared global nodes is that this will provide all necessary information for the general lookup process, thus enabling the further scanning of the documents themselves.

In the next step, once the candidate subtree is enumerated, it needs to be counted in order to determine whether it is frequent so that the infrequent ones can be removed from the process. In counting the enumerated subtrees, the subtrees need to be represented in a proper condition that will enable the counting process to work properly. A *hashing* method (Agrawal & Srikant, 1994; Chi et al., 2005) is a conventional and effective way of counting the subtrees (Tan, Hadzic, Dillon, & Chang, 2008). The pre-order string encoding (Zaki, 2005) is a *hashing* method utilized in this thesis for counting the subtrees. With this approach, the labels of subtrees are listed sequentially according to the pre-order traversal of each subtree. In tracking the order of the subtrees, a backtrack symbol ("/") is used to track the movement node in the tree during the pre-order traversal of the tree being represented. As asserted earlier, space optimization and ease-of-manipulation are the major advantages of the string-like encoding technique. The string-like encoding treats every subtree as unique. All occurrences of a particular subtree in the document are counted in order to determine the frequency of a subtree. In the following section, the IMB3 Miner algorithm utilized in the aforementioned step to generate frequent subtrees is outlined.

7.3.2 IMB3 Miner

In this thesis, the IBM3 Miner (Tan et al., 2006) algorithm is utilized to discover frequent subtrees from a tree database. This algorithm as stated by (Tan, Hadzic, Dillon, & Chang, 2008) was developed based on the Tree Model Guided (TMG) candidate enumeration framework and it utilizes the level of embedding constraint to

mine induced subtrees. While the details regarding the development of this algorithm are not part of this thesis, in general this algorithm involves several processes as follows. The tree-structured database is first transformed into a database of a rooted integer-labelled ordered tree. The modelling process of the tree database is demonstrated in Section 7.2. For the generation of the frequent subtrees, the tree database is traversed once to create a global sequence which stores each node in the pre-order traversal together with the necessary node information. The Recursive List (RL) is constructed based on the global sequence generated from the traversal database. When the RL is constructed, the TMG is utilized to generate the subtree candidates. Before certain a subtree is stored as a frequent subtree, the pruning task is undertaken to ensure that all generated subtrees do not contain infrequent subtrees. To determine whether a subtree is frequent, the occurrence of that subtree is counted and checked to ascertain whether is greater than or equal to the specified minimum support.

7.4 Flat Data Format for Tree-Structured Data

As discussed in 7.3.4, IMB3 Miner is an algorithm that is useful for discovering frequent subtrees in tree-structured data. There are many well-developed frequent subtree mining algorithms tailored to solve certain applications and problems. However, the main concern regards their capabilities in handling instances in a tree database that are characterized by complex tree structures. In certain domain applications, capturing the information using an XML document may result in complex tree-structured data. This can happen due to a large number of elements (attributes) that are likely to be present in every instance or transaction. Additionally, these attributes might not be as useful for the decision-making process, but they significantly increase the complexity of the frequent subtree mining task due to the combinatorial complexity. As mathematically shown in (Hadzic, Tan et al., 2011; Tan, Hadzic, Dillon, Chang et al., 2008), certain structural characteristics of data may cause infeasibility in the frequent subtree mining task due to the enormous volume of candidate subtrees that need to be enumerated.

Moreover, even though frequent subtree patterns supplied with certain threshold parameters might discover interesting patterns, indeed the pattern set itself can be

very large and unwieldy, with many patterns within it being irrelevant and not useful, generated based on random discovery. For example, a very low support threshold may be provided to extract certain underlying rules from certain applications. However, lowering the support threshold might affect the performance of the algorithm as the inherent complexity of the task is increased, and in the worst case, no results would be obtained. In addition, at low support thresholds, the frequent subtree patterns themselves may be so large in number that they cause significant delays in the analysis and interpretation of the results.

Given the above situations, the applicability of frequent subtree mining based approach might not be a suitable preference to implement. A new method for effectively representing tree-structured data in a structure-preserving flat format has recently been proposed in (Hadzic, 2011). The main motivation of the method is to enable a wider range of well-established data mining analysis/techniques, previously developed for flat data format, to be applied directly to tree-structured data. It is promising in the sense that many of the complexity issues caused by the structural properties in the document can be overcome, and class distinguishing criteria can be directly sought after. In the context of the work of this thesis, the framework for verifying the interestingness of the rules using statistical analysis, and redundancy and contradictive assessment methods has proven effective in the context of flat or relational data. Having a structure-preserving flat format for tree-structured data will enable such a framework to be applied to tree-structured data, and the statistical analysis, and redundancy and contradictive assessment methods will not need to undergo major modifications in order to adapt to structural characteristics. However, there are still some important implications that need to be discussed when applying statistical analysis, and redundancy and contradictive assessment methods to determine the interestingness of tree-structured association rules.

7.4.1 Database Structure Model (*DSM*)

The definition given by (Hadzic, 2011) is utilized here to describe the *Database Structure Model* (*DSM*). In a relational table format, the first row of a table consists of attribute name. In a tree database, however, the attributes themselves are scattered throughout the independent tree instances. To address this problem, (Hadzic, 2011)

has utilized a structure according to which all the instances/transactions are organized. Each of the transactions in a tree-structured document should be a valid subtree of this assumed structure, which is referred to as the *Database Structure Model* (*DSM)* (Hadzic, 2011). Generally, the string-like representation of a tree database, an example of which was given in Table 7.3, is converted into a flat data format while preserving the ancestor-descendant and sibling node relationships. Henceforth, this structure-preserving flat data representation will be simply referred to as 'table'.

The first row of the table contains the *DSM* without any specific attribute names. It represents only the most general structure where every instance from the tree database can be matched to. This will ensure that when the labels of a particular transaction from the tree database are processed, they are placed in the correct column, corresponding to the position in the *DSM* that this label matches. The following explanation of the *DSM* technique is reproduced from (Hadzic, 2011);

"*The labels (attribute names) of this DSM will correspond to pre-order positions of the nodes of the DSM and sequential position of the backtrack ('-1) symbols from the string encoding of this DSM. The process of extracting a DSM from a tree database consists of traversing the tree database and expanding the current DSM as necessary so that every tree instance can be matched against DSM. Let the tree database consisting of n transactions be denoted as* $Tdb = \{tid_0, tid_1, ..., tid_{n-1}\}$*, and let the string encoding of the tree instance at transaction* tid_i *be denoted as* $\varphi(tid_i)$*.). Further, let* $|\varphi(tid_i)|$ *denote the number of elements in* $\varphi(tid_i)$*, and* $\varphi(tid_i)_{\mathbf{k}}$ $(k = \{0, 1, ..., |\varphi(tid_i)|-1\})$ *denote the* k_{th} *element (a label or a backtrack '-1') of* $\varphi(tid_i)$*. The same notation for the string encoding of the (current) DSM is used, i.e.* $\varphi(DSM)$*".* The pseudo code of extracting the *DSM* from *Tdb* process is shown in Figure 7.4

```
Input: Tdb
Output: DSM
inputNodeLevel = 0 // current level of φ(tid_i)_k
DSMNodeLevel = 0 // current level of φ(T(h_max, d_max))_k
φ(DSM) = φ(tid_0) // set default DSM
for i = 1 to n – 1  // n = |Tdb|
  for each φ(tid_i)_k in φ(tid_i)
    for each p = 0 to (|φ(DSM)|-1)
      if  φ(tid_i)_k = -1 then inputNodeLevel– – else inputNodeLevel++
      if φ(DSM)_p='b_i' then DSMNodeLeve– – else DSMNodeLevel++
      if inputNodeLevel ≠ DSMNodeLevel
          if  φ(tid_i)_k = -1 then
            while inputNodeLevel ≠ DSMNodeLevel
              p++
              if  φ(DSM)_p = -1 then DSMNodeLevel– – else DSMNodeLevel++
            endwhile
        else
            while inputNodeLevel ≠ DSMNodeLevel
              append φ(tid_i)_k at position p+1 in φ(DSM)
              k++
              p++
              if  φ(tid_i)_k = -1 then inputNodeLevel– – else inputNodeLevel++
          endwhile
    endfor
  endfor
endfor
return DSM
```

Figure 7.4: Model Tree Extraction from a Tree database *Tdb* reproduced from (Hadzic, 2011)

To illustrate the complete conversion process using *DSM,* please refer to Figure 7.1. Using the string encoding format representation (Zaki, 2005), the tree database *Tdb* from Figure 7.1 would be represented as is shown in Table 7.1, where the left column corresponds to the transaction identifiers, and the right column is the string encoding of each subtree.

In this example, the *DSM* is reflected in the structure of T_6 in Figure 7.1 and it becomes the first row of the table to reflect the attribute names as explained previously. The string encoding is used to represent this uniform structure and since the order of the nodes (and backtracks ('-1)) is important, the nodes and backtracks are labeled sequentially according to their occurrence in the string encoding. For nodes (labels in the string encoding), x_i is used as the attribute name, where i corresponds to the pre-order position of the node in the tree, while for backtracks, b_j is used as the attribute name, where j corresponds to the backtrack number in the string encoding. Hence, from our example in Figure 7.1 and Table 7.1, $\varphi(DSM)$ = 'x_0 x_1 x_2 x_3 b_0 x_4 b_1 b_2 b_3 x_5 x_6 x_7 b_4 x_8 b_5 b_6 b_7 '.

To fill in the remaining rows, every transaction from *Tdb* is scanned and when a label is encountered, it is placed in the matching column (i.e. under the matching node (x_i) in the *DSM*), and when a backtrack ('-1') is encountered, a value '1' (or 'y') is placed in the matching column (i.e. matching backtrack (b_j) in *DSM*). The remaining entries are assigned values of '0' (or 'no', indicating non existence). The flat data format of *Tdb* from Table 7.1 (and Figure 7.1) is illustrated in Table 7.4.

Table 7.4: Flat representation of *Tdb* in Figure 7.1 and Table 7.1

$\mathbf{x_0}$	$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{b_0}$	$\mathbf{x_4}$	$\mathbf{b_1}$	$\mathbf{b_2}$	$\mathbf{b_3}$	$\mathbf{x_5}$	$\mathbf{x_6}$	$\mathbf{x_7}$	$\mathbf{b_4}$	$\mathbf{x_8}$	$\mathbf{b_5}$	$\mathbf{b_6}$	$\mathbf{b_7}$
a	b	d	n	1	e	1	1	1	c	0	0	0	0	0	0	1
b	c	0	0	0	0	0	0	1	b	e	0	0	0	0	1	1
b	d	e	0	0	0	0	1	1	0	0	0	0	0	0	0	0
l	m	0	0	0	0	0	0	1	n	0	0	0	0	0	0	1
k	l	m	0	0	0	0	1	1	n	0	0	0	0	0	0	1
b	a	c	f	1	0	0	1	1	d	0	0	0	0	0	0	1
a	b	c	d	1	e	1	1	1	f	g	h	1	i	1	1	1

The conversion process can be formalized as follows. Let the tree database consisting of n transactions be denoted as $Tdb = \{tid_0, tid_1, \ldots, tid_{n-1}\}$, and let the string encoding of the tree instance at transaction tid_i be denoted as $\varphi(tid_i)$. The *DSM* is extracted from *Tdb* using the procedure explained earlier. Further, let $|\varphi(tid_i)|$ denote the number of elements in $\varphi(tid_i)$, and $\varphi(tid_i)_{\mathbf{k}}$ ($k = \{0, 1, \ldots, |\varphi(tid_i)|-1\}$) denote the k_{th} element (a label or a backtrack '-1') of $\varphi(tid_i)$. The flat data format or table F_T (C, R) (C = columns, R = rows) is set up where $C = \{c_0, c_1, \ldots, c_{m-1}\}$ ($m = |C| = |\varphi(DSM)|$), and $R = \{r_0, r_1, \ldots, r_{p-1}\}$ ($p = |R| = n+1$ (i.e. extra column for attribute names). The value in column number x and row number y is denoted as $F_T (c_x, r_y)$. Hence, to set the attribute names $F_T (c_i, r_0) = \varphi(DSM)_k$ where $i = k = \{0, 1, \ldots, (|\varphi(DSM)|-1)\}$. The process of populating the entries from F_T using *Tdb* is explained by the pseudo code in Figure 7.5.

```
Input: Tdb, DSM
Output: F_T
// set up the attribute name row in F_T
F_T (c_i,r_0)=φ(DSM)_k ∀ i=k={0, …, (|φ(DSM)|-1))
inputNodeLevel = 0 // current level of φ(tid_i)_k
DSMNodeLevel = 0 // current level of φ(DSM)_k
// populate F_T
for i = 0 to n – 1  // n = |Tdb|
  for each φ(tid_i)_k in φ(tid_i)
    for p = 0 to (|φ(DSM)|-1)
      if  φ(tid_i)_k = -1 then inputNodeLeve– – else inputNodeLevel++
      if φ(DSM)_p='b_i' then DSMNodeLevel– – else DSMNodeLevel++
      if inputNodeLevel = DSMNodeLevel
        if  φ(tid_i)_k = -1 then F_T (c_p,r_i+1) = 1 else F_T (c_p,r_i+1) = φ(tid_i)_k
       else // level mismatch traverse φ(DSM) until match
         while inputNodeLevel ≠ DSMNodeLevel
           F_T (c_p,r_i+1) = 0
           p++
           if  φ(DSM)_p ='b_i' then DSMNodeLevel– – else DSMNodeLevel++
         endwhile
         if  φ(tid_i)_k = -1 then F_T(c_p,r_i+1) = 1
         else F_T (c_p,r_i+1) = φ(tid_i)_k
    endfor
  endfor
endfor
return F_T
```

Figure 7.5: Tree database *Tdb* to flat data format (*FDT*) conversion reproduced from (Hadzic, 2011)

In addition to that, during the conversion process as mentioned in (Hadzic, Hacker, & Tagarelli, 2011; Hadzic & Hecker, 2011), one can incorporate the minimum support threshold *s* so that the *DSM* captures only those structural characteristics that have occurred in at least *s*% of the tree database. Hence, in some cases only a fraction of a tree instance can be matched to the *DSM* due to low occurrences in the tree database, but the partial information still needs to be included in the resulting flat table.

As an example, refer to the tree database (*Tdb*) in Table 7.4 and Figure 7.1, in mining the subtrees with minimum support threshold of 3, the resulting *DSM* would be as follows: "$x_0, x_1, x_2, b_0, b_1, b_2, x_3, b_3$" and the new table is shown in Table 7.5.

Table 7.5: Flat representation of *Tdb* in Figure 7.1 and Table 7.1 when minimum support = 3

x_0	x_1	x_2	x_3	b_0	b_1	b_2	x_4	b_3
a	b	c	n	1	1	1	c	1
b	c	0	0	0	0	1	b	1
b	d	e	0	0	1	1	0	0
l	m	0	0	0	0	1	n	1
k	l	m	0	0	1	1	n	1
b	a	c	f	1	1	1	d	1
a	b	c	d	1	1	1	f	1

7.4.1.1 Tree to Flat Conversion Example using DEBII WebLogs Data

Referring to the DEBII WebLogs data example in Section 7.2.1, the pre-order encoding format of the tree database need to be converted into a flat representation as proposed by (Hadzic, 2011). The *DSM* application and the algorithm were described earlier in Section 7.4.1. In this section, a detailed example is provided using the DEBII WebLogs example as reference.

Initially, the *DSM* needed to be constructed. In this example, the *DSM* is reflected in the structure of T_0 in Table 7.3 and the corresponding tree is shown in Figure 7.3 (Session 1). Transaction "T_0" becomes the general structure of *DSM* (Figure 7.6) and the first row in Figure 7.7 to reflect the attributes names. The string encoding is utilized to represent the *DSM* and since the order of the nodes (and backtracks (-1)) is important, the nodes and backtracks are labeled sequentially according to their occurrence in the string encoding.

"x_0 x_1 x_2 x_3 b_0 x_4 b_1 b_2 x_5 x_6 b_3 b_4 b_5 x_7 x_8 x_9 b_6 b_7 b_8 x_{10} x_{11} b_9 x_{12} b_{10} b_{11}"

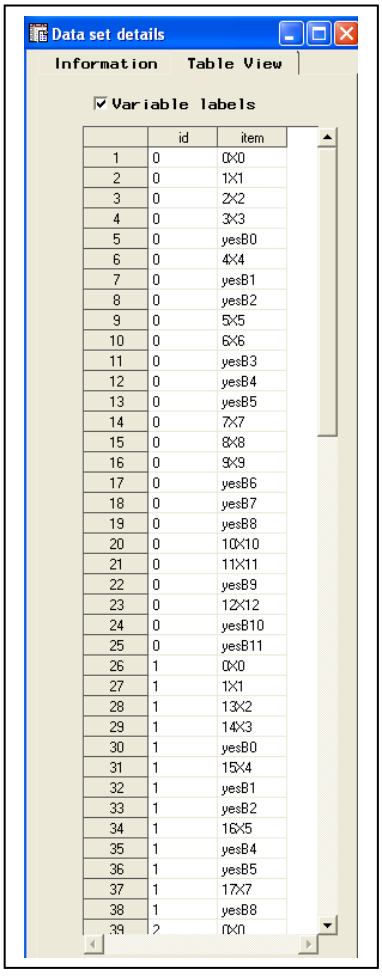
Figure 7.6: General Structure for *DSM*

As the *DSM* has been chosen according to the aforementioned criteria, consequently this will be the first row of the newly-generated flat table. Every transaction that remains in the *Tdb* will be matched against the *DSM* and every node label placed in the matching column (i.e. under the matching node (x_i) in the *DSM*), and when a backtrack ('-1') is encountered, a value 'yes' is placed to the matching column (i.e. matching backtrack (b_j) in *DSM*). The remaining entries are assigned a value of 'no' (indicating non existence). The flat data format of *Tdb* from Table 7.3 is illustrated in Figure 7.7.

x_0	x_1	x_2	x_3	b_0	x_4	b_1	b_2	x_5	x_6	b_3	b_4	b_5	x_7	x_8	x_9	b_6	b_7	b_8	x_{10}	x_{11}	b_9	x_{12}	b_{10}	b_{11}
0	1	2	3	yes	4	yes	yes	5	6	yes	yes	yes	7	8	9	yes	yes	yes	10	11	yes	12	yes	yes
0	1	13	14	yes	15	yes	yes	16	no	no	yes	yes	17	no	no	no	no	yes	no	no	no	no	no	no
0	1	18	19	yes	no	no	yes	no	no	no	no	yes	20	21	no	no	yes	yes	7	22	yes	no	no	yes
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

Figure 7.7: Flat representation of DEBII WebLogs *Tdb* in Table 7.3

As described in Chapter 4 (Framework *B*), the flat data format as illustrated in Figure 7.7 is used to generate the association rules. The Apriori algorithm is then utilized to generate the frequent rules. Figure 7.8 below depicts the data format fed to the SAS Enterprise Miner software to generate the frequent rules (this data format was described earlier in Chapter 5 (Section 5.6)).



	id	item
1	0	0X0
2	0	1X1
3	0	2X2
4	0	3X3
5	0	yesB0
6	0	4X4
7	0	yesB1
8	0	yesB2
9	0	5X5
10	0	6X6
11	0	yesB3
12	0	yesB4
13	0	yesB5
14	0	7X7
15	0	8X8
16	0	9X9
17	0	yesB6
18	0	yesB7
19	0	yesB8
20	0	10X10
21	0	11X11
22	0	yesB9
23	0	12X12
24	0	yesB10
25	0	yesB11
26	1	0X0
27	1	1X1
28	1	13X2
29	1	14X3
30	1	yesB0
31	1	15X4
32	1	yesB1
33	1	yesB2
34	1	16X5
35	1	yesB4
36	1	yesB5
37	1	17X7
38	1	yesB8
39	2	0X0

Figure 7.8: Data format (DEBII WebLogs) for Apriori algorithm based on Flat Representation in Figure 7.7

7.4.1.2 Representing Disconnected Trees w.r.t. *DSM*

As discussed earlier in Section 7.4.1, the *DSM* preserves the structural characteristics in the data mining results, since the extracted patterns are mapped onto the *DSM* to re-generate the pre-order string encoding of subtrees. Even though the rules from *DSM* can be converted into pre-order string encoding of the subtrees, and hence are represented as subtrees of the tree database, still the implication of the conversion process needs to be highlighted. There are some rules that may not be representatives of valid subtrees, due to the fact that data mining tasks used for structure-preserving flat data representation are not performed by following the structure, but rather, structural properties are preserved by the *DSM*.

