STATUS OF THESIS

Title of thesis: A New Scheme for Extracting Association Rules

I AIMAN MOYAID SAID

Hereby allow my thesis to be placed at the information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. The thesis is classified as:

   □ Confidential

   ☑ Non-confidential

If this thesis is confidential, please state the reason:

____

The contents of the thesis will remain for _____ years.

Remarks on Disclosure:

____

Endorsed by

Aiman Moyaid Said
Universiti Teknologi PETRONAS
Malaysia.

Date: 3-3-2010

Dr. Dhanapal Durai Dominic
Universiti Teknologi PETRONAS
Malaysia.

Date: 2/3/10

UNIVERSITI TEKNOLOGI PETRONAS

Approval by Supervisor (s)

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance, a thesis titled "**A New Scheme for Extracting Association Rules**" submitted by (**Aiman Moyaid Said**) for the fulfillment of the requirements for the degree of Master of Science in Information Technology.

____

Date

Signature :

Dr. P.D.D. Dominic
Senior Lecturer
Department of Computer & Information Sciences
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750 Tronoh,
Perak Darul Ridzuan, MALAYSIA.

Main Supervisor : Dr. Dhanapal Durai Dominic

Date : 3/3/10

Signature :

Co-Supervisor : Dr. Azween Abdullah

Dr. Azween Bin Abdullah
Associate Professor
Department of Computer and Information Sciences
Universiti Teknologi Petronas
Tel: 05-3687507 Fax: 05-3656180

Date : 3/3/10

TITLE PAGE

UNIVERSITI TEKNOLOGI PETRONAS

A New Scheme for Extracting Association Rules

By

Aiman Moyaid Said

A THESIS
SUBMITTED TO THE POSTGRADUATE STUDIES PROGRAMME
AS A REQUIREMENT FOR THE
DEGREE OF MASTER OF SCIENCE
INFORMATION TECHNOLOGY
BANDAR SERI ISKANDAR PERAK
OCTOBER, 2009.

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledge. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Signature :

Name : Aiman Moyaid Said

Date : 3-3-2010

ABSTRACT

Data mining is the process of exploring and analyzing large databases to extract interesting and previously unknown patterns and rules. In the age of information technology, the amount of accumulated data is tremendous. Extracting the association rule from this data is one of the important tasks in data mining.

In Data mining, association rule mining is a descriptive technique which can be defined as discovering meaningful patterns (itemsets tend to take place together in the transactions) from large collections of data. Mining frequent patterns is a fundamental part of association rules mining.

Most of the previous studies adopt an a priori-like candidate set generation-and-test approach to generate the association rules from the transactional database. The priori-like candidate approach can suffer from two nontrivial costs: it needs to generate a huge number of candidate sets, and it may need to repeatedly scan the database and check a large set of candidates by pattern matching.

In this thesis, the purpose of this study was to explore association rules and present a new scheme for mining the association rules from transactional database that can guarantee better performance than the priori-like schemes. The proposed scheme is using the integration of both the pattern growth approach and apriori rule generation approach.

Another aim was to apply the new scheme using real market basket dataset case study to assess its effectiveness. The knowledge obtained from the analysis of the dataset using the proposed scheme can be used to improve the efficiency of a promotional campaign and a store layout.

Several performance experiments were carried out on the collected data set and existing data set and the result of the study was that the new scheme outperformed the apriori-like schemes for both the dense data and the sparse data.

ABSTRAK

Perlombongan data adalah proses mengeksplorasi dan menganalisa pangkalan data yang besar untuk mengekstrak pola- pola dan peraturan- peraturan yang menarik serta yang sebelum ini tidak diketahui. Di era teknologi maklumat, jumlah data yang terkumpul sangat banyak. Mengekstrak peraturan kombinasi daripada data ini adalah salah satu tugas penting dalam perlombongan data.

Dalam perlombongan data, peraturan kombinasi perlombongan adalah teknik gambaran yang boleh ditakrifkan sebagai mengenali pola- pola yang bermakna (set item – set item berlaku bersama- sama dalam transaksi- transaksi) dari kumpulan data yang besar. Perlombongan pola secara berkali-kali adalah asas kepada peraturan- peraturan kombinasi perlombongan.