For example, it is possible that some items in the rules correspond to sibling nodes in the original tree, while the parent or any ancestor node connecting those in the original tree is not present in the rules discovered using *DSM* approach. Hence, this would result in an invalid subtree as the nodes are disconnected. In addressing this matter, one can add the other nodes that make it into a valid subtree but flag them as irrelevant. Furthermore, the user can always choose to match the formed subtree to the general structure model represented by the *DSM*. The process consists of sequentially listing the values of each matched node in *DSM*, while retaining the level of information of each current node in *DSM* and in the subtree pattern. Since the *DSM* itself is ordered according to the pre-order traversal, this results in pre-order string encodings of the subtrees.

As a simple illustrative example, consider the following associations/patterns extracted from DEBII WebLogs Data;

P_1: *business-intelligence & human-space-computing & phd-msc*
P_2: *scholarships & management & phd-a-msc*

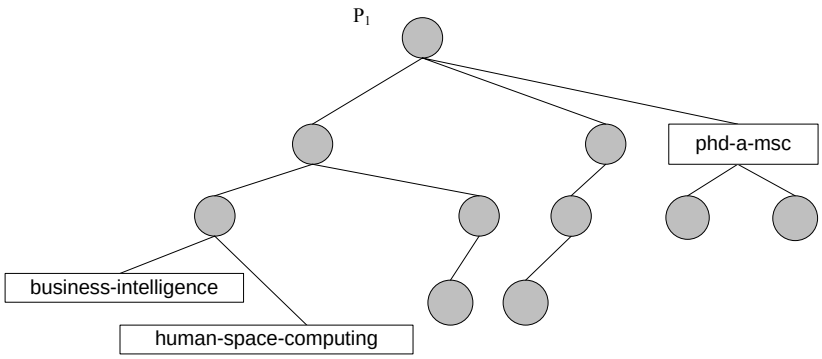


Figure 7.9: Displaying Pattern (P_1) w.r.t *DSM* in Figure 7.6

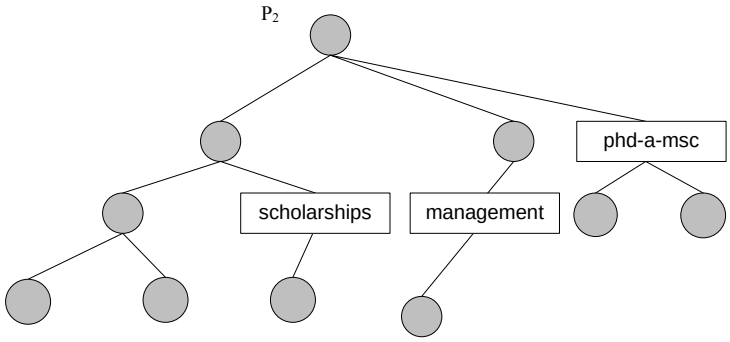


Figure 7.10: Displaying Pattern (P_2) w.r.t *DSM* in Figure 7.6

With refer to pattern (P_1) in Figure 7.9 and pattern (P_2) in Figure 7.10, there are items (nodes) in the rule correspond to sibling nodes in the original tree, while the parent or any ancestor node connecting those in the original tree is not present in the rule. Hence, this would result in an invalid subtree as the nodes are disconnected. This is illustrated in both Figure 7.9 and Figure 7.10, where irrelevant nodes are shaded grey. One can also choose to display the labels of nodes that are there to contextualize the information, i.e. *scholarships* and *management* and *phd-a-msc*, which would essentially contextualize the specific rule constraints. Additionally, the labels of nodes can be displayed in order to contextualize the information in the tree. In this work, these rules are recognized as *FullTree* rules.

7.5 Evaluation of Framework for Tree-Structured Data

The evaluation of the unification framework is performed using the Prion (Hadzic, Dillon, Sidhu, Chang, & Tan, 2006), DEBII WebLogs (Hadzic, 2011), web access trees from the computer sciences department of the Rensselaer Polytechnic Institute (CSLogs) (Zaki & Aggarwal, 2003) and CRM dataset. These datasets are all in the form of integer labeled trees. The purpose of the experiments is to demonstrate that by using the tree database transformation, one can still discover useful knowledge from tree-structured data using techniques developed for a flat data format. A general description of datasets is provided in Section 7.5.1, followed by the dataset characteristics for flat table formats in Table 7.7 to Table 7.11.

The evaluation processes are then separated into two parts. The first part is explained in Section 7.6. As an initial experiment, the focus is solely on the evaluation of frequent subtrees from Prion data that has been generated using the IMB3-Miner (Tan et al., 2006). Since this dataset was rather limited in depth, only a small adjustment was needed in the general framework for measuring rule interestingness proposed in this thesis. However, in tree databases where the average depth of every tree instance can be large, and complex structural properties exist, the proposed framework would need to undergo considerable adjustments in order to be applied. To alleviate this problem, the *DSM* approach is used (explained earlier) in the second part of the experiment where more complex tree databases are considered. Hence, the focus is on evaluating the frequent subtrees from flat table format using the *DSM* approach. The experiments are conducted using 3 sets of data, namely the DEBII WebLogs, CSLogs and CRM data. The whole set of experiments is described in Section 7.7.

Table 7.6 sums up the structural characteristics of each dataset and its content variations, and the following notation is used: |Tr| - Number of transactions (independent tree instances); |L| - Number of unique labels; |T| - Number of nodes (size) in a transaction; |D| - Depth; |F| - Fan-out-factor (or degree); (Avg = average). As exhibited in Table 7.6, the DEBII WebLogs, CSLogs and CRM dataset contains more complex data compared with Prion data, especially with respect to the tree instances having relatively more depth.

Table 7.6: General Dataset Characteristics

Dataset	|Tr|	|L|	Avg |T|	Avg |D|	Avg |F|	Max |T|	Max |D|	Max |F|
Prions	17511	46851	12.97	1	11.98	19	1	18
DEBII WebLogs	18836	34052	9.63	4.98	1.56	60	59	37
CSLogs	68302	16207	7.80	3.45	1.82	313	123	137
CRM	1181	10611	52.97	4.89	8.00	533	5	46

7.5.1 Dataset Characteristics

Framework *B* is evaluated on four sets of XML data, namely the Prion, DEBII WebLogs, CSLogs and CRM dataset. The details of these datasets are as follows:

Prions

Proteinaceous Infectious Particle data or Prion is a database of rooted ordered labeled subtrees that described the protein instances stored for Human Prion Proteins (Sidhu, Dillon, & Chang, 2006). The Prion dataset has been utilized by (Sidhu et al., 2006) in describing Protein Ontology database for Human Prion proteins in XML format (Sidhu, Dillon, Sidhu, & Setiawan, 2004). Within this XML format, the Prion's XML tags are mapped to integer indexes similar to the format used in (Tan, Dillon, Hadzic, Chang, & Feng, 2005) and (Zaki, 2005). The general characteristics of Prion data are provided in Table 7.6 and detailed information about the data is given in Section 7.6.

DEBII WebLogs

The DEBII WebLogs data is an apache2 (v2.2.3) web server logs files taken from the DEBII website (debii.curtin.edu.au). The DEBII WebLogs data was initially used in (Hadzic & Hecker, 2011) in utilizing the *DSM* application. For the purpose of the work in this thesis, the similar setting of the DEBII WebLogs data as described in (Hadzic & Hecker, 2011) has been utilized. The data was collected for a four-month period in its native (default) format. During this period, all access to the DEBII website was stored in logs files, while messages stored in the normal error message logs were excluded. The access to the website was then classified as "internal" (within the university) and "external" (outside the university). The grouped user sessions were converted to trees as was explained with the illustrative example in Section 7.2.1. The resulting dataset had 18836 instances, of which 66% was used for training and the remainder for testing. The details of the setting of the DEBII

WebLogs access can be found in (Hadzic & Hecker, 2011). Table 7.6 shows the general characteristics of the data, while in Table 7.7 and Table 7.8 the characteristics of DEBII WebLogs data in flat table format are given.

CSLogs

CSLogs data comprises the web access trees from the computer science department of the Rensselaer Polytechnic Institute previously used in (Zaki & Aggarwal, 2003) to evaluate the *XRules* structural classifier. The tree instances are labeled according to two classes, namely the internal and external web site access. For the purpose of the work of this thesis, all of the three datasets (US1924, US2430, and US304) were combined and instances were replicated to make the class distribution even. The total number of combined instances is 68302. Sixty-six percent of the data was used for the training set and the remainder for the testing set. Since different support thresholds were used, in our approach the flat data representation of the dataset is done separately for each support threshold, as the extracted *database structure model* (*DSM*) varies; hence, the number of attributes used during frequent pattern generation. The general characteristics of the data are shown in Table 7.6, while the flat table format for CSLogs data is provided in Table 7.9 and Table 7.10.

CRM

CRM data refers to Customer Relationship Management and is a real-world dataset relating to the handling of complaints in the area of real estate. These complaints are then classified into "WorkCompletion", with 2 possible values (work completion $\leq$ 3 or $\geq$ 4 weeks). The dataset consists of 1181 instances with 675 attributes, of which 66% was used for training and 34% for the testing set. However, there are many complex classes within this CRM data which may interest the users of the data. Nevertheless in this case, as our main purposes is not to analyze the problem of CRM itself, but to look at the CRM data as an example of tree-structured data, the attention is confined to the aforementioned class. The general characteristics of the data are shown in Table 7.6, while the flat table format for CRM data is provided in Table 7.11.

Table 7.7 to Table 7.11 summarizes the general characteristics of the datasets based on the flat table format representation. It shows the number of transactions, the

number of attributes, and the number of selected attributes based on Symmetrical Tau (ST) feature selection in each dataset. The table also shows the initial rule sets that have been generated, namely *FullTree*, *Embedded* and *Induced*.

FullTree refers to the rules that may contain a disconnected tree, while *embedded* and *induced* is a subset of rules from *FullTree* rules (a definition of *FullTree* is given in Section 7.4.1.2). Generally, rules from *FullTree* may consist of invalid subtrees, but one can add the other nodes that make them into valid subtrees but flag them as irrelevant. *Induced* and *Embedded* rules contain only connected trees from *FullTree,* and induced subtrees preserve parent-child relationships, while in embedded subtrees, a parent of a node can correspond to an ancestor of the node in the original tree The embedded and induced approaches were included in this experiment as the rules that occur in *FullTree* may not be representative of valid subtrees due to the presence of disconnected nodes. This is because the data mining task, statistical analysis, and redundancy and contradictive assessment methods employed in the framework are not performed by following the structure.

The selected attributes based on ST in Column 4 in Table 7.7 to Table 7.11 are measured according to their capabilities in predicting the values of the attribute class in each dataset such as "DEBII WebLogs Data: Access (Internal and External)", CSLogs Data: Access (Internal and External)" and "CRM Data: WorkCompletion (work completion $\leq$ 3 or $\geq$ 4 weeks)". As previously discussed in Chapters 3 and 6, the ST feature selection criterion was used earlier in the framework to remove any irrelevant attributes. Furthermore, this would reduce the number of subtrees that may contain irrelevant attributes/nodes. A relevant cut-off point is chosen based on decreasing ranking of the nodes. The significant difference was considered to occur in the ranking at the position where that attribute's ST value is less than half of the previous attribute's ST value in the ranking. At this point and below in the ranking, all attributes are considered as irrelevant.

As for confidence thresholds, the default confidence parameter was set at 50%, while the support was varied from 30% to 1%. The flat data representation of the dataset was done separately for each support threshold. This resulted in variations in the

extracted *database structure model* (*DSM*) and the number of attributes used for rules verification purposes.

Table 7.7: DEBII WebLogs Data with Backtrack Characteristics based on *DSM* Application

DEBII WebLogs Dataset	**# Transactions**	**#Attr.* (*DSM*)**	**# Selected Attr. (Sym. Tau)**	**# of Rules with Target Variable**		
				FullTree	***Embedded***	***Induced***
1%	18836	442	437	-	-	-
5%	18836	126	123	28282	28280	28280
10%	18836	70	63	234	234	234
20%	18836	36	29	50	49	49
30%	18836	26	19	14	13	13

Table 7.8: DEBII WebLogs Data without Backtrack Characteristics based on *DSM* Application

DEBII WebLogs Dataset	**# Transactions**	**#Attr.* (*DSM*)**	**# Selected Attr. (Sym. Tau)**	**# of Rules with Target Variable**		
				FullTree	***Embedded***	***Induced***
1%	18836	222	221	308	278	156
5%	18836	64	63	17	15	15
10%	18836	36	35	8	7	4
20%	18836	19	18	2	1	1
30%	18836	14	13	1	No Rules	No Rules

Table 7.9: CSLogs Data with Backtrack Characteristics based on *DSM* Application

CSLogs Dataset	**# Transactions**	**#Attr.* (*DSM*)**	**# Selected Attr. (Sym. Tau)**	**# of Rules with Target Variable**		
				FullTree	***Embedded***	***Induced***
1%	68302	222	217	13835	13833	13809
5%	68302	64	52	920	919	918
10%	68302	40	29	216	215	215
20%	68302	24	11	48	47	47
30%	68302	16	7	32	31	31

Table 7.10: CSLogs Data without Backtrack Characteristics based on *DSM* Application

CSLogs Dataset	**# Transactions**	**#Attr.* (*DSM*)**	**# Selected Attr. (Sym. Tau)**	**# of Rules with Target Variable**		
				FullTree	***Embedded***	***Induced***
1%	68302	127	125	144	138	138
5%	68302	43	41	8	7	7
10%	68302	24	22	4	3	3
20%	68302	17	15	No Rules	No Rules	No Rules
30%	68302	12	10	No Rules	No Rules	No Rules

Table 7.11: CRM Data with Backtrack Characteristics based on *DSM* Application

CRM Data	**# Transactions**	**#Attr.* (*DSM*)**	**# Selected Attr. (Sym. Tau)**	**# of Rules with Target Variable**		
				FullTree	***Embedded***	***Induced***
5%	1181	*675	586	+27116	+5270	+5270

Legend (Table 7.7-7.11):

(-) = Unavailable

(*) = Including the Class variable

(+) = Number of rules after ST application

7.6 Evaluation of Frequent Subtrees from IMB3-Miner

The IMB3 algorithm is utilized to discover the frequent patterns from the Prion dataset. The IMB3 algorithm discovered a total of 27 occurring patterns. The minimum support value was set at 10%. Table 7.12 shows three examples of patterns discovered.

Table 7.12: Examples of Several Patterns Discovered Based on IMB3-Miner Algorithm

Patterns #	**Patterns**	**# of Occurrences**
1	ATOMChain(A) Element(C)	3957
2	ATOMChain(A) ATOMResidual(TYR) Occupancy (1)	1743
3	ATOMChain(A) Occupancy(1) Temperature(0) Element(C)	3805

Pattern number 1 shows an association between *ATOMChain(A)* with *Element(C)* and this pattern was discovered 3957 times. Here, the *ATOMChain* with value *A* is associated with *Elements* with value *C*. The patterns discovered by the IMB3 Miner algorithm can aid in discovering potentially useful pattern structures in Protein Ontology datasets, which makes it useful for comparison of protein datasets taken across protein families and species and helps to establish interesting similarities and differences. However, the question still remains whether these patterns are discovered due to pure coincidence resulting from certain randomness in the particular dataset being analyzed. Furthermore, they are often quite large in number, which can degrade the analysis procedure, and hence in the next section the

statistical significance of the discovered patterns is measured in order to remove any non-significant patterns.

7.6.1 Subtrees Significance Test

Statistical analysis approaches, namely the chi-squared test and log-linear analysis were used to determine the usefulness of frequent rules obtained. The results from the chi-squared test are discussed first.

Table 7.13: Patterns Verification Based on Chi-Squared Test

Node Name		**Sig. Att. Value**
ATOMResidual(TYR)	Occupancy(1)	Not Sig.
Occupancy(1)	Temperature(0)	Not Sig.
ATOMChain(A)	Occupancy(1)	Not Sig.
ATOMChain(A)	Element(C)	Sig.
ATOMChain(A)	Element(H)	Sig.
Temperature(0)	Element(C)	Sig.
Temperature(0)	Element(H)	Sig.
ATOMChain(A)	ATOMResidual(TYR)	Sig.
ProteinOntoLogsyID(3)	Occupancy(1)	Not Sig.
ProteinOntoLogsyID(3)	Element(C)	Sig.
ATOMChain(A)	Temperature(0)	Sig.
Occupancy(1)	Temperature(1)	Not Sig.
Occupancy(1)	Element(N)	Not Sig.
Occupancy(1)	Element(C)	Not Sig.
Occupancy(1)	Element(O)	Not Sig.
Occupancy(1)	Element(H)	Not Sig.

Table 7.13 shows that there are 16 association relationships among structured items discovered using the IMB3 algorithm. Based on the chi-squared test, 7 of the 16 relationships are significant. Table 7.14 shows 11 patterns with more than two nodes. The log-linear analysis is used to examine the association between these nodes. Only one pattern out of 11 patterns is accepted as a significant pattern based on this analysis. Based on the log-linear analysis, the result revealed that there is a significant association between *ATOMChain(A)*, *Temperature(0)* and *Element(H)*.

Table 7.14: Patterns Verification Based on Log-Linear Analysis

Node Name				**Sig. Att. Value**
ATOMChain(A)	ATOMResidue(TYR)	Occupancy(1)		Not Sig.
ATOMChain(A)	Occupancy(1)	Temperature(0)		Not Sig.
ATOMChain(A)	Occupancy(1)	Element(C)		Not Sig.
ProteinOnto(3)	Occupancy(1)	Element(C)		Not Sig.
ATOMChain(A)	Occupancy(1)	Element(H)		Not Sig.
Occupancy(1)	Temperature(0)	Element(C)		Not Sig.
ATOMChain(A)	Temperature(0)	Element(C)		Not Sig.
Occupancy(1)	Temperature(0)	Element(H)		Not Sig.
ATOMChain(A)	Temperature(0)	Element(H)		Sig.
ATOMChain(A)	Occupancy(1)	Temperature(0)	Element(C)	Not Sig.
ATOMChain(A)	Occupancy(1)	Temperature(0)	Element(H)	Not Sig.

7.6.2 Prion as a Classification Problem

Previous works in (Shaharanee, Hadzic, & Dillon, 2009; Shaharanee, Hadzic, & Dillon, 2011) show that the unification framework involves several steps in evaluating the rules discovered from the association rules mining process. For the Prion dataset, similar steps were followed. A new variable (target variable) identified as *Human Protein* or *Animal Protein* class is defined. This new variable was derived from *ProteinOntologyID* and *SuperFamily* variables. Hence, the *ProteinOntologyID* and S*uperFamily* variables are excluded from the dataset used for this task. Thus, for this classification problem, the target variable (i.e., Human or Animal's Protein) is chosen as the right hand side/consequence of the association rules.

In this experiment, the Prion dataset is divided into training set and testing set. Then the pre-processing techniques are applied including the removal of missing values and discretization of attributes with continuous data. The equal depth binning approach method was selected as it produces a better result as discussed in (Shaharanee, Hadzic, & Dillon, 2009; Shaharanee, Hadzic, & Dillon, 2011). The determination of relevant attributes with respect to being able to predict the target attributes is shown in Table 7.15. This is based on Symmetrical Tau (Zhou & Dillon, 1991). The Symmetrical Tau (ST) measure is effective in distinguishing criteria for the class to be predicted, as it does not favor multi-valued attributes. The attributes with ST values that are respectively lower than other attribute's ST values, are considered as irrelevant for the task. The significant difference was considered to occur at the position where that attribute's ST value is less than half of the previous

attribute's ST value in the ranking. Hence for this dataset, attributes '*Occupancy*' and '*Y*' were considered as irrelevant for the prediction task and were removed.

Table 7.15: Symmetrical Tau Result for Prion Dataset

Variables	ST Values
ATOMChain	0.2088
Temperature	0.1230
Z	0.0812
ATOMid	0.0407
ATOMResSeqNum	0.0280
X	0.0256
Element	0.0153
Atom	0.0109
ATOMResidue	0.0082
Y	0.0029
Occupancy	0.0001

Next, the rules were generated based on the minimum support of 5% and confidence of 50% respectively. Table 7.16 shows examples of the generated rules. The discovered rules were then verified with statistical techniques, namely the chi-squared test (Agresti, 2007) and logistics regression analysis (Agresti, 2007). The result produced by these statistical analyses indicates that only variables *ATOMChain*, *ATOMResidual*, *ATOMResSeqNum*, *X* and *Z* were significant contributors to the target variable of class Human or Animal.

Table 7.16: Examples of Prion Rules

Set Size	Confidence (%)	Support (%)	Count	Rules
2	75.32	8.97	934	X(g) ==> Class (Animal)
4	61.71	6.66	693	X(d) & Z(b) & ATOMChain(A) ==> Class (Human)

Additional constraint measurement techniques were applied in order to discard any redundant rules (Shaharanee et al., 2011; Webb, 2007). The combination of these rule evaluation strategies will enable the association rule mining framework to determine the accuracy and high quality of rules. Table 7.17 shows the progressive difference in the number of rules generated as statistical analysis and redundancy assessment are applied. There are 116 initial rules. Upon the removal of 75% rules following ST and statistical analysis, the AR for both training set and testing set has increased by more than 1%., while the CR is still preserved at 100%. This demonstrates the importance of evaluating the association rules by statistical

analysis, as in this particular scenario, the simplified rule set is more general and performs better on unseen data. Within the context of the experiment for this dataset in particular, the statistical significance analysis was an appropriate means of discarding non-significant rules, which results in a significant reduction in the overall complexity of the rule set. Moreover, as the redundancy assessment was progressively utilized, 6 redundant rules emerged, but there were no contradictive rules. Table 7.17 shows that this substantial reduction of rules was not at the cost of a reduction in accuracy; in fact, this increased for the Prion dataset in classifying and predicting the protein classes (Shaharanee, Hadzic, & Dillon, 2010).

Moreover, the effect of altering the minimum confidence threshold was observed and it was found that the changes in *confidence* values have a direct impact on the size of the rule sets' AR and CR values. A similar experiment was carried out and described in Section 6.2.6 of Chapter 6. By increasing the confidence values from 50% to 60%, the AR for both training and testing set increased by more than 19%. However, this comes with a trade-off of losing more than 34% CR of the dataset. This is because the number of rules was reduced from 23 to 10. Of the 13 discarded rules, some might have been rules capable of capturing more instances in the database. Thus, by discarding these 13 rules, the capability of the remaining rules to cover all instances in the dataset has been decreased. The result demonstrated that there is a trade-off in accepting either high accuracy of rules with fewer instances captured or capturing more, but less accurate, instances in the dataset in their prediction/classification capabilities. Moreover, these findings are similar to those previously discussed when evaluating rules interestingness for relational data in Chapter 6. Thus, it is important to balance the trade-off between AR and CR in order to determine the optimal value for the minimum *confidence* threshold, which may differ depending on the sensitivity of the domain at hand.

Table 7.17: Accuracy Rate (AR) and Coverage Rate (CR) for Prion Data

Type of analysis	Data Partition	# Of Rules	AR %	CR%
Initial # of Rules	Training	116	59.00	100.00
	Testing		58.65	100.00
# of Rules after ST	Training	89	59.75	100.00
	Testing		59.35	100.00
Statistics Analysis	Training	29	60.22	100.00
	Testing		59.89	100.00
Redundancy Removal	Training	23	60.52	100.00
	Testing		60.12	100.00
Contradictive Removal	Training	23	60.52	100.00
	Testing		60.12	100.00
Conf. 60%	Training	10	78.59	65.84
	Testing		79.04	66.88

7.7 Evaluation of Frequent Subtree-based Rules Extracted using the *DSM* Approach

In Section 7.6, the interestingness of frequent subtree based rules from XML could directly be measured using the aforementioned statistical analysis, and redundancy and contradictive assessment methods. For these particular Prion XML documents, the evaluation task was applicable and viable as the data is a limited XML document structure with a maximum height of 1 as shown in Table 7.6.

However, as the analysis was extended to dealing with complex XML documents (Refer to Row 3, 4 and 5; the DEBII WebLogs, the CSLogs and the CRM dataset respectively in Table 7.6), it is difficult for the aforementioned measures to be applied directly to evaluate the interestingness of frequent subtrees from these datasets.

Chapter 3, with reference to the evaluation of rules from tree-structured data, presented a discussion of the complexities of XML format and the difficulties of analyzing it. One reason for this is the existence of a large number of attributes that are likely to be present in all instances or transactions (Hadzic, 2011).

In the context of the work in this thesis, following the approach taken in Chapter 5 and Chapter 6, having the data in a flat table format offers a better way of applying the statistical analysis, and redundancy and contradictive assessment methods to

evaluate the interestingness of frequent rules derived from relational data. Thus, by having a structure-preserving flat format for tree-structured data, our proposed framework can be applied directly to tree-structured data, and the statistical analysis, and redundancy and contradictive assessment methods will not need to undergo major adaptive changes according to structural characteristics.

(Hadzic, 2011) has initiated the *DSM* approach in order to represent a tree-structured database in a flat table data format. The motivation for this work was to ease the burden of analyzing XML documents and enabling the direct application of a wider range of data mining/analysis techniques to tree-structured data.

For the experiments conducted here, an evaluation of the unification Framework *B* was performed using the web access trees from the computer sciences department of the Rensselaer Polytechnic Institute (CSLogs), DEBII WebLogs and CRM dataset. As stated in Chapter 3 (Section 3.6), the focus in this thesis is on evaluating frequent subtrees with a predefined class which presents a classification problem.

The purpose of the experiments was to evaluate the frequent subtree generated from *DSM* approach. Each dataset underwent conversion into a structure-preserving flat data format (henceforth *FDT*) using the *DSM* approach. The backtrack attributes information was kept in the *database structure model* (*DSM*) as this is important for preserving the structural information. Hence, this can be used to represent the resulting rules as trees/subtrees. The backtrack attributes can be optionally kept in the *FDT* as they can indicate the existence/non-existence of a node irrespective of the label (Hadzic & Hecker, 2011). The effect of the inclusion/exclusion will be empirically studied in a later section for the datasets used in this experimental evaluation. With this application, the rules can be converted into the pre-order string encoding of the subtrees, and hence represented as subtrees of the tree database.

Using the *DSM* approach, implies that a subtree will be considered as the same entity only if all of its nodes have occurred in the same position with respect to the extracted *DSM*, as was discussed in (Hadzic, 2011). Once the frequent subtrees have been converted into flat format, the next step is to remove the irrelevant node/nodes using the ST feature selection tool. This reduces the number of frequent subtree

based rules which contain irrelevant node/nodes. However, in some cases the pre-processing task may need to occur prior to the application of feature subset selection. This was the case for the CRM dataset since the discretization approach was utilized in handling the continuous attributes, as discussed earlier in Chapter 5.

The Apriori algorithm was then used to discover frequent rules from the resulting flat table format. While the Apriori algorithm may be applied with some constraints and parameters to reduce the number of rules, nevertheless, many misleading, uninteresting and insignificant rules might still be present as discussed earlier in Chapter 3 and Chapter 6. Thus, similar to the evaluation of relational data in Chapter 5 and Chapter 6, the chi-squared test, logistic regression analysis and redundancy removal are used to find and discard irrelevant input nodes and the rules that contain them, as they are useless for predicting the target class.

However, some rules generated by the *DSM* approach may not always be representatives of valid subtrees (Hadzic, 2011) due to the data mining tasks not being performed by following the structure as discussed in Section 7.4.1.2. The rules that reflect invalid subtrees can be determined by matching them against the *DSM*. However, even if they are representatives of invalid subtrees, they may still contain useful associations. Hence, the quality of the initially extracted subtree sets that contain both valid and invalid (disconnected subtrees) will also be assessed. This set will be referred to as *FullTree*. Thus in these experiments, the initial *FullTree* set is filtered to exclude invalid/disconnected subtrees which results in a rule set based on embedded subtrees. The definitions of embedded and induced subtrees were given in Chapter 3, and generally speaking, in an embedded subtree a parent-child relationship can correspond to an ancestor-descendant relationship in the original tree; while in an induced subtree, all parent-child relationships are exactly preserved. Hence, from the rule set based on embedded subtrees, any subtrees that have a level of embedding between the nodes > 1 are removed, which results in a rule set based on induced subtrees. Hence, the induced rule set $\subseteq$ embedded rule set $\subseteq$ *FullTree* rule set.

The evaluation of the rule sets comprises four main tasks. The first analysis of the usefulness of the rules generated and the sequence of usage of certain parameter is

discussed. The progressive difference in the number of rules, the Accuracy Rate (AR) and the Coverage Rate (CR) values are revealed as the statistical analysis and redundancy assessment method are utilized. The second evaluation is to compare the rules based on the existence of backtrack and non-existence of backtrack in the dataset (*FDT*) (referred to as WithBacktrack or/and WithOutBacktrack). The third analysis is to ascertain the differences between *FullTree*, *Embedded* and *Induced* rule sets based on the AR and CR criteria. In addition, the *XRules* classifier (Zaki & Aggarwal, 2003) which is capable of discovering association rules based on the algorithm for discovering frequent ordered embedded subtrees is used to evaluate the usefulness of the rules discovered from the Framework *B* which applied the *DSM*. The experiments were run on HP Compact Intel(R) Core ™ 2 Duo CPU E6750 @ 2.66Hz 1.97 GHz, 1.96 GB of RAM, 100GB Hard Disk Space and SAS@ Enterprise Miner.

7.7.1 Rules Set Optimization

The first experiment focussed on evaluating the interestingness of frequent subtrees generated from a flat table using the *DSM* application, with steps similar to those of the Framework *A* described in Chapter 5 and Chapter 6. Firstly, in this section we discuss the reduction in the number of rules from the initial frequent rule sets and rule sets after using the Symmetrical Tau (ST) features selection approach. Next, the focus is on the progressive frequent rules verification based on statistical analysis, namely the chi-squared test, logistic regression analysis and redundancy assessment method.

As shown in Tables 7.7 to 7.11, the conversion of the original tree-structured data into the flat data format representation, created a very large number of input attributes, especially at lower support thresholds. By utilizing the Apriori algorithm to generate all frequent rules, one might encounter difficulties in analyzing all rules given certain support and confidence constraints as proven by our experiment for relational data. By referring to the entire Tables 7.20 to Table 7.24, even with the given support constraint, the number of extracted rules (Initial Rule Set) is large and unfeasible to analyze. In addition to that, within the support of 1% for DEBII WebLogs WithBacktrack (Table 7.20) and 5% support value for CRM data (Table

7.24), the initial rules cannot be displayed due to limitations of software and tools. As a remark, for the DEBII WebLogs WithBacktrack, the rules are generated based on the attributes that satisfy both ST and chi-squared test. In this case, the chi-squared test is used to identify and remove any attributes that are not significant in predicting the class value within the dataset. For the CRM dataset, the rules are generated based on the attributes that satisfy the ST measure. Moreover, one can observed that, in CSLogs WithOutBacktrack (Table 7.23) for support values 20% and 30%, no rules are meeting the given thresholds.

A large volume of rules may be discovered due to the presence of irrelevant attributes in the dataset. The capabilities of ST in selecting appropriate attributes, thereby removing irrelevant attributes, are shown in our previous experiments for relational data problems. For this particular task of evaluating tree-structured rules, similar experiments were conducted. The attributes for each different support were ranked according to their decreasing ST and a relevance cut-off point was chosen.

Table 7.18 indicates the differences between the number of initial input attributes and the number of attributes after applying Symmetrical Tau (ST) with their respective rule number (below) for each dataset for each different support. All attributes that have been removed from DEBII WebLogs WithBacktrack and CSLogs WithBacktrack are backtrack attributes; moreover, in CSLogs WithBacktrack the ST also removed one label node of X_0 which in the context of CSLogs data is a single value attribute. While for the DEBII WithOutBacktrack, no attributes were removed and only 1 attribute was removed from CSLogs WithOutBacktrack which is the same X_0 attribute in CSLogs WithBacktrack.

This indicates that the inclusion of these backtrack nodes may not be useful or have low capabilities in predicting the class attributes in this dataset. As expected, the input variable that contains a single value is unable to distinguish the class variables. Such input attributes have been discarded as they are considered irrelevant based on the ST value calculated. With the application of ST feature selection technique, rules that contain attributes that failed the ST measure are discarded. Furthermore, in two specific cases as shown in Table 7.18 for CSLogs WithBacktrack with support 20%

and 30%, as the ST measures been utilized; consequently, all rules have been discarded as these rules consist of attributes that failed the ST measure.

Table 7.18: Number of Attributes Removes By ST and Respective Number of Rules

		WithBacktrack					**WithOutBacktrack**				
		1%	**5%**	**10%**	**20%**	**30%**	**1%**	**5%**	**10%**	**20%**	**30%**
DEBII WebLogs	# of Initial Input Attr.	441	125	69	35	25	221	63	35	18	13
	(# of Initial Rules)	(*)	(28282)	(234)	(50)	(14)	(308)	(17)	(8)	(2)	(1)
	# of Attr. After ST	437	123	63	29	19	221	63	35	18	13
	(# of Rules After ST)	(*)	(8031)	(43)	(8)	(3)	(308)	(17)	(8)	(2)	(1)
CSLogs	# of Initial Input Attr.	221	63	39	23	15	126	42	23	16	11
	(# of Initial Rules)	(13835)	(920)	(216)	(48)	(32)	(144)	(8)	(4)	(-)	(-)
	# of Attr. After ST	217	52	29	11	7	125	41	22	15	10
	(# of Rules After ST)	(6084)	(99)	(25)	(-)	(-)	(72)	(4)	(2)	(-)	(-)

(*) = Unavailable

(-) = No rules

The next task in measuring the interestingness of frequent subtrees is the application of the chi-squared test and logistic regression analysis for rule verification. The statistical analysis is applied in this framework in order to determine the usefulness and significance of input variables in predicting the class variables as was the case in relational data problems. Hence, this will provide an appropriate means of identifying and discarding rules that are not significant. However, to some extent some implications and issues arose when a similar approach was applied to the frequent subtrees extracted from the flat data format.

To illustrate this, an example from CSLogs WithBacktrack with 1% support is used. As all the input attributes were included in the SAS@ software, the statistical measurements for large-valued attributes cannot be performed because the software limits the maximal number of unique values to 125. Hence, one additional assumption included here is that any input attributes exceeding a certain limit of the number of its possible values will be omitted. For example, in this dataset, input attribute X_1 is being omitted from further analysis as it consists of 535 unique values. There are some implications behind this assumption, and alternatives need to be

taken in the pre-processing stage. However, this is left as future work and the related discussion is provided in Chapter 8. After the application of the chi-squared test, 182 attributes failed the chi-squared test and were removed. Another 13 attributes were also rejected because they exceeded the value's limit per attribute which is considered unimportant based on the software used. The remaining 18 input attributes were selected for the logistic regression analysis. Table 7.19 presents the summary of attributes for CSLogs Data WithBacktrack for 1%, while Figure 7.11 displays a screen shot of the chi-squared test results.

Table 7.19: Summary of Attributes for CSLogs Data WithBacktrack

Total # of Input Attr.	**# of Class Attr.**	**# of Input Attr. After ST**	**# of Attr. With Exceeding Limit Values per Attr.**	**# of Attr. with Exceeding Chi Squared Limit**	**# of Attr. with Small Chi Squared Value**	**# of Attr. for Logistic Regression analysis**
221	1	217	13	4	182	18

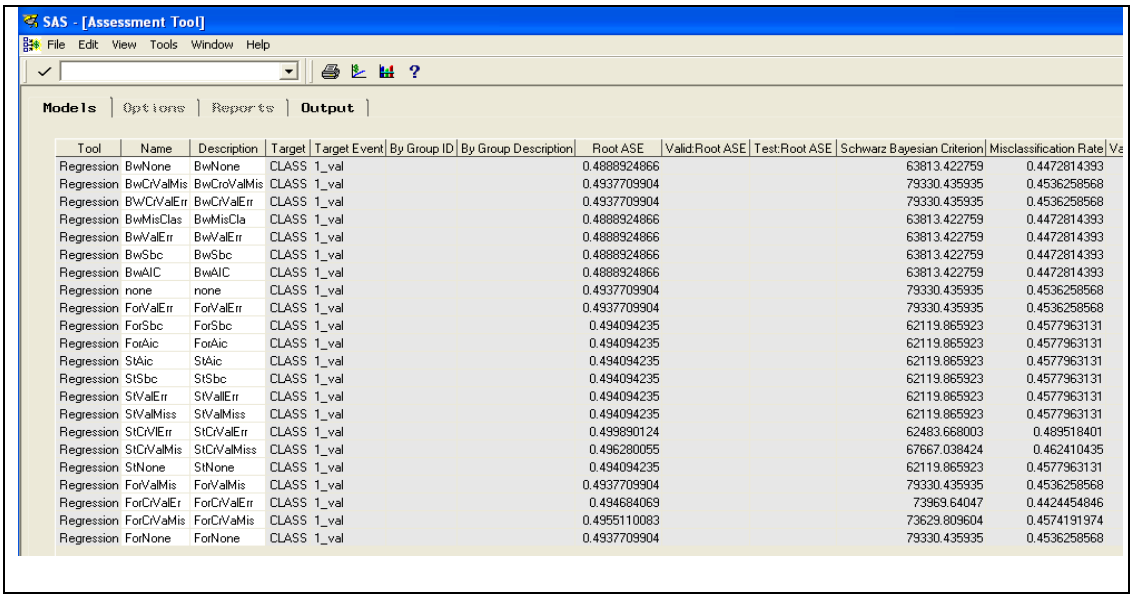
Name	Role	Rejection Reason	# of Levels
X6	input		67
X7	input		34
X13	input		14
X14	input		12
X15	input		6
X16	input		4
B5	input		2
X55	input		54
X71	input		17
X77	input		33
X78	input		10
X80	input		28
X82	input		26
X84	input		18
X85	input		11
X86	input		9
B94	input		2
X102	input		43
X21	rejected	Levels exceed set maximum	110
X31	rejected	Levels exceed set maximum	87
X60	rejected	Levels exceed set maximum	67
X67	rejected	Levels exceed set maximum	69
X8	rejected	Small chi-square	30
X9	rejected	Small chi-square	24
X10	rejected	Small chi-square	22
X11	rejected	Small chi-square	16

Figure 7.11: Variables Selection based on Chi-squared test for CSLogs WithBacktrack for Support 1%

Several logistic regression models were then developed for the remaining 18 input attributes. The selected model is the model that proves to be the most parsimonious with the lowest misclassification rate. This result is possible because different

variable combinations may contain different/complementary information that contributes to the prediction of the value of the target variable.