Kebanyakan daripada kajian yang terdahulu mengadaptasi teknik seperti-Apriori di mana calon-calon dibina dan diuji untuk menghasilkan peraturan kombinasi daripada pengkalan data transaksi. Cara seperti-Apriori ini mempunyai dua kekurangan; di mana ianya perlu menghasilkan sejumlah besar set-set calon, dan mungkin perlu membaca pangkalan data berulang kali dan mengenalpasti set calon yang sangat banyak menggunakan pola penyesuaian.

Dalam tesis ini, tujuan kajian ini adalah untuk mengeksplorasi peraturan- peraturan kombinasi dan mempersembahkan satu skema baru untuk perlombongan peraturan-peraturan kombinasi daripada pangkalan data transaksi yang boleh menjamin pencapaian yang lebih baik berbanding skema - skema seperti-Apriori. Skema yang telah dicadangkan menggunakan gabungan daripada kedua-dua pola pembesaran dan peraturan pembinaan apriori.

Kajian ini juga bertujuan untuk mengaplikasikan skema baru menggunakan set data pasaran sebenar untuk menguji keberkesanannya melalui kajian kes. Maklumat yang didapati daripada analisa set data adalah skema yang dicadangkan boleh digunakan untuk meningkatkan pencapaian promosi kempen dan susun atur kedai.

Beberapa eksperimen pencapaian telah dijalankan pada set data yang telah dikumpulkan dan set data yang sedia ada, dan keputusan daripada kajian ini adalah pencapaian skema yang baru mengatasi pencapaian skema- skema seperti-apriori dari segi kepadatan data dan keluasan data.

ACKNOWLEDGEMENTS

Thanks to Allah, The Almighty God, for giving me the power and courage to endlessly pursue what is so called knowledge.

I would like to express deep gratitude to my supervisors Dr. P. D. D. Dominic and Dr. Azween Abdullah for their continuous guidance, support, reviews and helpful suggestions. I also would like to offer special thanks to Dr Ahmed M. Zeki for his ample guidance and support during my research.

I would like to thank my parents, my sister, and my brothers for their support and love which gave me the inspiration and persistence to overcome all the difficulties I had over all the times of working on my thesis.

Special thanks also go to my friends; their friendship and help are priceless.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE: INTRODUCTION

In the age of information technology, the size of the databases created by the organizations is tremendous, due to the availability of low-cost storage and the evolution in the data capturing technologies. These organizations sectors include retail, petroleum, telecommunications, utilities, manufacturing, transportation, credit cards, insurance, banking and many others. Over years, a lot of data is stored in large databases. Representative examples of such huge database, Yahoo has more than five hundred million unique visitors per month who produce more than ten Terabytes of click-stream data each day, resulting in Petabyte scale data warehouse. Whether these databases are from business enterprise or scientific experiment, there is a need to explore the massively volume of data in order to extract valuable information. Knowledge discovery in databases (KDD) represents the exploration processes which identifying precious information in such huge databases (Fayyad et al. 1996). This valuable information can help the decision maker to make accurate future decisions. In addition, KDD is the process of searching the database for finding potentially useful patterns and models which describe the structure of the data by using techniques from statistics, pattern recognition and machine learning. KDD applications deliver measurable benefits, including reduced cost of doing business, enhanced profitability, and improved quality of service.

1. 1 Knowledge Discovery in Databases (KDD)

In general, the process of KDD consists of a sequence of the following steps (Fayyad et al. 1996, Zhang et al. 2002)

- Defining the problem
- Data preprocessing
- Data mining
- Data post processing

1. 1. 1 Defining the Problem

In this step the goal of the knowledge discovery must be identified. The unifying aim of the KDD process is to create knowledge from data in the large databases. The goals should be verified as actionable. For example, once the goals are achieved, the knowledge discovered can be applied to industry or research area. The data to be used should also be identified.

1. 1. 2 Data Preprocessing

Data preprocessing is the process of converting the data into a suitable format for subsequent analysis, a large amount of time is needed for the data preprocessing. It comprises data collecting, data cleaning, data selection, and data transformation. Data collecting is to collect necessary data from various internal and external sources, resolve representation and encoding differences, and put data together. Data cleaning means checking and resolving data conflicts, unusual or exceptional values, missing data, noisy data, and vagueness. Conversions and combinations might be required to produce new data fields such as ratios or rolled-up summaries. The task of data selection is to select related data to an analysis task from a given database. After related data has been chosen, data transformation process transforms or merges data into forms suitable for mining by performing aggregation or summary operations.