The selected model consists of these attributes: "X_{102}, X_{13}, X_{14}, X_{16}, X_{16},X_{55}, X_6, X_7, X_{71}, X_{77}, X_{78}, X_{80}, X_{82}, X_{84}, X_{85}, X_{86}" However, as one can observe in the developed models in Figure 7.12, the misclassification rate for the each model is within the range of 44% - 45%. Compared to all results for the logistic regression models for relational data problems, this misclassification rate is lower, thereby possibly having further implications for the subtrees verification process. This may be due to the existence of sparse values within each attribute because of the greater variation among users who accessed the CSLogs's server. In this particular case for this kind of data, further analysis needs be done to determine an appropriate measure that can handle the many values within an attribute. These issues are reserved for our future work as discussed in Chapter 8.



Tool	Name	Description	Target	Target Event	By Group ID	By Group Description	Root ASE	Valid:Root ASE	Test:Root ASE	Schwarz Bayesian Criterion	Misclassification Rate	Va
Regression	BwNone	BwNone	CLASS	1_val			0.4888924866			63813.422759	0.4472814393	
Regression	BwCrValMis	BwCroValMis	CLASS	1_val			0.4937709904			79330.435935	0.4536258568	
Regression	BWCrValErr	BwCrValErr	CLASS	1_val			0.4937709904			79330.435935	0.4536258568	
Regression	BwMisClas	BwMisCla	CLASS	1_val			0.4888924866			63813.422759	0.4472814393	
Regression	BwValErr	BwValErr	CLASS	1_val			0.4888924866			63813.422759	0.4472814393	
Regression	BwSbc	BwSbc	CLASS	1_val			0.4888924866			63813.422759	0.4472814393	
Regression	BwAIC	BwAIC	CLASS	1_val			0.4888924866			63813.422759	0.4472814393	
Regression	none	none	CLASS	1_val			0.4937709904			79330.435935	0.4536258568	
Regression	ForValErr	ForValErr	CLASS	1_val			0.4937709904			79330.435935	0.4536258568	
Regression	ForSbc	ForSbc	CLASS	1_val			0.494094235			62119.865923	0.4577963131	
Regression	ForAic	ForAic	CLASS	1_val			0.494094235			62119.865923	0.4577963131	
Regression	StAic	StAic	CLASS	1_val			0.494094235			62119.865923	0.4577963131	
Regression	StSbc	StSbc	CLASS	1_val			0.494094235			62119.865923	0.4577963131	
Regression	StValErr	StValErr	CLASS	1_val			0.494094235			62119.865923	0.4577963131	
Regression	StValMiss	StValMiss	CLASS	1_val			0.494094235			62119.865923	0.4577963131	
Regression	StCrVlErr	StCrValErr	CLASS	1_val			0.499890124			62483.668003	0.489518401	
Regression	StCrValMis	StCrValMiss	CLASS	1_val			0.496280055			67667.038424	0.462410435	
Regression	StNone	StNone	CLASS	1_val			0.494094235			62119.865923	0.4577963131	
Regression	ForValMis	ForValMis	CLASS	1_val			0.4937709904			79330.435935	0.4536258568	
Regression	ForCrValEr	ForCrValErr	CLASS	1_val			0.494684069			73969.64047	0.4424454846	
Regression	ForCrVaMis	ForCrVaMis	CLASS	1_val			0.4955110083			73629.809604	0.4574191974	
Regression	ForNone	ForNone	CLASS	1_val			0.4937709904			79330.435935	0.4536258568	

Figure 7.12: Logistic Regression Model Selection for CSLogs WithBacktrack for Support 1%

Redundant rules assessment is then performed in determining the existence of any redundant rules in the database. As defined by (Bayardo, Agrawal, & Gunopulos, 2000; Webb, 2007), redundant rules are those rules that include items in the antecedent that are entailed by the other elements of the antecedents. Of all datasets in Table 7.20 to Table 7.24 (the number of rules is shown in brackets), there were a

number of rules detected as redundant based on the application of the redundancy removal with 1% support for DEBII WebLogs and CSLogs; and 5% support for CRM dataset. This is due to the large number of existing rules even when the confidence was constrained to 50%.

These large numbers of rules managed to be reduced with a proper sequence of usage of parameters including the ST feature selection, statistical analysis and the redundancy assessment method. For this experiment, and with reference to the specific datasets, the reduction of the number of rules increased the AR for DEBII WebLogs WithBacktrack for 10%, 20% and 30% support; CSLogs WithBacktrack 1%, 5% and 10% but this produces a reduction in CR capabilities.

One may observe that, for DEBII WebLogs WithBacktrack for 1% and 5% support and DEBII WebLogs WithOutBacktrack for 1%, 5% and 10% support; CSLogs WithOutBacktrack for 1%, both AR and CR are slightly reduced with the reduction in the number of rules. Moreover, there are no rules available to be evaluated once the ST is used for the CSLogs WithBacktrack and CSLogs WithOutBacktrack for 20% and 30% support.

However, for CSLogs WithOutBacktrack with 5% and 10% support thresholds, the AR and CR are both preserved even with the reduced number of rules. However, there are no rules to be removed for DEBII WebLogs WithOutBacktrack with 20% and 30% support, thereby preserving the same AR and CR values.

Several variations occurred within the experiments for each different dataset and for each different support. One might observe that the results are not consistent throughout the entire experiments. This might be a case as noted by (Hadzic & Hecker, 2011) of there being differences between period of web access between DEBII WebLogs (4 months) and CSLogs (3 weeks) which may account for the inconsistency in results. However, one can observe that, in most of the cases demonstrated earlier, the reduction of rules have increased their prediction/classification ability but weakened their ability to capture/cover more instances in the datasets.

Furthermore, as two web logs access data were utilized in this experiment (the CRM dataset will be discussed in Section 7.7.5), the result demonstrated that, because of the large navigational variants between users who accessed different web pages from the website, this consequently created sparse values within attributes. This will affect the application of statistical analysis including the chi-squared test and logistic regression analysis. Thus, one may need to limit the number of maximum values for each attribute, or group the values under certain categories, if meaningful categories can be devised. However, this issue is outside of the scope of the work of this thesis.

Table 7.20: DEBII WebLogs WithBacktrack

Type of analysis	**Data Partition**	**Support : 1%**		**Support : 5%**		**Support : 10%**		**Support : 20%**		**Support : 30%**	
		AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
Initial # of Rules	Training	*	*	90.61 (28282)	100.00 (28282)	64.27 (234)	100.00 (234)	58.36 (50)	100.00 (50)	59.62 (14)	100.00 (14)
	Testing			94.41 (28282)	100 (28282)	70.06 (234)	100.00 (234)	62.01 (50)	100.00 (50)	57.90 (14)	100.00 (14)
# of Rules after ST	Training	*	*	90.50 (8031)	100.00 (8031)	75.19 (43)	73.95 (43)	68.60 (8)	64.20 (8)	68.36 (3)	59.16 (3)
	Testing			94.26 (8031)	100.00 (8031)	74.94 (43)	74.09 (43)	72.48 (8)	57.23 (8)	74.57 (3)	51.95 (3)
Chi Squared	Training	83.13 (528)	94.03 (528)	73.97 (22)	91.26 (22)	78.21 (11)	64.47 (11)	74.08 (3)	59.32 (3)	75.81 (1)	38.64 (1)
	Testing	80.23 (528)	93.75 (528)	68.77 (22)	91.68 (22)	74.96 (11)	60.12 (11)	75.51 (3)	53.20 (3)	80.49 (1)	35.77 (1)
Logistic Regression	Training	82.62 (496)	94.03 (496)								
	Testing	80.02 (496)	93.75 (496)								
Redundancy Removal	Training	78.52 (189)	94.03 (189)								
	Testing	75.06 (189)	93.75 (189)								

(*) = Unavailable

Table 7.21: DEBII WebLogs WithOutBacktrack

Type of analysis	Data Partition	Support : 1%		Support : 5%		Support : 10%		Support : 20%		Support : 30%	
		AR %	CR %	AR %	CR %	AR %	CR %	AR %	CR %	AR %	CR %
Initial # of Rules	Training	86.04 (308)	82.70 (308)	80.72 (17)	68.13 (17)	79.40 (8)	56.30 (8)	74.36 (2)	53.43 (2)	75.81 (1)	38.64 (1)
	Testing	83.66 (308)	72.08 (308)	77.90 (17)	59.21 (17)	76.12 (8)	49.92 (8)	75.84 (2)	47.17 (2)	80.49 (1)	35.77 (1)
# of Rules after ST	Training	86.04 (308)	82.70 (308)	80.72 (17)	68.13 (17)	79.40 (8)	56.30 (8)	74.36 (2)	53.43 (2)	75.81 (1)	38.64 (1)
	Testing	83.66 (308)	72.08 (308)	77.90 (17)	59.21 (17)	76.12 (8)	49.92 (8)	75.84 (2)	47.17 (2)	80.49 (1)	35.77 (1)
Chi Squared	Training	86.04 (308)	82.70 (308)	80.72 (17)	68.13 (17)	79.40 (8)	56.30 (8)				
	Testing	83.66 (308)	72.08 (308)	77.90 (17)	59.21 (17)	76.12 (8)	49.92 (8)				
Logistic Regression	Training	84.68 (260)	82.70 (260)	78.72 (10)	62.19 (10)	76.61 (4)	54.50 (4)				
	Testing	83.13 (260)	72.08 (260)	78.18 (10)	53.86 (10)	75.42 (4)	48.20 (4)				
Redundancy Removal	Training	82.18 (140)	82.70 (140)								
	Testing	80.51 (140)	72.08 (140)								

Table 7.22: CSLogs WithBacktrack

Type Of analysis	Data Partition	Support : 1%		Support : 5%		Support : 10%		Support : 20%		Support : 30%	
		AR %	CR %	AR %	CR %	AR %	CR %	AR %	CR %	AR %	CR %
Initial # of Rules	Training	68.09 (13835)	98.59 (13835)	62.79 (920)	98.33 (920)	60.14 (216)	98.32 (216)	56.49 (48)	91.65 (48)	56.39 (32)	91.65 (32)
	Testing	69.94 (13835)	98.60 (13835)	63.58 (920)	98.35 (920)	60.73 (216)	98.34 (216)	57.10 (48)	91.45 (48)	57.13 (32)	91.45 (32)
# of Rules after ST	Training	69.94 (6084)	98.59 (6084)	64.71 (99)	98.33 (99)	60.98 (25)	98.32 (25)				
	Testing	72.01 (6084)	98.60 (6084)	65.47 (99)	98.35 (99)	61.66 (25)	98.34 (25)				
Chi Squared	Training	79.22 (73)	48.97 (73)	66.04 (11)	74.81 (11)	62.16 (8)	81.53 (8)				
	Testing	78.78 (73)	48.77 (73)	65.94 (11)	74.60 (11)	62.45 (8)	81.49 (8)				
Logistic Regression	Training	79.22 (73)	48.97 (73)								
	Testing	78.78 (73)	48.77 (73)								
Redundancy Removal	Training	79.01 (61)	48.97 (61)								
	Testing	78.53 (61)	48.77 (61)								

Table 7.23: CSLogs WithOutBacktrack

Type of analysis	Data Partition	Support : 1%		Support : 5%		Support : 10%		Support : 20%		Support : 30%	
		AR %	CR %	AR %	CR %	AR %	CR %	AR %	CR %	AR %	CR %
Initial # of Rules	Training	79.25 (144)	48.31 (144)	79.31 (8)	20.40 (8)	76.69 (4)	20.36 (4)	No Rules		No Rules	
	Testing	78.77 (144)	48.01 (144)	78.73 (8)	20.35 (8)	76.90 (4)	20.30 (4)				
# of Rules after ST	Training	79.25 (72)	48.31 (72)	79.31 (4)	20.40 (4)	76.69 (2)	20.35 (2)				
	Testing	78.77 (72)	48.01 (72)	78.73 (4)	20.35 (4)	76.90 (2)	20.30 (2)				
Chi Squared	Training	79.25 (72)	48.31 (72)								
	Testing	78.77 (72)	48.01 (72)								
Logistic Regression	Training	79.25 (72)	48.31 (72)								
	Testing	78.77 (72)	48.01 (72)								
Redundancy Removal	Training	79.05 (60)	48.31 (60)								
	Testing	78.53 (60)	48.01 (60)								

Table 7.24: CRM Data WithBacktrack

Type of analysis	Data Partition	Support : 5%		
		# Rules	AR %	CR %
Initial # of Rules	Training	*	*	*
	Testing			
# of Rules after ST	Training	27116	83.02	100.00
	Testing		83.74	100.00
Chi Squared	Training	91	79.85	100.00
	Testing		80.95	100.00
Redundancy Removal	Training	51	76.78	100.00
	Testing		77.72	100.00

(*) = Unavailable

7.7.2 Comparing Rules with Backtrack and Rules without Backtrack

By referring to the DEBII WebLogs and CSLogs dataset characteristics in Tables 7.7 to 7.10, one can see that if there are *x* non-backtrack attributes in the dataset WithOutBacktrack, there will be *x-1* additional backtrack attributes in the dataset WithBacktrack. Hence, the number of rules from the datasets WithBacktrack is always higher compared to rules within dataset WithOutBacktrack. All initial rules are discovered for every dataset for different support. However, there are no initial rules discovered for DEBII Weblogs data WithBacktrack with 1% support (Refer to

Table 7.26). This is due to the large number of attributes (441 attributes for Initial Rules and 437 attributes after applying Symmetrical Tau and discarding irrelevant attributes - Refer to Table 7.18) and the limitation of memory of our software and tools.

For DEBII Weblogs data (Refer to Table 7.26 to Table 7.30), the coverage rate (CR) for rules WithBacktrack is always higher compared to the CR for rules WithOutBacktrack. In contrast, the accuracy rate (AR) for rules WithOutBacktrack is higher compared to rule set WithBacktrack. Table 7.25 gives an example of the final rules from both WithBacktrack and WithOutBacktrack from DEBII Weblogs data with 20% support to demonstrate the aforementioned issues and implications.

Table 7.25: Comparison between DEBII WebLogs *FullTree* WithBacktrack and WithOutBacktrack for 20% Support

***FullTree* WithBacktrack**	***FullTree* WithOutBacktrack**
X0(7) ==> Class(1)	X0(7) ==> Class(1)
B11(yes) ==> Class(1)	X1(7) ==> Class(0)
X1(7) ==> Class(0)	

The final 3 rules and 2 rules from *FullTree* are the significant rules based on the unification framework as discussed earlier in Section 7.7.1. The CR in the testing set for the 3 final rules from rule set WithBacktrack is 53.20% which is higher compared to the 2 final rules from rule set WithOutBacktrack of 47.17%. The AR for the 3 final rules from rules WithBacktrack testing set is 75.51% which is slightly less than the AR for the 2 final rules in the dataset WithOutBacktrack for 75.84%. This observation indicates that the rules can capture more instances in the dataset when a backtrack node is included. Each node in a pre-order string encoding upon which *DSM* is based, implies the existence of a specific backtrack attribute in the string encoding. Rules indicating existence/non-existence of a backtrack attribute also indicate the existence of a particular node in the structure irrespective of that node's actual value. Such rules may be useful in cases when the specific values cannot become part of a rule because they do not satisfy the specified support threshold (20% in this example). However, it can come with a trade-off that the rules WithBacktrack are not specific enough, thereby reducing the AR.

Additionally, one can observe that in DEBII Weblogs WithBacktrack for support 5% (refer to Table 7.27) both AR and CR for the initial rule set and rule set after the ST application is higher compared to the initial rule set and rule set after ST application extracted from WithOutBacktrack dataset variation. This exception occurs due to the large number of rules been discovered from the dataset WithBacktrack thus creating possibility that the initial rules and rules after ST are capable of capturing every instance in the database and still maintain a high level of accuracy. However, as emphasized in our evaluation framework, the initial set of rules may be contaminated with insignificant, random and redundant rules. Even their AR and CR are higher, creating possible difficulties in interpreting, handling and presenting the rules.

Table 7.26: The comparison between rules for DEBII WebLogs WithBacktrack and DEBII WebLogs WithOutBacktrack using 1% support

Type of analysis	**Data Partition**	**WithBacktrack**		**WithOutBacktrack**	
		AR %	**CR %**	**AR %**	**CR %**
Initial # of Rules	Training	*	*	86.04 (308)	82.70 (308)
	Testing			83.66 (308)	72.08 (308)
# of Rules after ST	Training	*	*	86.04 (308)	82.70 (308)
	Testing			83.66 (308)	72.08 (308)
Chi Squared	Training	83.13 (528)	94.03 (528)	86.04 (308)	82.70 (308)
	Testing	80.23 (528)	93.75 (528)	83.66 (308)	72.08 (308)
Logistic Regression	Training	82.62 (496)	94.03 (496)	84.68 (260)	82.70 (260)
	Testing	80.02 (496)	93.75 (496)	83.13 (260)	72.08 (260)
Redundancy Removal	Training	78.52 (189)	94.03 (189)	82.18 (140)	82.70 (140)
	Testing	75.06 (189)	93.75 (189)	80.51 (140)	72.08 (140)

(*) = Unavailable

Table 7.27: The comparison between rules DEBII WebLogs WithBacktrack and DEBII WebLogs WithOutBacktrack using 5% support

Type of analysis	**Data Partition**	**WithBacktrack**		**WithOutBacktrack**	
		AR %	**CR %**	**AR %**	**CR %**
Initial # of Rules	Training	90.61 (28282)	100.00 (28282)	80.72 (17)	68.13 (17)
	Testing	94.41 (28282)	100.00 (28282)	77.90 (17)	59.21 (17)
# of Rules after ST	Training	90.50 (8031)	100.00 (8031)	80.72 (17)	68.13 (17)
	Testing	94.26 (8031)	100.00 (8031)	77.90 (17)	59.21 (17)
Chi Squared	Training	73.97 (22)	91.26 (22)	80.72 (17)	68.13 (17)
	Testing	68.77 (22)	91.68 (22)	77.90 (17)	59.21 (17)
Logistic Regression	Training			78.72 (10)	62.19 (10)
	Testing			78.18 (10)	53.86 (10)
Redundancy Removal	Training				
	Testing				

Table 7.28: The comparison between rules DEBII WebLogs WithBacktrack and DEBII WebLogs WithOutBacktrack using 10% support

Type of analysis	Data Partition	WithBacktrack		WithOutBacktrack	
		AR %	CR %	AR %	CR %
Initial # of Rules	Training	64.27 (234)	100.00 (234)	79.40 (8)	56.30 (8)
	Testing	70.06 (234)	100.00 (234)	76.12 (8)	49.92 (8)
# of Rules after ST	Training	75.19 (43)	73.95 (43)	79.40 (8)	56.30 (8)
	Testing	74.94 (43)	74.09 (43)	76.12 (8)	49.92 (8)
Chi Squared	Training	78.21 (11)	64.47 (11)	79.40 (8)	56.30 (8)
	Testing	74.96 (11)	60.12 (11)	76.12 (8)	49.92 (8)
Logistic Regression	Training			76.61 (4)	54.50 (4)
	Testing			75.42 (4)	48.20 (4)
Redundancy Removal	Training				
	Testing				

Table 7.29: The comparison between rules DEBII WebLogs WithBacktrack and DEBII WebLogs WithOutBacktrack using 20% support

Type of analysis	Data Partition	WithBacktrack		WithOutBacktrack	
		AR %	CR %	AR %	CR %
Initial # of Rules	Training	58.36 (50)	100.00 (50)	74.36 (2)	53.43 (2)
	Testing	62.01 (50)	100.00 (50)	75.84 (2)	47.17 (2)
# of Rules after ST	Training	68.60 (8)	64.20 (8)	74.36 (2)	53.43 (2)
	Testing	72.48 (8)	57.23 (8)	75.84 (2)	47.17 (2)
Chi Squared	Training	74.08 (3)	59.32 (3)		
	Testing	75.51(3)	53.20 (3)		
Logistic Regression	Training				
	Testing				
Redundancy Removal	Training				
	Testing				

Table 7.30: The comparison between rules DEBII WebLogs WithBacktrack and DEBII WebLogs WithOutBacktrack using 30% support

Type of analysis	Data Partition	WithBacktrack		WithOutBacktrack	
		AR %	CR %	AR %	CR %
Initial # of Rules	Training	59.62 (14)	100.00 (14)	75.81 (1)	38.64 (1)
	Testing	57.90 (14)	100.00 (14)	80.49 (1)	35.77 (1)
# of Rules after ST	Training	68.36 (3)	59.16 (3)	75.81 (1)	38.64 (1)
	Testing	74.57 (3)	51.95 (3)	80.49 (1)	35.77 (1)
Chi Squared	Training	75.81 (1)	38.64 (1)		
	Testing	80.49 (1)	35.77 (1)		
Logistic Regression	Training				
	Testing				
Redundancy Removal	Training				
	Testing				

As for CSLogs data, regarding the final rules of each dataset for each different support, one can observe that the number of rules extracted from WithBacktrack dataset is always higher. In comparing the AR and CR within the CSLogs dataset, and the similar pattern as for DEBII WebLogs data also occurs here. It was found that in the final rule sets with support of 1%, 5% and 10%, the CR is higher and the AR is lower for the *FullTree* WithBacktrack, while in contrast, the AR is always higher and CR is lower for *FullTree* WithOutBacktrack.

Table 7.31: The comparison between rules CSLogs WithBacktrack and CSLogs WithOutBacktrack using 1% support

Type of analysis	**Data Partition**	**WithBacktrack**		**WithOutBacktrack**	
		AR %	**CR %**	**AR %**	**CR %**
Initial # of Rules	Training	68.09 (13835)	98.59 (13835)	79.25 (144)	48.31 (144)
	Testing	69.94 (13835)	98.60 (13835)	78.77 (144)	48.01 (144)
# of Rules after ST	Training	69.94 (60.84)	98.59 (60.84)	79.25 (72)	48.31 (72)
	Testing	72.01 (6084)	98.60 (6084)	78.77 (72)	48.01 (72)
Chi Squared	Training	79.22 (73)	48.97 (73)	79.25 (72)	48.31 (72)
	Testing	78.78 (73)	48.77 (73)	78.77 (72)	48.01 (72)
Logistic Regression	Training	79.22 (73)	48.97 (73)	79.25 (72)	48.31 (72)
	Testing	78.78 (73)	48.77 (73)	78.77 (72)	48.01 (72)
Redundancy Removal	Training	79.02 (61)	48.97 (61)	79.05 (60)	48.31 (60)
	Testing	78.53 (61)	48.77 (61)	78.53 (60)	48.01 (60)

Table 7.32: The comparison between rules CSLogs WithBacktrack and CSLogs WithOutBacktrack using 5% support

Type of analysis	**Data Partition**	**WithBacktrack**		**WithOutBacktrack**	
		AR %	**CR %**	**AR %**	**CR %**
Initial # of Rules	Training	62.79 (920)	98.33 (920)	79.31 (8)	20.40 (8)
	Testing	63.58 (920)	98.35 (920)	78.73 (8)	20.35 (8)
# of Rules after ST	Training	64.71 (99)	98.33 (99)	79.31 (4)	20.40 (4)
	Testing	65.47 (99)	98.35 (99)	78.73 (4)	20.35 (4)
Chi Squared	Training	66.04 (11)	74.81 (11)		
	Testing	65.94 (11)	74.60 (11)		
Logistic Regression	Training				
	Testing				
Redundancy Removal	Training				
	Testing				

Table 7.33: The comparison between rules CSLogs WithBacktrack and CSLogs WithOutBacktrack using 10% support

Type of analysis	Data Partition	WithBacktrack		WithOutBacktrack	
		AR %	CR %	AR %	CR %
Initial # of Rules	Training	60.14 (216)	98.32 (216)	76.69 (4)	20.36 (4)
	Testing	60.73 (216)	98.34 (216)	76.90 (4)	20.30 (4)
# of Rules after ST	Training	60.98 (25)	98.32 (25)	76.69 (2)	20.35 (2)
	Testing	61.66 (25)	98.34 (25)	76.90 (2)	20.30 (2)
Chi Squared	Training	62.16 (8)	81.53 (8)		
	Testing	62.45 (8)	81.49 (8)		
Logistic Regression	Training				
	Testing				
Redundancy Removal	Training				
	Testing				

Table 7.34: The comparison between rules CSLogs WithBacktrack and CSLogs WithOutBacktrack using 20% support

Type of analysis	Data Partition	WithBacktrack		WithOutBacktrack	
		AR %	CR %	AR %	CR %
Initial # of Rules	Training	56.49 (48)	91.65 (48)	No Rules	
	Testing	57.10 (48)	91.45 (48)		
# of Rules after ST	Training				
	Testing				
Chi Squared	Training				
	Testing				
Logistic Regression	Training				
	Testing				
Redundancy Removal	Training				
	Testing				

Table 7.35: The comparison between rules CSLogs WithBacktrack and CSLogs WithOutBacktrack using 30% support

Type of analysis	Data Partition	WithBacktrack		WithOutBacktrack	
		AR %	CR %	AR %	CR %
Initial # of Rules	Training	56.39 (32)	91.65 (32)	No Rules	
	Testing	57.13 (32)	91.45 (32)		
# of Rules after ST	Training				
	Testing				
Chi Squared	Training				
	Testing				
Logistic Regression	Training				
	Testing				
Redundancy Removal	Training				
	Testing				

As indicated by both DEBII Weblogs and CSLogs datasets, in most of the cases, the final rule set WithBacktrack may increase the CR, but reduce some AR. This is because the rules extracted from the WithBacktrack dataset variations can be larger in number and contain rules based on backtrack attributes, which essentially constrain the existence/non-existence of a node irrespective of the label. Such rules are more general than those rules constraining the existence of a specific label at a node, and consequently can capture more instances from the dataset. However, in some cases (e.g. support = 1%), the differences in AR between rule sets from WithBacktrack and WithOutBacktrack data are too minimal to completely discard one option in favour of another. Thus, there is a trade-off between rules more capable of correctly classifying/predicting class variables thus obtaining the higher AR, and rules that cover more instances, thus achieving a higher CR. Therefore, the choice to include/exclude backtrack attributes from the dataset and hence the rule set, may be dependent on the application domain.

7.7.3 *FullTree, Embedded* and *Induced* Subtree Rules

In Section 7.4.1, the characteristics of the *DSM* approach are overviewed. The frequent rules generated using *DSM* approach might be representatives of invalid/disconnected subtrees. The implications and the way in which disconnected subtrees can be represented with respect to the extracted *database structure model* (*DSM*) were discussed in Section 7.4.1.2.

Thus, in this section, the rules from *FullTree, Embedded* and *Induced* rule sets are compared in terms of their accuracy and coverage rate. The *FullTree* consist of the largest number of rules (including rules based on disconnected subtrees), followed by *Embedded* (rules based on embedded subtrees) and *Induced* (rules based on induced subtrees). This is because the rules generated from embedded and induced subtrees are constrained by the following characteristics: in an embedded subtrees the ancestor-descendant relationship is preserved over several levels at the parent-child relationship, while in an induced subtree the parent-child relationships are preserved. Hence, the *Induced* rule set $\subseteq$ *Embedded* rule set $\subseteq$ *FullTree* rule set. Note also that these rule sets can both vary depending on whether the backtrack attributes were left

in the dataset. When they are left in the datasets, the rules can potentially be comprised of constraints on the existence/non-existence of a backtrack attribute.

7.7.3.1 *FullTree, Embedded* and *Induced* Rules Optimization based on the Sequences of Usage of Parameters

The evaluation of a *FullTree* rule set was discussed earlier in Section 7.7.1. In this section, similar to the experiments described in Section 7.7.1, rules from *Embedded* and *Induced* rule sets have been progressively assessed with statistical analysis and redundancy assessment method. The results demonstrate that characteristics revealed from *FullTree* rule sets in Section 7.7.1 are similar to the ones observed in these experiments for both *Embedded* and *Induced* rule sets. With reference to Table 7.36, with the reduction of number of rules for *Embedded* and *Induced* rule sets for DEBII Weblogs WithBacktrack (10% Support) the AR are increased but at the cost of a decrease in CR. One can also notice that the AR for the *FullTree* rule set is initially slightly lower than the AR of the *Embedded* and *Induced* rule set, but after Symmetrical Tau is applied, the accuracy of *FullTree* is higher and remains higher after chi-squared rule filtering.

Table 7.36: The comparison between *FullTree, Embedded* and *Induced* Rule Sets using 10% Support value for DEBII Weblogs WithBacktrack

Type of analysis	**Data Partition**	**WithBacktrack**					
		FullTree		***Embedded***		***Induced***	
		AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
Initial # of Rules	Training	64.27 (234)	100.00 (234)	64.54 (232)	100.00 (232)	64.54 (232)	100.00 (232)
	Testing	70.06 (234)	100.00 (234)	70.55 (232)	100.00 (232)	70.55 (232)	100.00 (232)
# of Rules After ST	Training	75.19 (43)	73.95 (43)	74.94 (42)	73.95 (42)	74.94 (42)	73.95 (42)
	Testing	74.94 (43)	74.09 (43)	74.84 (42)	74.09 (42)	74.84 (42)	74.09 (42)
Chi Squared	Training	78.21 (11)	64.47 (11)	77.56 (10)	64.47 (10)	77.56 (10)	64.47 (10)
	Testing	74.96 (11)	60.12 (11)	74.58 (10)	61.02 (10)	74.58 (10)	61.02 (10)
Logistic Regression	Training						
	Testing						
Redundancy Removal	Training						
	Testing						

The similar characteristics of the CSLogs dataset for *FullTree* rule sets are also observable in *Embedded* and *Induced* rule sets. As an example, CSLogs WithBacktrack (1% Support) is extracted and displayed in Table 7.37. The result revealed that the reduction of the number of rules from *Embedded* and *Induced* rule sets has increased the AR but at the cost of reduced CR capabilities.

To conclude, the characteristics of the *FullTree* rule set described in Section 7.7.1 are similar to those of the *Embedded* and *Induced* rule sets for each datasets for each different support thresholds, and the accuracy and coverage rates are very similar or the same for the different rule sets. This is because the rules from *Embedded* and *Induced* rule sets are subsets of *FullTree*, and in this dataset there were not so many variations among the rule sets among the level of embedding in subtrees or frequent patterns that produce disconnected subtrees. By selecting important input attributes with ST and evaluating the rules with statistical analysis and redundancy assessment method, there is a reduction in the number of rules.

The increase in prediction/classification accuracy comes with a trade-off since fewer instances are captured from the datasets. On the positive side, a smaller number of discovered rules will increase their generalization power and make it easier for the user to understand and utilize these rules for decision support purposes.

Table 7.37: The comparison between *FullTree, Embedded* and *Induced* Rule Sets using 1% support value for CSLogs WithBacktrack

Type of analysis	**Data Partition**	**WithBacktrack**					
		FullTree		***Embedded***		***Induced***	
		AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
Initial # of Rules	Training	68.09 (13835)	98.59 (13835)	68.12 (13834)	98.59 (13834)	68.11 (13810)	98.59 (13810)
	Testing	69.94 (13835)	98.60 (13835)	69.94 (13834)	98.60 (13834)	69.93 (13810)	98.60 (13810)
# of Rules After ST	Training	69.94 (60.84)	98.59 (60.84)	70.02 (6083)	98.59 (6083)	70.02 (6081)	98.59 (6081)
	Testing	72.01 (6084)	98.60 (6084)	72.10 (6083)	98.60 (6083)	72.10 (6081)	98.60 (6081)
Chi Squared	Training	79.22 (73)	48.97 (73)	79.02 (72)	48.39 (72)	78.41 (65)	48.39 (65)
	Testing	78.78 (73)	48.77 (73)	78.57 (72)	48.25 (72)	78.06 (65)	48.25 (65)
Logistic Regression	Training	79.22 (73)	48.97 (73)	79.02 (71)	48.39 (71)	78.41 (64)	48.39 (64)
	Testing	78.78 (73)	48.77 (73)	78.57 (71)	48.25 (71)	78.06 (64)	48.25 (64)
Redundancy Removal	Training	79.02 (61)	48.97 (61)	78.71 (54)	48.97 (54)	78.71 (54)	48.97 (54)
	Testing	78.53 (61)	48.77 (61)	78.53 (54)	48.77 (54)	78.53 (54)	48.77 (54)

7.7.3.2 Comparing the final rule set of *FullTree, Embedded* and *Induced* Subtree Rules

The final *FullTree* and *Embedded* set of rules extracted from DEBII WebLogs WithBackTrack at 10% support and 50% confidence (Refer to Table 7.36) is displayed in Table 7.38. Please note that the "final" set of rules in this setting corresponds to the rule set remaining after the sequential application of statistical analysis and redundancy assessment method used within the framework. However, the application of certain criteria could result in the complete rule set being removed, and therefore the "final" rule set reflects the last non-empty set of rules. R_7 from *FullTree* is the rule that does not exist in embedded rules as it is disconnected, and hence not part of a valid embedded subtree set.

Table 7.38: Comparison between the final rules for DEBII WebLogs WithBackTrack data with 10% support

#	***FullTree***	#	***Embedded***
R_1	X0(7) ==> Class(1)	R_1	X0(7) ==> Class(1)
R_2	B23(yes) ==> Class(1)	R_2	B23(yes) ==> Class(1)
R_3	X1(7) ==> Class(0)	R_3	X1(7) ==> Class(0)
R_4	X0(13) ==> Class(0)	R_4	X0(13) ==> Class(0)
R_5	X22(7) ==> Class(0)	R_5	X22(7) ==> Class(0)
R_6	B17(yes) ==> Class(1)	R_6	B17(yes) ==> Class(1)
R_7	*X22(7) & X1(7) ==> Class(0)*	R_7	X1(7) & X0(13) ==> Class(0)
R_8	X1(7) & X0(13) ==> Class(0)	R_8	X22(7) & X0(13) ==> Class(0)
R_9	X22(7) & X0(13) ==> Class(0)	R_9	B23(yes) & X0(7) ==> Class(1)
R_{10}	B23(yes) & X0(7) ==> Class(1)	R_{10}	X22(7) & X1(7) & X0(13) ==> Class(0)
R_{11}	X22(7) & X1(7) & X0(13) ==> Class(0)		

In describing the *FullTree* rules and the embedded rules in Table 7.38, R_7 is mapped from *FullTree* onto the extracted *DSM* from DEBII WebLogs for 10% Support (Refer to Table 7.7 - Row 4 Column 3), to generate the pre-order string encoding of subtree as shown in Figure 7.13. While the actual *DSM* for support of 10% contains 70 attributes (including the Class Attribute), many attributes have been omitted as including those would create a deep tree structure that is hard to display.

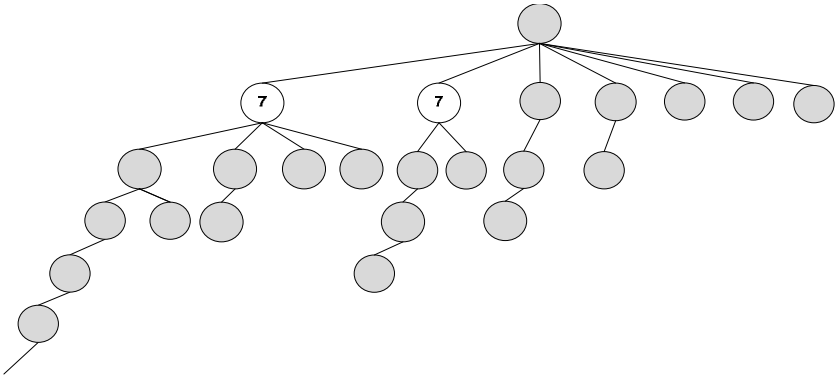


Figure 7.13: Displaying Rules WithBacktrack (R_7 from Table 7.38) with Class 0 w.r.t *DSM* with support 10% that been removed from embedded rules

This rule corresponds to the existence of particular nodes with respect to their occurrence in *DSM* when accessing the DEBII website. The grey nodes represent attributes that did not have specific values that occurred frequently enough with the specific value of "*7*" for nodes X_1 and X_{22}, and they are in that sense irrelevant for rule R_7. The other possibility is that those attributes/nodes have been determined as irrelevant during feature subset selection phase. Hence, one can present the whole

DSM, and flag irrelevant node/s to represent structural information of the rule pattern. Another option would be to display only the nodes that will reveal the structure of the disconnected subtree, i.e. only the root node (X_0) and the two value constraints on nodes X_1 and X_{22}. The rule R_7 from *FullTree* is omitted from embedded rules because the rule is based on a disconnected subtree

For the final rule sets from DEBII WebLogs for testing data (Refer to Table 7.39 to Table 7.43 where number of rules is shown in brackets) for both WithBacktrack and WithOutBacktrack, and for each different support, the AR and CR from *FullTree* rule set are the highest. However, in some cases, as the rules are progressively evaluated, the result demonstrate that the final rules sets for embedded and induced are the same as those of the *FullTree*, thus resulting in the same AR and CR respectively.