1. 1. 3 Data Mining (DM)

Data mining is an essential part of KDD. In the data mining step the preprocessed data is analyzed by a specific data-mining algorithm. Such algorithms contain looking for patterns of interest in a particular representational form or set of such representation, classification rules or trees, clustering, regression, sequence modeling, dependency, and so forth. A more detailed overview of data mining is given in Section 1. 2.

1. 1. 4 Data Post Processing

Data post processing contains pattern evaluation, deploying the model, and all the operations that are carried out to make the data mining output easy to understand. Pattern evaluation identifies the interesting patterns representing knowledge based on some interesting measures. Based on the identified patterns, prediction model is built and tested for accuracy on an independent dataset one that has not been used to build the model. After the predictive model has been tested, it is deployed to predict results for new cases. This process might require building computerized systems that capture the suitable data and generate a prediction in real time so that a decision maker can apply the prediction. Whatsoever models are being deployed, maintenance is required to adjust to changes in economy, customer behavior, and so on. This step needs continuous revalidation of the model, with new data to evaluate if it is still suitable. The final process in KDD is knowledge demonstration. It demonstrates mined knowledge to users.

1. 2 Data Mining

This is the important part of KDD. Data mining (DM) is a step in the knowledge discovery process, consisting of particular algorithms (methods), is the task of drilling through the huge volumes of data to discover useful knowledge. The knowledge discovery process is iterative process; Figure 1-1 illustrates a classic KDD process.

1. 2. 1 Data Mining Task

Decide the type of data mining tasks; we have to confirm that the functions and tasks to be achieved by new system belong to which kind of data mining task. Data mining generally involves four classes of task; Classification, Clustering, regression, and Association rule learning.