Table 7.39: Comparison for the final rule set for DEBII WebLogs Data for 1% Support

DEBII WebLogs Data	***FullTree***		***Embedded***		***Induced***	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
WithBacktrack	75.06 (189)	93.75 (189)	74.71 (179)	93.75 (179)	73.89 (156)	93.75 (156)
WithOutBacktrack	80.51 (140)	72.08 (140)	79.48 (113)	72.08 (113)	78.98 (106)	72.08 (106)

Table 7.40: Comparison for the final rule set for DEBII WebLogs Data for 5% Support

DEBII WebLogs Data	***FullTree***		***Embedded***		***Induced***	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
WithBacktrack	68.77 (22)	91.68 (22)	68.39 (21)	91.68 (21)	68.39 (21)	91.68 (21)
WithOutBacktrack	78.18 (10)	53.86 (10)	78.18 (10)	53.86 (10)	78.18 (10)	53.86 (10)

Table 7.41: Comparison for the final rule set for DEBII WebLogs Data for 10% Support

DEBII WebLogs Data	***FullTree***		***Embedded***		***Induced***	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
WithBacktrack	74.96 (11)	60.12 (11)	74.58 (10)	61.02 (10)	74.58 (10)	61.02 (10)
WithOutBacktrack	75.42 (4)	48.20 (4)	75.42 (4)	48.20 (4)	75.42 (4)	48.20 (4)

Table 7.42: Comparison for the final rule set for DEBII WebLogs Data for 20% Support

DEBII WebLogs Data	***FullTree***		***Embedded***		***Induced***	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
WithBacktrack	75.51 (3)	53.20 (3)	75.51 (3)	53.20 (3)	75.51 (3)	53.20 (3)
WithOutBacktrack	75.84 (2)	47.17 (2)	75.84 (2)	47.17 (2)	75.84 (2)	47.17 (2)

Table 7.43: Comparison for the final rule set for DEBII WebLogs Data for 30% Support

DEBII WebLogs Data	***FullTree***		***Embedded***		***Induced***	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
WithBacktrack	80.49 (1)	35.77 (1)	80.49 (1)	35.77 (1)	80.49 (1)	35.77 (1)
WithOutBacktrack	80.49 (1)	35.77 (1)	80.49 (1)	35.77 (1)	80.49 (1)	35.77 (1)

Similar patterns occurred within the CSLogs dataset (Table 7.44 – Table 7.48), as there is only a slight difference in AR of *FullTree* rule set for support = 1%. However, for the support of 20% and 30% for CSLogs dataset, there are no final rule sets for comparison purposes. Only the initial rule sets for CSLogs WithBacktrack (refer to Table 7.22) are captured, as all of these rules were discarded after ST application. For CSLogs WithOutBacktrack (refer to Table 7.23), no rules could be extracted based on the minimum support and confidence thresholds settings.

Table 7.44: Comparison for the final rule set for CSLogs Data for 1% Support

CSLogs Data	***FullTree***		***Embedded***		***Induced***	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
WithBacktrack	78.53 (61)	48.77 (61)	78.53 (54)	48.77 (54)	78.53 (54)	48.77 (54)
WithOutBacktrack	78.53 (60)	48.01 (60)	78.52 (53)	48.01 (53)	78.52 (53)	48.01 (53)

Table 7.45: Comparison for the final rule set for CSLogs Data for 5% Support

CSLogs Data	***FullTree***		***Embedded***		***Induced***	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
WithBacktrack	65.94 (11)	74.60 (11)	65.94 (11)	74.60 (11)	65.94 (11)	74.60 (11)
WithOutBacktrack	78.73 (4)	20.35 (4)	78.73 (4)	20.35 (4)	78.73 (4)	20.35 (4)

Table 7.46: Comparison for the final rule set for CSLogs Data for 10% Support

CSLogs Data	***FullTree***		***Embedded***		***Induced***	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
WithBacktrack	62.45 (8)	81.49 (8)	62.45 (8)	81.49 (8)	62.45 (8)	81.49 (8)
WithOutBacktrack	76.90 (2)	20.30 (2)	76.90 (2)	20.30 (2)	76.90 (2)	20.30 (2)

Table 7.47: Comparison for the final rule set for CSLogs Data for 20% Support

CSLogs Data	***FullTree***		***Embedded***		***Induced***	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
WithBacktrack	No Rules					
WithOutBacktrack	No Rules					

Table 7.48: Comparison for the final rule set for CSLogs Data for 30% Support

CSLogs Data	***FullTree***		***Embedded***		***Induced***	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
WithBacktrack	No Rules					
WithOutBacktrack	No Rules					

In comparing the number of rules between *FullTree* and *Embedded*, the results revealed that the rules from *FullTree* are discovered based totally on the user defined support and confidence thresholds. In most of the cases, the total number of rules from *FullTree* is larger than embedded and induced rules. While in some cases, the number of rules in *FullTree*, *Embedded* and *Induced*, are the same. One can conclude that for both datasets, the reduced number of rules in embedded subtrees is caused by the unique properties of embedded characteristics which preserve the ancestor-descendant relationship of several levels at the parent-child relationship while this was not the case for the rules from *FullTree*.

To conclude, by referring to the final rule sets for DEBII WebLogs WithBacktrack and WithOutBacktrack for all support thresholds, and CSLogs WithBackTrack and WithOutBackrack with 1%, 5% and 10% support thresholds, the AR and CR for *FullTree* is higher or the same when compared to *Embedded* and *Induced* rule sets, while in some cases, both are similar. This indicates the capabilities of *FullTree* and the unique *DSM* structure-preservation application (Hadzic, 2011) in capturing useful rules that increase the rules' accuracy and coverage. Hence, by default one can aim to discover the *FullTree* set as it will have at least as high an accuracy and coverage rather than when the rules are constrained to be based on embedded or induced subtree. If an important association exists in the data that is not necessarily representative of a valid (connected) subtree, it is still important that it be discovered, which will be the case when considering the *FullTree* rule set.

When comparing *Embedded* and *Induced* rule sets, in Tables 7.39 to 7.46, these are mainly equal except for DEBII WebLogs Data for 1% support where more rules in the *Embedded* set have slightly increased the accuracy. One may expect that the additional rules in embedded rule set may be essential in capturing important associations deeply embedded within the tree structures, i.e. not limited to parent-child relationships as is the case in the *Induced* rule set.

7.7.4 Comparing *FullTree* Classification Results with *XRules*

As an overall comparison, the *XRules* classifier classifier (Zaki & Aggarwal, 2003) is utilized as a benchmark for the proposed framework. The *XRules* approach is a traditional approach for generating frequent subtrees, and hence the rules discovered from *XRules* are based on subtree patterns that are not constrained by the position of the node/nodes. In contrast, within the frequent subtrees extracted from *FDT* using the *DSM*, the node/s positions are preserved in the extracted patterns. The patterns are then mapped to the *DSM* to re-generate the pre-order string encoding of subtrees. With the *XRules* approach, the default confidence threshold of 50% is used while the support thresholds were set at 1%, 5%, 10%, 20% and 30%.

Table 7.49: Comparison of Rules Accuracy and Coverage for DEBII WebLogs Data using the *XRules* and *FullTree* (WithBacktrack dataset/*FDT* variation is used)

	Support: 1%		**Support: 5%**		**Support: 10%**		**Support :20%**		**Support :30%**	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
XRules	79.72 (50000)	88.59 (50000)	78.13 (45)	75.00 (45)	74.50 (14)	70.70 (14)	73.04 (3)	62.40 (3)	71.35 (1)	62.27 (1)
FullTree	80.51 (140)	72.08 (140)	78.18 (10)	53.86 (10)	75.42 (4)	48.20 (4)	75.84 (2)	47.17 (2)	80.49 (1)	35.77 (1)

Refer to support 30% in Table 7.49 (the number of rules is shown in brackets); the final rule is the same for both approaches and it is shown in Table 7.50. The AR in *FullTree* is higher compared to *XRules* but lower in terms of CR. In *FullTree* this specific rule is constrained by the exact node position, i.e. to occur as the first (root) node in the tree (in this dataset it corresponds to the first web page accessed within a user session). Hence, it is more specific than the corresponding rule in the *XRules* approach, as in *XRules* there is no constraint on where a particular node (web page) occurs. As expected, the rule extracted using the *DSM* approach will classify specific instances and hence have higher accuracy, but results in a smaller coverage rate as fewer instances are covered than when there is no constraint on the exact position of a particular node.

Table 7.50: Rules comparison for *Support* 30%

#	***XRules***	*#*	***FullTree***
1	7 ==> Class(1)	1	X0(7) ==> Class(1)

A similar observation can be made for CSLogs dataset in Table 7.51. The AR for rule set in *FullTree* is always higher compared to the rule in *XRules*, although this comes at the cost of lower CR for *FullTree*.

Table 7.51: Comparison of Rules Accuracy and Coverage for CSLogs Data using the *XRules* and *FullTree (*WithBacktrack dataset/*FDT* variation is used)

	Support: 1%		**Support: 5%**		**Support: 10%**		**Support :20%**		**Support :30%**	
	AR %	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**	**AR %**	**CR %**
XRules	72.72 (298)	66.04 (298)	61.74 (20)	40.70 (20)	56.90 (3)	23.21 (3)	Default rules	Default rules	Default Rules	Default Rules
FullTree	78.53 (61)	48.77 (61)	78.73 (4)	20.35 (4)	76.90 (2)	20.30 (2)	No Rules	No Rules	No Rules	No Rules

For the support value of 10% for CSLogs dataset (Refer to Table 7.52), one can observe that the *FullTree* rule set does not contain a rule that corresponds to rule number 3 in *XRules* even though it was considered frequent by *XRules*. The reason for this is that the particular node with label "6" with "Class(0)", where "6" occurs at the same node/position in DSM did not occur in 10% of the instances to be considered frequent and part of the *FullTree* rule set.

Table 7.52: Rules comparison for *Support* 10%

#	***XRules***	**#**	***FullTree***
1	1 ==> Class(0)	1	X1(1) ==> Class(0)
2	12811 ==> Class(1)	2	X1(12811) ==> Class(1)
3	6 ==> Class(0)		

To conclude, given the result demonstrated in both Table 7.49 and Table 7.51, the proposed framework can achieve comparable results with the rules from *XRules* in evaluating the AR and CR from both datasets. It is important to note that there are crucial differences between the *XRules* and the *DSM* approach which makes the results somewhat incompatible for comparison purposes. However, the general framework developed in this thesis could be directly applied to the rules extracted using the *DSM* approach as it utilizes the structure-preserving flat representation. Hence, the comparison performed with the *XRules* approach in this section, served mainly as a benchmark for the kind of accuracy and coverage rate that is to be

obtained when basing the classification on frequent patterns/subtrees extracted using the support and confidence thresholds.

7.7.5 Classification Problems for Customer Relationship Management (CRM)

In the last experiment of evaluating the interestingness of frequent subtree based rules extracted using the *DSM* approach, the evaluation of frequent subtrees is performed for the CRM data which is a real-world problem involving complaint handling in the real estate area.

In this experiment, the steps given in Framework *B* are followed in order to determine the interestingness of frequent subtrees. The complexity of the CRM dataset in XML documents format and general characteristic of CRM data in flat table format are exhibited in Table 7.6 and Table 7.11 (In Section 7.5.1) respectively. In the flat table format, there are 1181 instances in the database; however, the numbers of attributes are 676 which is larger compared to other datasets used in the previous experiments. These include backtrack attributes and attributes with labels. The CRM dataset was pre-processed to make it suitable for the structural classification problem. In here the class attribute is specified into "WorkCompletion", with 2 possible values (work completion $\leq$ 3 or $\geq$ 4 weeks). The attributes containing similar information or referring to work/task completion duration have then been removed. By employing the ST feature selection application, the numbers of attributes have been reduced to 586 input variables. The rules are then generated based on support of 5% and confidence of 50%. As shown in Table 7.53, there are 27116 rules. The statistical analysis has then been used to determine the usefulness and significance of input variables in predicting the target variables. Table 7.53 illustrates the results as the statistical analysis and the redundancy assessment have progressively been utilized to evaluate the interestingness of rules. Based on the results shown, as the CRM dataset underwent conversion into *FDT* using the *DSM* approach, a large volume of input attributes appeared, including the backtrack and label nodes. However, for this particular data in this particular experiment, only the attributes with labels are considered for the statistical analysis and redundancy assessment. Since the backtrack attribute just indicates the existence of nodes irrespective of the label, they can be removed during

the statistical analysis. Additionally, since the structures of the subtrees were preserved using the *DSM,* the rules can be mapped to *DSM* to regenerate the pre-order string encoding of the subtrees.

As can be seen in Table 7.53, in this experiment, *FullTree* rule set is the most optimal one, as it is not only capable of classifying/predicting specific instances in the database, but also achieves a higher coverage rate compared to *Embedded*/*Induced* rule sets. As discussed earlier, this is due to the fact that the *FullTree* rule set can contain rules that do not convert to valid (connected) subtrees when matched to *DSM*. Nevertheless, these are important to include as they may represent important associations that should not be lost because they do not convert to connected valid subtrees. Another interesting feature that has been confirmed by this experiment is the effect of minimum confidence. As verified in relational data problems, and again in this experiment, by increasing the minimum confidence to a certain threshold one may increase the AR but at the same time may lose some CR. One can confirm that, when choosing appropriate optimal confidence values, one may need to trade off in order to either select rules with higher AR but lower CR, or lower AR but higher CR. This could be largely dependent on the application domain, as for example, in more critical domains, one may prefer to use only those rules that have very high confidence, since the misclassification of only one instance could have severe consequences.

Table 7.53: Frequent Subtrees Evaluation for CRM Data

Type of analysis	**Data Partition**	***FullTree***			**Embedded**			**Induced**		
		# Rules	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**	**# Rules**	**AR %**	**CR %**
Initial # of Rules	Training	*	*	*	*	*	*	*	*	*
	Testing									
# of Rules after ST	Training	27116	83.02	100.00	5270	81.56	100.00	5270	81.56	100.00
	Testing		83.74	100.00		83.40	100.00		83.40	100.00
Statistics Analysis	Training	91	79.85	100.00	17	68.54	100.00	17	68.54	100.00
	Testing		80.95	100.00		70.57	100.00		70.57	100.00
Redundancy Removal	Training	51	76.78	100.00	17	68.54	100.00	17	68.54	100.00
	Testing		77.72	100.00		70.57	100.00		70.57	100.00
Min_Conf. 60%	Training	44	83.82	95.50	12	77.20	91.53	12	77.20	91.53
	Testing		84.57	96.15		79.18	93.59		79.18	93.59

(*) = Unavailable

7.8 Conclusion

This chapter extends the framework for evaluating the association rules from relational data to tree-structured or semi-structured data, such as XML. At the start of the chapter, a detailed explanation is provided of the way in which the problem of evaluating frequent subtrees is handled. The results are then presented and discussed after the proposed framework has been applied to several datasets.

Initially, the focus was on evaluating the frequent subtrees discovered by using the traditional frequent subtree mining algorithm, namely the IMB3 Miner. The results demonstrate that, the combination of statistical analysis; and redundancy and contradictive assessment methods provided a means of discarding non-significant rules, which significantly reduces the overall complexities of the rule set. However, these come with a trade-off between obtaining a higher AR but lower CR. Thus, as asserted in Chapter 6, it is important to strike a balance between obtaining a larger set of rules that may lack generalization power, thus weakening the AR, but which are capable of covering more instances, or choosing fewer rules, thereby obtaining higher AR but capturing fewer instances. As the framework uses more complex tree-structured data, the evaluation task using the statistical analysis, and redundancy and contradictive assessment methods becomes infeasible due to the existence of large volume of elements in every transaction/instance. A new direction is then taken to evaluate the interestingness of frequent subtrees using a previously proposed approach for preserving tree-structured data in a flat table format.

Following this new direction, five experiments were undertaken to evaluate the frequent subtrees generated from a flat table using the *DSM* approach. Two variants for each dataset were used. The first variant corresponds to the dataset that contains the backtrack nodes and the second variant corresponds to that without backtrack nodes. The support values for each variant were chosen to be 30%, 20%, 10%, 5% and 1%. In the first experiments, the intention is to optimize the rules based on the sequence of usage of parameters. The results demonstrated that the numbers of rules have been reduced with a proper sequence of usage of parameters including the ST feature selection, statistical analysis and redundancy assessment method. In addition to that, the AR and CR are varied with the reduction of rules based on the aforementioned parameters. In most of the cases, the reduction of rules has increased

their prediction/classification ability but at the same time weakened the capabilities of the rules to capture more instances in the datasets. However, in several cases, both AR and CR are decreased for smaller sizes of rule sets and in two datasets, both AR and CR are well preserved even with the reduced number of rules. Slightly different patterns in results are observed between each dataset for each different support, and this can be attributed to the characteristics of the weblog data used in this experiment. Since the weblogs were collected at different times during which the website itself might be modified (this is the case in DEBIIWebLogs data) this may contribute to the inconsistency of the results.

In the second experiment, the differences between rules with backtrack and rules without backtrack were explored. As discussed earlier, the existence/non-existence of the backtrack attribute indicates the existence of a particular node in the structure irrespective of the actual value/label of that the node. The results of this experiment show that the CR for the final rule sets for rules with backtrack is always higher compared to the CR for the final rules without backtrack. In contrast, the AR for the final rule set for rules without backtrack is higher compared to the rule set with backtrack. With the inclusion of the backtrack node, the rules can capture more instances in the dataset. However, this comes with a trade-off because the rules with backtrack are not specific enough, thereby reducing the AR. It was noted that the differences between AR with rule set with backtrack and without backtrack are still sensible if one prefers to trade off between presenting a set of rules that are more capable of classifying/predicting class variables thus obtaining higher AR with a set of rules that can cover more instances/transactions, thereby obtaining a higher CR.

In general, the *DSM* approach preserves the structural characteristics of frequent patterns, since the extracted patterns are mapped onto the *DSM* to re-generate the pre-order string encoding of subtrees, and hence represented as subtrees of the tree database. However, in some cases, there are some patterns that may not representative of valid subtrees, due to the data mining tasks utilized not being performed by following the structure, but rather structural properties are preserved by the *DSM*. Hence, in the third experiment, the quality of the initially extracted subtrees set that contains both valid and invalid (disconnected) subtrees identified as *FullTree* is evaluated. Consequently, the *FullTree* rule type is compared with

Embedded and *Induced* subtree-based rules (the rule sets that exclude invalid/disconnected subtrees). Experimental results show that in the final rule sets for the majority of the cases, the AR and CR for the *FullTree* is always higher compared to those of embedded subtrees while only in some cases are both AR and CR the same. Moreover for CSLogs dataset for support 20% and 30%, there are no final rule sets to be evaluated. The higher AR and CR for the rule sets from *FullTree* indicate the advantages of using the *DSM* approach as the frequent patterns extracted are not limited to representing connected subtrees. Hence, potentially interesting associations could be found that would not be detected using traditional frequent-subtree-based approaches. These unique rules could in turn increase the prediction/classification capabilities and capture more instances in the dataset (i.e. increase coverage rate). When comparing embedded and induced subtrees, it was found that the AR and CR are either equal, or embedded subtrees are always better. Generally speaking, since the induced subtrees $\subseteq$ embedded subtrees, the number of rules in embedded is always higher or in some cases the same. One may observe that the additional rules in embedded subtrees may be important rules capable of classifying/predicting specific classes and capturing more instances in the datasets. They can capture associations deeply embedded over several levels within the tree structure and are not limited to only one level (parent-child) as is the case in induced subtrees.

In the fourth experiment setting, the AR and CR of the final frequent subtrees rule sets that were extracted using the *DSM* approach are compared with the rules extracted using the *XRules* classifier. The frequent-subtree-based rules discovered from *XRules* are not constrained by the position of node/nodes, and they all need to be representatives of valid connected subtrees. On the other hand, with the *DSM* approach *FDT* the node/s positions are taken into account and subtrees are further distinguished based on their occurrence within the *DSM*. Furthermore, since the structural information is ignored during rule generation, some rules could be representing invalid/disconnected subtrees. The results show that the AR for *FullTree* is higher compared to *XRules* but lower in CR, which is mainly due to its characteristic of distinguishing rules based on the exact occurrence of the node(s)/subtree(s) within the *DSM*. Hence, fewer instances can be covered by the rules but the specific instances can be more accurately classified. Moreover, for the

XRules approach, to discover association rules from XML data, the node/s has to be constrained so that they have to reflect the valid subtree. However, as demonstrated in this experiment, some association rules with disconnected subtrees contain important associations for classification/prediction purposes. To conclude, in forming association rules for tree-structured data, one should not be constrained to a valid and connected subtree because an interesting association can be anywhere in a tree instance, and it does not need to be a connected subtree of that instance.

In the final experiment, the framework is evaluated using a real-world data problem. This experiment is presented in the last section of this chapter to demonstrate the differences between two types of tree-structured data using the *DSM* approach. In the initial experiment, the dataset used is a weblog data. For this type of data, every attempt to access pages in the website is stored, thereby creating a large log file with many unique navigations over the website. The CRM data used in the final experiment represent the information of customer relationship management that have been stored in a tree-structured format. The result demonstrated that the *FullTree* rule set produced more rules in comparison with *Embedded* and *Induced* rule sets. Rules were then verified in order to determine their validity and interestingness. The results show that it is more advantageous to remove the rules that failed the statistical analysis and redundant assessment in the initial evaluation process and utilize the *confidence* constraint only at the end of the process. This will result in a relatively small number of rules and at the same time any detected redundant rules will be removed. Moreover, the rules from the *FullTree* approach offer a better final rule set in terms of classification and prediction accuracy compared to *Embedded* and *Induced*. The *FullTree* approach, since it contains both valid and invalid (disconnected) subtrees, results in generally more rules and the additional rules in *FullTree* may be important rules capable of correctly classifying/predicting the class of specific instances as well as capturing more instances in the datasets.

Increasing the *confidence* threshold from 50% to 60% reduces the number of rules to those that have very high accuracy because of large *confidence*. However, as the rule sets are reduced, more instances will not be captured by the rule set; hence, typically there is deterioration in the CR. Choosing smaller *confidence* thresholds will result in larger sets of rules that may lack generalization power, thereby weakening the AR

performance, but are capable of covering more instances. Alternatively, choosing relatively high *confidence* thresholds will result in a smaller set of rules, thereby achieving higher AR with the trade-off of capturing fewer instances. Thus, it is important to balance the trade-off between AR and CR in order to determine the optimal value for the minimum *confidence* threshold, and the choice may also be dependent on the nature and sensitivity of the application domain.

References

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc.

Agresti, A. (2007). *An introduction to categorical data analysis* (2nd ed.). Hoboken, N.J.: Wiley-Interscience.

Bayardo, R. J., Agrawal, R., & Gunopulos, D. (2000). Constraint-Based Rule Mining in Large, Dense Databases. *Data Mining and Knowledge Discovery, 4*, 217-240.

Chi, Y., Muntz, R. R., Nijssen, S., & Kok, J. N. (2005). Frequent Subtree Mining - An Overview. *Fundamenta Informaticae, 66*, 161-198.

Hadzic, F. (2011). A Structure Preserving Flat Data Format Representation for Tree-Structured Data. In *2011 Workshop on Quality Issues, Measures of Interestingness and Evaluation of Data Mining Models (QIMIE'11)*. Shenzhen, China: Frontiers of Computer Science in China Journal, Springer.

Hadzic, F., Dillon, T. S., Sidhu, A. S., Chang, E., & Tan, H. (2006). Mining Substructures in Protein Data. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*: IEEE Computer Society.

Hadzic, F., Hacker, M., & Tagarelli, A. (2011). XML Document clustering using structure-preseving flat representation of XML content and structure. In *7th International Conference on Advanced Data Mining and Applications*. Beijing, China.

Hadzic, F., & Hecker, M. (2011). Alternative Approach to Tree-Structured Web Log Representation and Mining. In *EEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. Lyon, France.

Hadzic, F., Tan, H., & Dillon, T. S. (2011). *Mining of Data with Complex Structures* (Vol. 333): Springer-Verlag Berlin Heidelberg.

Shaharanee, I. N. M., Hadzic, F., & Dillon, T. (2009). Interestingness of Association Rules Using Symmetrical Tau and Logistic Regression. In A. Nicholson & X. Li (Eds.), *AI 2009* (Vol. 5866, pp. 442-431): LNAI.

Shaharanee, I. N. M., Hadzic, F., & Dillon, T. (2010). A Statistical Interestingness Measures for XML Based Association Rules. In B.-T. Zhang & M. Orgun (Eds.), *PRICAI 2010: Trends in Artificial Intelligence* (Vol. 6230, pp. 194-205): Springer Berlin / Heidelberg.

Shaharanee, I. N. M., Hadzic, F., & Dillon, T. S. (2011). Interestingness measures for association rules based on statistical validity. *Knowledge-Based Systems, 24*, 386-392.

Sidhu, A. S., Dillon, T. S., & Chang, E. (2006). Protein Ontology. In Z. Ma & J. Y. Chen (Eds.), *Database Modeling in Biology: Practices and Challenges* (pp. 39-60). New York: Springer.

Sidhu, A. S., Dillon, T. S., Sidhu, B. S., & Setiawan, H. (2004). A Unified Representation of Protein Structure Databases. In M. S. Reddy & S. Khanna (Eds.), *Biotechnological Approaches for Sustainable Development* (pp. 396-408). India: Allied Publishers.

Tan, H., Dillon, T., Hadzic, F., Chang, E., & Feng, L. (2005). MB3-Miner: efficiently mining eMBedded subTREEs using Tree Model Guided candidate generation. In *Proceedings of the 1st International Workshop on Mining Complex Data 2005 (MCD 2005)*. Houston, Texas, USA: IEEE Computer Society Press.

Tan, H., Dillon, T., Hadzic, F., Chang, E., & Feng, L. (2006). IMB3-Miner: Mining Induced/Embedded Subtrees by Constraining the Level of Embedding. In *Advances in Knowledge Discovery and Data Mining* (pp. 450-461).

Tan, H., Hadzic, F., Dillon, T. S., & Chang, E. (2008). State of the art of data mining of tree structured information. *International Journal of Computer Systems Science and Engineering, 23*.

Tan, H., Hadzic, F., Dillon, T. S., Chang, E., & Feng, L. (2008). Tree model guided candidate generation for mining frequent subtrees from XML documents. *ACM Trans. Knowl. Discov. Data, 2*, 1-43.

Webb, G. I. (2007). Discovering Significant Patterns. *Machine Learning, Springer*, 1-33.

Zaki, M. J. (2005). Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications. *IEEE Transactions on Knowledge and Data Engineering, 17*, 1021-1035.

Zaki, M. J., & Aggarwal, C. C. (2003). XRules: an effective structural classifier for XML data. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Washington, D.C.: ACM.

Zhou, X. J., & Dillon, T. S. (1991). A statistical-heuristic feature selection criterion for decision tree induction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*, 834-841.

CHAPTER 8: RECAPITULATION OF THE THESIS AND FUTURE WORKS

8.1 Introduction

This chapter recapitulates the research undertaken in this thesis. It discusses the importance of the contributions to the research fields addressed, and draws out the main conclusion from the research. Even though the research undertaken is important and contributes to the body of knowledge of the investigated domain, a number of significant problems and applications are worth further exploration as part of future works involved in extending the developed framework. These outstanding problems are beyond the scope of the work in this thesis, but are nevertheless very important problems in the data mining field. Therefore, this chapter will also provide an overview of the planned future work in the areas of rules interestingness for relational data and semi-structured data.

In Section 8.2, the chapter provides a summary of each of the chapters of the thesis, and in Section 8.3, future works are discussed. The chapter concludes in Section 8.4 with a summary of the main contribution of the thesis.

8.2 Recapitulation

The section summarizes each chapter by drawing out the key points and main conclusions from presented materials. The contribution of this thesis is twofold, namely: evaluating association rules from relational data and semi-structured data.

The introductory chapter (**Chapter 1**) provides a background for the knowledge discovery and data mining fields in general, beginning with a discussion of knowledge discovery techniques and important issues that need to be considered. Their strengths, weaknesses and opportunities that lie ahead are fast-tracked, and consequently important issues of interest regarding the quality of rules from knowledge discovery processes is discussed. To clarify the knowledge discovery process, there is a detailed explanation of the several phases involved. In the thesis, the data mining application is viewed as one of the phases in the knowledge process which includes data pre-processing, pattern evaluation and knowledge interpretation/presentation. The chapter also differentiates between two

functionalities of data mining: predictive modeling and descriptive modeling. The tasks that are associated with both functions, including the classification, prediction, outlier analysis, clustering, association analysis and sequence discovery are described. Next, the discussion is centered on the types of data commonly encountered in the data mining process including relational data, sequential data, semi-structured data and finally, unstructured data.

The fundamental aspects of the knowledge discovery process are summarized in the earlier section of the chapter, while later sections describe the motivations and the research boundary that delineates the scope of this thesis. The overall structure of the thesis is provided in the last section of the chapter.

Chapter 2 provided the literature review of the current state of the research in the problem areas that are within the scope of the thesis as identified in Chapter 1. It begins with a basic concept of association rule mining. This serves to provide a fundamental understanding of the general issues surrounding the task of association rules mining. The essential issues in association rule mining are then extended by venturing into specific data types, namely the relational and semi-structured data. The development of algorithms for association rules mining and its applications in certain domains are listed and discussed. While many works have focused on the development of efficient algorithms for mining association rules specifically for relational data, recently more attention has been paid to understanding and developing efficient association rules from semi-structured data. Thus, several algorithms for association rule mining from both data types are provided and reviewed.

The chapter then examines the interestingness of rules, an important issue in the context of rules generation from both relational and semi-structured data. Three types of measures for interestingness of rules are reviewed, namely, the objective, subjective and semantic. It was found that there is no specific agreement on which is the best measure that can be applied to all cases; usually, measures are complementary and correlated with one another. Special attention is given to describing the objective measures as this approach offers a rigorous way of

discovering quality patterns based on a statistical mechanism underlying the measures.

The final section of Chapter 2 highlights the relationship between reducing the size of attributes in rules and generation of rules interestingness. Several feature subset selection techniques and their suitable applications are discussed. The feature subset selection is utilized in this thesis to reduce the size of frequent rules, thus reducing the complexity of the rules.

In **Chapter 3**, the key concepts are defined and the formal definitions of the terms necessary for understanding the particular problem areas addressed in this thesis are provided. This chapter then explains the concept and provides a definition of relational and semi-structured data. This section includes the details of data pre-processing which is in important task in the knowledge discovery process. This is due to the fact that in some cases, the dataset used may be contaminated with incomplete, inconsistent or irrelevant information or in some cases is not suitable for the data mining task at hand. Thus the pre-processing techniques are used to prepare data suitable for further data mining application. The approach for handling missing data and continuous data is described. As semi-structured data is utilized in this thesis, essential issues pertaining to the conversion of semi-structured data from XML documents to tree-structured format are explained. This includes the representative model of XML as a rooted ordered labeled tree as it is the important format that has been utilized in mining association rules from tree-structured data.

The chapter then defines the keys concepts of frequent itemset mining and frequent subtree mining, including maximal and closed frequent itemset mining, and embedded and induced subtree types for frequent subtree mining. The different characteristics of support and confidence of each frequent mining task are presented in detail.

Finally, at the end of Chapter 3, the problems to be addressed within this thesis are defined. It begins with the general problem of finding quality rules from an association rules mining framework. First, the problems of rules interestingness and validity for relational data are discussed. The characteristics of Apriori, Maximal and

Closed frequent itemsets are given in detail. Then the problem of rules interestingness and validity of rules derived from tree-structured data is explained. These types of data present two problems: first, the frequent subtrees are those patterns that have not been assigned a particular class to be used for a prediction/classification task; second, the frequent subtrees are those patterns that contain a certain preferred node that will be utilized as a class label that can be used for the classification/prediction tasks. A relationship between the feature subset selection application and the interestingness of rules is defined. A detailed discussion is provided on how the application of the feature subset selection task is utilized to determine a quality rule.

Finally, the chosen methodology for approaching the problem of evaluating the interestingess of association rules is given. The feature subset selection task is utilized in order to determine irrelevant attributes in predicting the class variable. The use of statistical analysis for determining significant and quality rules is formalized. This has been done by employing the hypothesis testing, correlation analysis and regression analysis approach. Additional assessment methods, namely redundancy removal and contradictive removal, were then explained. A combination of these measures will offer a quality rule set that not only satisfies the statistical analysis, but also contains non-redundant and non-contradictive rules.

For frequent patterns from both relational data and tree-structured data, the aim is to investigate the use of statistical measures for the above problem, to determine how the existing interestingness measures and parameters can be utilized effectively, and the sequence of their use. With the aforementioned measures used for measuring the interestingness of rules from both frequent itemsets and frequent subtrees, another issue arises: whether the reduced size of rules resulting from these measures significantly reduces their usefulness for classification and prediction purposes. Hence, the rules accuracy rate (AR) and coverage rate (CR) and generalization capabilities are defined and measured.

A high level perspective of the proposed solution is provided in **Chapter 4**. An overview is provided of the way in which the problems that this thesis focuses on (defined in Chapter 3) are addressed, and in particular on how the rules and their

interestingness are measured. The proposed framework consists of two sections: the framework for evaluating the interestingness of association rules from relational data (based on frequent itemset mining) and the framework for evaluating the interestingness of association rules from tree-structured data (based on frequent subtree mining). Different statistical analysis, redundancy and contradictive assessment methods will be utilized, including correlation and regression analysis, and the redundancy and contradictive removal. Moreover, there is a review of the application of Symmetrical Tau feature selection in providing the relative usefulness of attributes to predict the value of class attributes. This chapter then describes the final step of each framework, that is, the way that the quality of frequent itemsets and frequent subtrees are measured based on their accuracy, coverage rate and generalization capabilities.

The following three chapters are concerned with the details of the development of the proposed framework for evaluating the interestingness of frequent patterns from both relational and tree-structured data, and the evaluation of the framework in those different settings. This framework provides a means of utilizing the interestingness and constraint-based parameters and the appropriate sequencing of their usage.

Chapter 5 describes in detail the development of the proposed framework for evaluating the association rules discovered from relational data using the Apriori, Maximal and Closed approaches. The aim of the developed framework is to investigate how data mining and statistical measurement techniques can be utilized to complement each other, and to develop a proper sequence of use of these techniques to arrive at a more reliable and interesting set of rules. There are several relevant tasks involved in the development of the proposed framework. These include the data pre-processing application such as the preparation of suitable data format and data cleaning for the association rule mining process. These pre-processing applications ensure that only appropriate and accurate data are supplied to the association rule mining process. Within the framework, a well-recognized feature subset selection approach is utilized. The advantage of selecting certain features for inclusion in the association rule mining process is that only those attributes relevant to the classification task at hand are used to generate the association rules. The frequent itemsets are then discovered using the aforementioned association rules mining

algorithms based on the selected attributes from the feature subset selection process. A formal approach on how the rule set verified is provided, including a detailed explanation of how statistical analysis, and redundancy and contradictive assessment methods are exploited in the framework. In regards to these measures, the chi-squared test is used to test the dependence between antecedent and the consequent of an association rule, while the logistic regression models are developed to measure the relationship between the target variable and the input variable. In this analysis, the target variable is recognized as the consequent and the input variable as the antecedent of an association rule. Moreover, any rules identified to be redundant or contradictive to other rules in the rule set are discarded. Within this chapter, the process of evaluating the rules based on the aforementioned statistical analysis, and redundancy and contradictive assessment methods are formalized. At the end of this chapter, the process of evaluating the rule reduction based on the statistical analysis, redundancy and contradictive assessment methods, as well as the confidence measure based filtering, is described with respect to measuring the accuracy and the coverage rate of the rule set.

Experiments undertaken to evaluate the proposed framework for rules interestingness from relational data are described in **Chapter 6**. The proposed framework was evaluated based on three association rule mining algorithms, namely the Apriori, Maximal and Closed algorithms. The quality of the rules discovered is measured using statistical analysis, and redundancy and contradictive assessment methods. Two variants of the Apriori algorithm were utilized. The first variant corresponded to the standard Apriori algorithm with both *support* and *confidence* threshold, while the second variant was constrained using only the minimum *support* threshold. The rationale for using two variants of the Apriori algorithm is to demonstrate the effect of the *confidence* application before and after the statistical analysis, and redundancy and contradictive assessment methods have been applied to filter the rule set in the framework. Three important findings are presented in this chapter.

The first finding from the first experiment reveals the effect of applying minimum *confidence* thresholds at the end of the rule evaluation process after discarding all the rules that failed the statistical test, the redundant rules, and the contradictive rules. The advantage of doing this is that any detected contradictive rules will be removed.

The disadvantage of applying the minimum confidence threshold at the start of the process is that the existence of a contradictive rule that has relatively low *confidence* will not be known. This lack of knowledge can cause an unreliable association rule to become part of the final rule set, which as demonstrated in Chapter 6, reduces the accuracy of the rule set in comparison to when the rule was removed.

In comparing the interestingness of rule sets from Apriori with *min_sup* (later referred to as Apriori), Closed and Maximal, the result varies and in some cases will depend on the sensitivity of the domain at hand. Generally, based on the dataset used in this thesis, the classification and prediction accuracies are better for the final rules set discovered from Closed approach for Wine, Iris and Adult datasets, while for the Mushroom dataset, the Apriori approach obtains the best results.

The third finding from the final experiment concerns the minimum *confidence* effect on the proposed framework of all three algorithms. It was found that initially setting a low *confidence* measure may be a desired approach to improve the rule coverage rate, as opposed to initially setting a high *confidence* measure which may discard some of the patterns/rules that cover a smaller subset of data objects from the domain at hand. The rules covering a smaller subset of data may be necessary to detect contradictions in the formed associations and discard those contradictive rules. This is experimentally demonstrated in this thesis where, for the majority of datasets, a better accuracy is achieved when the confidence constraint is applied after removing any contradictive rules. In addition, the changes in confidence values have a direct impact on the size of rule set, the accuracy rate (AR) and the coverage rate (CR). The trade-off between finding a rule set with optimal values of AR and CR is essential (Novak, Lavrač, & Webb, 2009; Wang & Dillon, 2006).