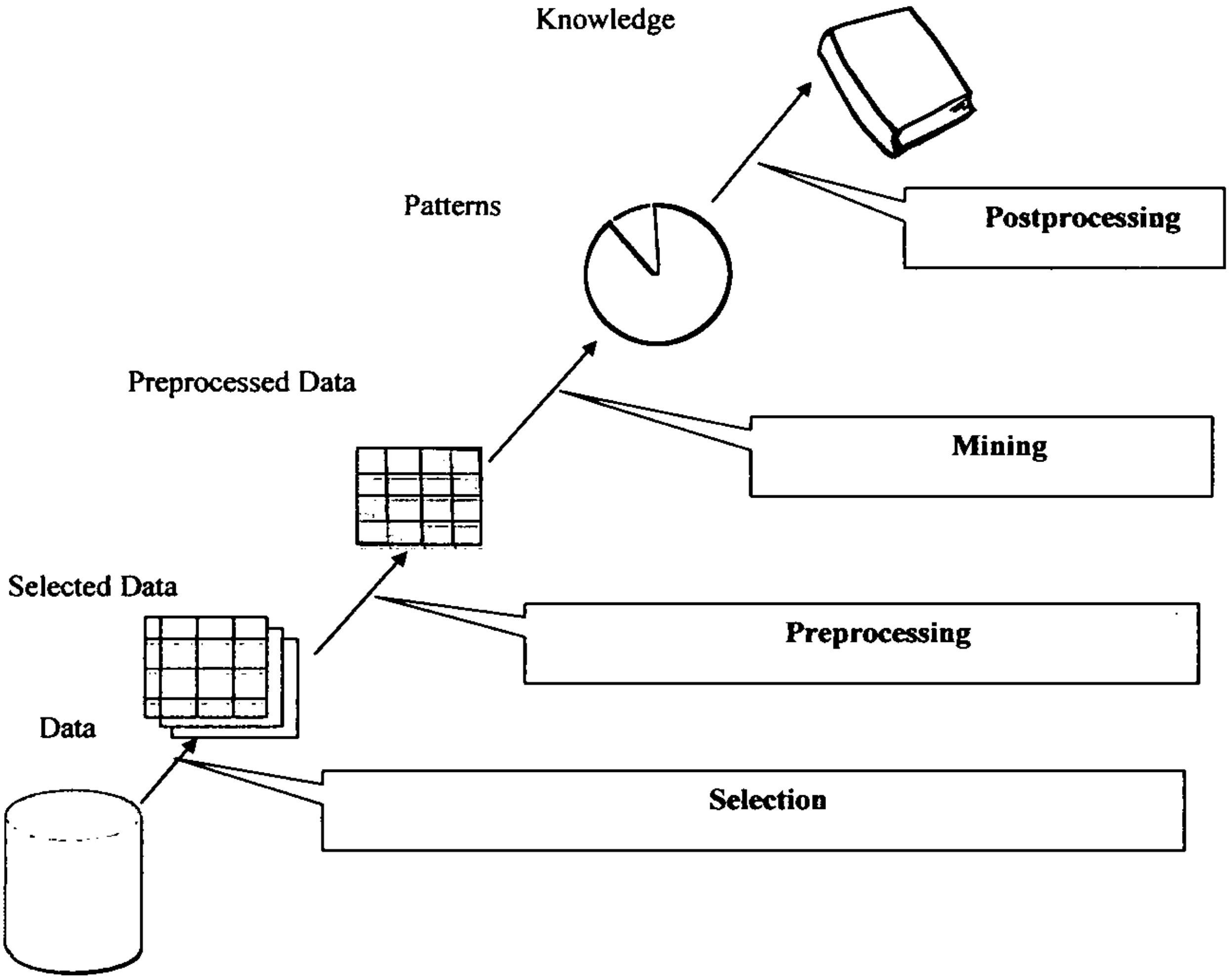


Figure 1-1. A knowledge discovery in databases process

1. 2. 2 Data Mining Methods (Technologies)

We can select the suitable data mining methods (technologies) based on the tasks we have confirmed. Such as, classification model often utilize learning neural network or decision tree to accomplish; while clustering usually utilize clustering analysis algorithms to accomplish; association rules often utilize association and sequence discovery to accomplish.

1. 2. 3 Choose the Algorithms

Based on the technologies have been chosen, we can choose a precise algorithm. Furthermore, a new efficient algorithm can be designed by the particular mining tasks. To pick data mining algorithms, we should find out the hidden pattern in selecting data.

1. 3 Data Mining Applications

Data mining has become an essential technology for businesses and researchers in many fields, the number and variety of applications has been growing gradually for several years and it is predicted that it will carry on to grow. A number of the business areas with an early embracing of DM into their processes are banking, insurance, retail and telecom. More lately it has been implemented in pharmaceutics, health, government and all sorts of e-businesses (Figure 1-2).

Data mining applications have proved greatly effective in addressing many important business problems. Giuffrida et al. , describes a prosperous application of personalization of online advertisement (Giuffrida et al. 2008). They use the Apriori algorithm for association rule mining and focus on the important issues of actionability, integration with the existing Information System and live testing.

An original DM application is described by (Tsai et al. 2008). It is well known that blogs are more and more regarded as a business tool by companies. The authors propose a scheme to investigate and analyze blogs. A particular search engine is built to incorporate the models developed.

Ni et al. describes a scheme to generate a whole set of trading strategies that take into account application constraints, for example timing, current position and pricing (Ni et al.

2008). The authors highlight the importance of developing a suitable backtesting environment that enables the gathering of sufficient evidence to convince the end users that the system can be used in practice. They use an evolutionary computation approach that favors trading models with higher stability, which is essential for success in this application domain.

In some applications, domain-dependent knowledge is integrated in the DM process in all steps except this one, in which off-the-shelf methods/tools are applied. Dong-Peng et al. described one such application where the implementations of decision trees (C4. 5) and association rules (Apriori algorithms) are applied in a risk analysis problem in banking in (Dong-Peng et al. 2008).

(Sharif et al. 2005) employ Apriori algorithm as recommendation engine in an E-commerce system. Based on each visitor's purchase history the system recommends related, potentially interesting, products. It is also used as basis for a CRM system as it allows the company itself to follow-up on customer's purchases and to recommend other products by e-mail.

A government application, proposed by (Luo et al 2008). The problem is connected to the management of the risk associated with social security clients in Australia. The problem is confirmed as a sequence mining task. The actionability of the model obtained is an essential concern of the authors. They concentrate on the difficult issue of performing an evaluation taking both technical and business interestingness into account.