Chapter 7 provides an extension of the work from evaluating the interestingness of association rules from the relational data to semi-structured data. This chapter demonstrates the extensibility of the developed framework for evaluating the frequent subtree rules. It also indicates the direction taken so that a frequent subtree can be evaluated using the proposed framework. In the initial experiment, the traditional frequent subtree mining algorithm is utilized to discover frequent subtrees, and subsequently, the interestingness of these frequent subtrees was measured using

the proposed framework. The results indicate that the combination of statistical analysis, redundancy and contradictive assessment methods is a means of discarding non-significant rules, which significantly reduces the overall complexity of the rule set. However, this requires a trade-off between higher accuracy at the cost of lower coverage rate. As more complex tree-structured data was explored by the aforementioned traditional frequent subtree mining algorithm, the rule evaluation task using the statistical analysis, redundancy and contradictive assessment methods based on the proposed framework became infeasible due to the large volume of elements present in all transactions/instances. A new means of preserving tree-structured data in a flat table format, namely the *Database Structure Model* (*DSM*) approach, was presented in (Hadzic, 2011) and was utilized in the later experiments to evaluate the interestingness of frequent subtrees.

In optimizing the number of rules based on sequence of usage of parameters, the result demonstrated that, in most of the cases for the dataset used in this chapter, the reduction of rules has increased the prediction/classification ability of the rules but at the same time weakened the capabilities of the rules to capture more instances in the datasets. However, in several cases, as the number of rules decreased, both AR and CR declined and in two cases even with the reduced number of rules, both AR and CR are well preserved. For the problem of evaluating the frequent subtree extracted from a dataset with backtrack attributes and without backtrack attributes, the result confirmed that the CR for the final rule sets for rules with backtrack is always higher compared to the CR for the final rules without backtrack. In contrast, the AR for the final rule set for rules without backtrack is higher compared with the rule set with backtrack. With the inclusion of the backtrack node, the rules can capture more instances in the dataset. However, this comes with a trade-off since the rules with backtrack attributes are not specific enough, thereby reducing the AR. In accessing the quality of the initially extracted subtree set that contains both valid and invalid (disconnected subtrees) identified as *FullTree*, a comparison is made with rule sets based on *embedded* and *induced* subtree (the rule set that exclude invalid/disconnected subtrees). The result shows that in the final rule sets for the majority of the cases, the AR and CR for the *FullTree* is always higher compared to embedded subtrees while only in several cases both AR and CR are the same. The higher AR and CR for the rule sets from *FullTree* indicates the advantages of using

the *DSM* approach as the frequent patterns extracted are not limited to the representation of connected subtrees. As demonstrated in the experiment, these additional rules can increase the prediction/classification capabilities and capture more instances in the datasets. When comparing embedded with induced subtrees, the experiment demonstrated that the AR and CR are either equal, or embedded subtrees achieve better values. One may observe that the additional rules in embedded subtrees may be important rules that are capable of classifying/predicting specific classes and capturing more instances in the datasets, as they can capture interesting associations embedded deeply in the tree instances (ancestor-descendant) as opposed to induced subtrees that preserve only parent-child relationships.

Another important finding emerged when comparing the AR and CR of the final frequent subtrees rule sets that been preserved based on the *DSM* approach with the *XRules* classifier classifier (Zaki & Aggarwal, 2003). The results show that the AR for *FullTree* is higher compared to *XRules* but lower in CR. This is due to its characteristics of distinguishing rules based on the exact occurrence of specific node(s)/subtree(s). Consequently, fewer instances can be covered by the rules but very specific instances can be classified and/or predicted. Moreover, for the *XRules* approach, to discover association rules on XML data, the node/s has to be constrained so that they have to reflect the valid subtree. However as demonstrated in this experiment, some association rules with disconnected subtree contain important pattern for classification/prediction purposes. To conclude, an interesting subtrees can exist in any place in a tree instances, thus one should not constrain themselves to valid and connected subtree when forming association rules from tree-structured data.

8.3 Future Works

This section is organized according to the problem areas that can be addressed by future extensions or research. Hence, for each contribution made by this thesis, some possible future extensions are indicated.

8.3.1 Refining the Statistical Analysis based Rule Filtering

The framework proposed in this thesis has demonstrated how data mining, statistical measurement techniques, redundancy and contradictive assessment methods can be utilized in a proper sequence to complement each other and to arrive at a more reliable and interesting set of rules. The chi-square and the logistic regression measures were used as a case in point for statistic-based rule filtering, while symmetrical tau was utilized in the feature subset selection process. However, by no means is any claim being made that these are the most optimal measures to be used. Hence, other statistic-based measures and/or techniques for rule removal/attribute relevance determination can be considered for use in the proposed framework. As part of the future work, it will therefore be interesting to explore the use of other statistical techniques and evaluate their advantages/disadvantages with respect to chi-square, logistic regression as well as any other techniques progressively being utilized. For example, several of the techniques mentioned in (Han, Kamber, & Pei, 2012) have proven to be effective when applied to data in the domains of science, economics and social sciences. The use of such techniques within the proposed framework will be explored in terms of the most effective techniques to satisfy a particular purpose (e.g. irrelevant attribute detection) as well as an optimal combination of techniques so that the benefits of the different measures are exploited so as to optimize the quality of the derived rule set.

8.3.2 Evaluation of Rules Interestingness for Maximal/Closed Frequent Subtree

As demonstrated in Chapters 4, 5 and 6, the developed framework is capable of evaluating Maximal and Closed frequent itemsets. Maximal and Closed algorithms for mining frequent maximal/closed itemsets are known for their ability to reduce the number of frequent itemset candidates that need to be enumerated. Even with a reduced set of frequent itemsets, no information is lost since the complete set of

frequent items can be obtained from both closed and maximal sets of frequent items. Thus, given these benefits, this work has been extended to frequent subtree mining to find frequent maximal/closed frequent subtrees. While these algorithms are capable of mining both maximal/closed subtrees from databases, a similar observation can be made that some of the subtrees discovered are due to pure coincidence resulting from certain randomness in the particular dataset being analyzed. Therefore, rules can be either a true discovery or merely an artifact of random association.

In terms of the proposed framework evaluating the interestingness of frequent subtrees generated based on *DSM* approach, the representation of frequent subtrees as a flat table format might offer a way for a prominent maximal and closed itemset mining to be utilized. As described in Chapters 4 and 5, the *GenMax* (Gouda & Zaki, 2001) and *CHARM* (Zaki & Hsiao, 2002) are capable of discovering maximal and closed frequent patterns based on relational data format. Hence, it is feasible to extend both approaches in discovering maximal and frequents pattern from a flat table format (*FDT*). Similar to the experiment conducted in Chapter 7, it is worth assessing the quality of the rules and issues such as the implication of the existence/non-existence of backtrack node discovered from maximal and closed based frequent subtrees.

Thus, the task of evaluating frequent maximal/closed subtrees is highly important and these problems are yet to be resolved in the tree-structured data mining research field. Given the results of evaluating maximal/closed itemsets, extensions to the framework presented in this thesis will be made in order to evaluate the interestingness of rules based on frequent maximal/closed subtrees.

8.3.3 Extended Statistical Analysis for Tree-Structured Data

In Chapter 7, the capability of the software used for building the logistic regression model is limited. For example, there is a limit on the number of unique values each attribute can have. Hence, an additional assumption that was included in this work is that, any input attributes exceeding a certain limit of its possible values will be omitted. The removal of these attributes might be useful in reducing the number of attributes for logistic regression models; however, they might also contain important

information for the classification task, and thus their removal may result in deterioration of the model. Another possible direction is to apply an additional pre-processing task such as the *cardinality reduction* (Refaat, 2007) to reduce the number of categories of attributes. This task, as described by (Refaat, 2007), reduces the number of categories and can be done either by ignoring same categories with small frequencies or by creating a schema to group categories into a smaller set of new super categories. This can be done by investigating the content of each category and grouping them on the basis of their meaning with respect to the specific domain. Moreover, as asserted by (Refaat, 2007), if there is no agreement on format when combining the categories, one can opt to combine the categories to increase the overall contribution of the variables to the model being developed.

Moreover, rather than using the proposed statistical analysis, it is worth considering several other statistical techniques for data analysis. Several of these techniques as mentioned in (Han et al., 2012) have been proven effective and been extensively applied to data in the domains of science, economics and social sciences. This will be useful because, as demonstrated in the framework for evaluating frequent subtrees, most of the tree datasets consist of large volumes of transactions and attributes, where attributes can potentially have a large domain of possible values. The complexities of the attributes require us to use statistical measures that are better suited for tree-structured types of data and those observed characteristics. This provides an interesting research problem that merits future investigation.

8.3.4 Evaluation of Interestingness of Rules based on Unordered Subtrees

This thesis provides a framework that evaluates only the ordered frequent subtrees; hence, its limitation. However, as stated in Chapter 3, a tree-structured pattern/subtree may be unordered and the order of the sibling nodes does not need to be preserved, or in other words, the order of sibling nodes can be swapped and the resulting subtree is considered the same. Unordered subtrees are important in many applications where the order of the sibling-nodes is considered unimportant or irrelevant to the domain, or is simply unavailable (Hadzic, Tan, & Dillon, 2011). This is especially the case when the data comes from heterogeneous sources where there may be no standard way of representing the available domain information, and

hence the same aspects of the domain may be represented at different sibling nodes. (Chi, Muntz, Nijssen, & Kok, 2005; Tan, Hadzic, Dillon, & Chang, 2008) have overviewed several prominent algorithms for mining frequent unordered subtrees in terms of their performances. While these algorithms offer a way of discovering frequent unordered subtrees, some work needs to be done on evaluating the interestingness of the unordered subtrees and their usefulness in the classification/prediction task. Extending the current framework to evaluate these types of rules appears to be feasible; however, additional work and understanding is required in order to systematically arrive at desirable results. Since in this thesis, the tree-structured data has been preserved in a flat table format, an appropriate flat table format needs to be developed for the mining of tree data in unordered contexts. Since the order of sibling nodes is considered unimportant, one might preserve a certain structure that relatively will give the user the most information. Moreover, one needs to determine an appropriate representation of rules, and an updated evaluation phase to ignore the order when determining rules accuracy and coverage rate.

8.3.5 Evaluation of Rules Interestingness from Sequential Data

Mining frequent subsequences is another major development in the association rule mining field that specifically deals with sequential data, as explained in Chapter 1. The application of the proposed framework for evaluating the interestingness of sequential patterns is worthy of exploration. A sequence as described in (Han et al., 2012) refers to an ordered list of events. There are three main characteristics of sequence, namely the *time-series* data, *symbolic sequence* data and *biological sequences*. The sequential data may involve a certain timestamp, occurrences, streams and gaps. The sequence mining task is a challenging one due to the large search space. As pointed out by (Zaki, 2001), this task requires the algorithm to make a full database scan for the longest frequent sequences. To address this problem, (Zhao & Bhowmick, 2003) applied a minimum support threshold that can prune those sequential patterns out of user interest, consequently making the mining process less complex. There are many sequence mining algorithms as reviewed by (Hadzic et al., 2011; Zhao & Bhowmick, 2003) capable of performing well in generating frequent subsequence patterns. However, motivated by the problems defined in this thesis, one still needs to evaluate the quality of the subsequences

produced. In doing this, several aspects need to be taken into consideration. One important consideration when mining sequential data is the need to preserve the order of objects in the extracted knowledge patterns or rules. Hence, the proposed framework needs to be modified and readapted for this purpose. Moreover, certain tools of statistical analysis for time series forecasting can be integrated with the current statistical tools available in the proposed framework. As most of the time series data for statistical analysis as provided by (Hyndman, 2011) is in relational format, thus preserving the subsequence into flat table format, this might offer a way for these statistical based time series analysis tools to be utilized to evaluate the subsequences rules.

8.3.6 Evaluation of Rules Interestingness from Unstructured Data

Currently, many enterprises implement certain business processes to streamline their operations, and the majority of these processes are automated, thereby generating numerous documents. This results in the proliferation of unstructured information within the enterprise such as email communication, power point presentation, word-processing, notes and customer reviews/comments. Therefore, there is a need to extract more structured information from this unstructured data for business intelligence purposes. (Han et al., 2012) provided several examples of data mining applications for unstructured data such as mining text data from documents, user comments of product, multimedia contents and the web.

The main characteristic of unstructured data is that there is no description of the underlying structure of the data, consequently making the process of frequent rules discovery from unstructured data more complicated compared to mining relational or semi-structured data. Some work has been done on mining rules from unstructured data such as the mining of text documents by (Feldman & Dagan, 1995; Mahgoub, Rösner, Ismail, & Torkey, 2007; Nahm & Mooney, 2002).

In developing the current framework so that it can be applied to unstructured data, one needs to determine the statistical techniques that are appropriate for such data. While with the proposed framework, the data is partitioned into training and testing sets, this might be a challenging task when dealing with unstructured data as there

are no pre-determined structures available, thereby making it difficult to determine the appropriate attributes/concepts and instances to be partitioned.

Given the premise that unstructured documents can be converted into XML format (Mahgoub et al., 2007), this offers a direction for the current proposed framework to be readapted and extended into evaluating unstructured patterns. This can be done by identifying important keywords that satisfy a pre-determined threshold such as support and confidence, or by utilizing a domain expert to analyze the text, thus providing some insight in determining a suitable structure for the documents. By doing this, certain information can be mapped/tagged into pre-determined concepts. Hence, the conversion of unstructured features into a structured format using the XML format offers a way for the proposed framework to be utilized for the evaluation of the unstructured patterns. However, one need to be careful in defining the concepts as these may force the unstructured information into a certain structure and in some cases may result in the partial loss of information originally residing in the unstructured data. As the unstructured data will be transformed into XML format, ideally a step similar to that given in Framework *B* can be utilized to evaluate the rules. While this offers an opportunity to examine the capabilities of the proposed framework with more complex data, this will require major development to adapt the framework and this work is reserved for future research.

8.3.7 Incorporating Domain Knowledge within the Proposed Framework

Domain-driven data mining (D^3M)(Cao, 2010) is a new frontier in data mining research which is currently under extensive investigation. The D^3M clearly is an extended phase of data mining and points to the future direction of data mining research. The development of D^3M is motivated by the gap created between the traditional data mining research and executable capabilities of data mining techniques in real-world applications. Presenting already known patterns is of no particular interest; nor are those patterns that are hard to apply or used for decision support. Given such issues, the development of D^3M is of crucial importance.

The proposed framework provides a means of using statistical analysis, redundancy and contradictive assessment methods and their sequence of usage to produce a set of

quality rules from either relational or semi-structured data. Considering the current trend in data mining research in producing "domain-driven and actionable knowledge discovery" (Cao, 2010), the next important task is to enhance the capabilities of the current framework to produce "quality actionable rules". This will need to be done by incorporating domain knowledge into the whole framework in the form of a machine-readable knowledge base. Generally, the knowledge database can be developed by a domain expert, and the framework will be made general enough to accept input from any domain a knowledge base and a dataset from which the association rules are to be discovered. This knowledge base will serve the purpose of verifying whether any of the discovered rules already exist in the domain knowledge, which will prevent the inclusion of "common sense"/uninteresting rules. Similarly, if certain discovered rules represent a contradiction or refinement of the current body of domain knowledge, the interestingness of these will be high as they could potentially extend the current body of knowledge or correct previously drawn conclusions that are proven wrong as new data/evidence emerges. Furthermore, in certain application areas, the aim is to find rare or exceptional events, known as outlier detection, and the knowledge base representing the domain knowledge could indicate what the "norm" is considered to be in the particular domain under investigation. Utilizing this domain knowledge will make the outlier detection phase more accurate and will make it easier to distinguish those outliers that represent true exceptions to the domain knowledge from simply noise in data. Hence, this is a rather important extension of the developed framework which will make its utilization in different domains/application tasks more attractive.

8.4 Conclusion

This chapter has summarized the thesis, highlighting the key points and the main conclusions from each chapter. This chapter has also identified the research problems that will be addressed in future, as these were outside of the scope of this thesis. These include the exploration of suitable statistical techniques to measure the interestingness of rules from sparse and complex data. Furthermore, in order to extend the application of the current framework to evaluate other types of subtrees and data formats, the framework may require additional modification to suit certain data formats and applications. To conclude, the research undertaken has proven to be

important as the developed framework is capable of achieving the desired result of filtering our irrelevant/uninteresting rules without negatively affecting accuracy in both relational and tree-structured data domains. Given the quality of results obtained for relational and tree-structured data, a number of significant problems and applications are worth further research exploration to adapt and extend the proposed framework to be applicable to a wider range of data formats and application tasks.

References

Cao, L. (2010). Domain-Driven Data Mining: Challenges and Prospects. *Knowledge and Data Engineering, IEEE Transactions on, 22*, 755-769.

Chi, Y., Muntz, R. R., Nijssen, S., & Kok, J. N. (2005). Frequent Subtree Mining - An Overview. *Fundamenta Informaticae, 66*, 161-198.

Feldman, R., & Dagan, I. (1995). Knowledge Discovery in Textual Databases (KDT). In *Knowledge Discovery and Data Mining*. Portland, USA.

Gouda, K., & Zaki, M. J. (2001). Efficiently Mining Maximal Frequent Itemsets. In *First IEEE International Conference on Data Mining (ICDM'01)* (Vol. 0, pp. 163).

Hadzic, F. (2011). A Structure Preserving Flat Data Format Representation for Tree-Structured Data. In *2011 Workshop on Quality Issues, Measures of Interestingness and Evaluation of Data Mining Models (QIMIE'11)*. Shenzhen, China: Frontiers of Computer Science in China Journal, Springer.

Hadzic, F., Tan, H., & Dillon, T. S. (2011). *Mining of Data with Complex Structures* (Vol. 333): Springer-Verlag Berlin Heidelberg.

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: concepts and techniques* (3rd ed.). Waltham, Mass.: Elsevier/Morgan Kaufmann.

Hyndman, R. J. (2011). Time Series Data Library. http://robjhyndman.com/TSDL. Accessed on 25 December 2011.

Mahgoub, H., Rösner, D., Ismail, N., & Torkey, F. (2007). A Text Mining Technique Using Association Rules Extraction. *INTERNATIONAL JOURNAL OF COMPUTATIONAL INTELLIGENCE, 4*, 21-28.

Nahm, U. Y., & Mooney, R. J. (2002). Text mining with information extraction. In *The Spring Symposium on Mining Answers from Texts and Knowledge Bases* (pp. 60-67). Stanford/CA.

Novak, P. K., Lavrač, N., & Webb, G. I. (2009). Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Patterns and Subgroup Mining. *J. Mach. Learn. Res., 10*, 377-403.

Refaat, M. (2007). *Data preparation for data mining using SAS*. San Francisco: Morgan Kaufmann Publishers.

Tan, H., Hadzic, F., Dillon, T. S., & Chang, E. (2008). State of the art of data mining of tree structured information. *International Journal of Computer Systems Science and Engineering, 23*.

Wang, D., & Dillon, T. S. (2006). Extraction of classification rules characterized by ellipsoidal regions using soft-computing techniques. *International Journal of Systems Science, 37*, 969 - 980.

Zaki, M. J. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning, 42*, 31-60.

Zaki, M. J., & Hsiao, C.-J. (2002). CHARM: An Efficient Algorithm for Closed Itemset Mining. In *2nd SIAM International Conference in Data Mining*.

Zaki, M. J., & Aggarwal, C. C. (2003). XRules: an effective structural classifier for XML data. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Washington, D.C.: ACM.
Zhao, Q., & Bhowmick, S. S. (2003). Sequential Pattern Mining: A Survey. In: Nanyang Technological University, Singapore.