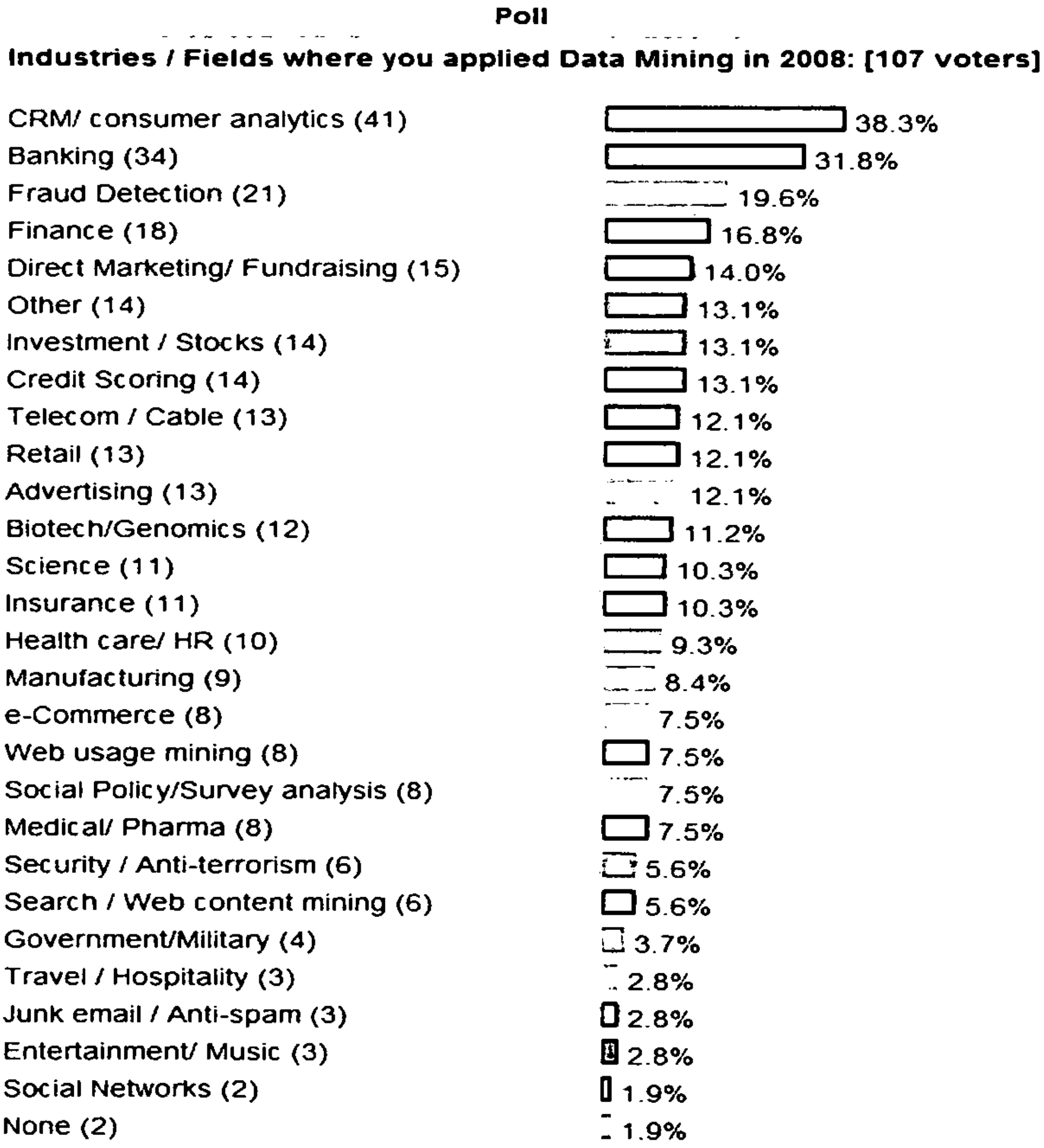


Figure 1-2. Data mining applications in 2008 (http://www. kdnuggets. com).

1. 4 Market Basket Analysis

Market basket analysis, also known as affinity analysis, it allows retailers to rapidly and simply look at the size, contents, and value of their customer's market basket to comprehend the patterns in how products are purchased together, or basic product affinities (Gordon 2008).

With market basket analysis, retailers can drive more profitable advertising and promotions, attract more customers, increase the value of the market basket, and much more.

In market basket analysis, mining knowledge on customer behavior, which is actually helpful to support marketing actions, is a tricky task, which needs non-trivial methods of employing and combining the data mining technologies. The appeal of market basket analysis comes from the clearness and value of its results, which are explicated in the form association rules.

1. 5 Association Rules Mining

Association rules learning or association rules mining are one of the major techniques of data mining and it is perhaps the most common form of local-pattern discovery in unsupervised learning systems. The technique is likely to be very practical in applications which use the similarity in customer buying behavior in order to make peer recommendations.

Association Rules will permit you to discover rules of the kind If X then (likely) Y where X and Y can be particular items ,values, words, etc. , or conjunctions of values, items, words, etc. (e. g. , if (Car=BMW and Gender=Male and Age<20) then (Risk=High and Insurance=High)).

Data patterns and models can be mined from many different kinds of databases, such as Relational Databases, Data Warehouses, Transactional Databases, and Advanced Database Systems (Object-Relational, Spatial and Temporal, Time-Series, Multimedia, Text, Heterogeneous, Legacy, Distributed, and WWW).

1. 6 Problem Statement

Database has been used in business management, government administration, scientific and engineering data management and many other important applications. The newly extracted information or knowledge may be applied to information management, query processing, process control, decision making and many other useful applications. With the explosive growth of data, mining information and knowledge from large databases has become one of the major challenges for data management and mining community.

The association rule mining is motivated by problems such as market basket analysis. A tuple in a market basket database is a set of items purchased by customer in a transaction. An association rule mined from market basket database states that if some items are purchased in transaction, then it is likely that some other items are purchased as well. Finding all such rules is valuable for guiding future sales promotions and store layout.

The problem of mining association rules is essentially, to discover all rules, from the given transactional database D that have support and confidence greater than or equal to the user specified minimum support and minimum confidence. The problem of association rules mining is usually broken down into two sub problems:

1- To find all the sets of items whose support is greater than or equal to the user predetermined minimum threshold.
2- To generate the association rules desired from the frequent itemsets.

Most of the previous studies adopt an a priori-like candidate set generation-and-test approach to generate the association rules from the transactional database. The priori-like candidate approach can suffer from two nontrivial costs: it needs to generate a huge number of candidate sets, and it may need to repeatedly scan the database and check a large set of candidates by pattern matching.

1. 7 Objectives of Thesis

Association rule mining, studied for over ten years in the literature of data mining, its goal is to help enterprises with sophisticated decision making, but the resulting rules typically cannot be straightforwardly applied and need further processing, Our main objectives are:

- To design new scheme for extracting association rules, this considers the time, the memory consumption, and the interestingness of the rules.
- To verify the effectiveness of the new scheme via a case study.

1. 8 Methodology

This thesis is conducted through: a review of the current status and the relevant work in the area of data mining in general and in the area of association rules in particular; analyze these works in the area of mining association rules; propose the new scheme for extracting the association rules that has high efficiency in term of the time and the space; collecting transactional data; analysis the collected data by using the propose scheme to validate its efficiency ; seek avenues for further research.

1. 9 Scope and Limitation

The aim of this research is to develop new scheme for extracting the association rules. To achieve this, different issues regarding association rules mining must be addressed. Addressing all of these issues and developing solution for them in a single research project is impractical. Consequently, the essential and more important issues are addressed in this work. The remaining issues will be left as future works.

The scope of this research is three folds:

- To study and analyze notable frequent itemsets mining approaches.
- To devise a new scheme for extracting the association rules based on the frequent itemsets mining approaches in the previous step.
- To conduct case study in retail industry to validate the proposed scheme.

No single empirical scientific study exists without difficulties and this study is no exception, the data in this study is confidential.

1. 10 Thesis Outline

The rest of the thesis is organized as follows: Chapter two presents formally the problem statement of frequent itemsets mining and shows the recent studies and research in the area of association rules mining. Chapter three Methodology, this chapter describes the main approaches used by our scheme. Chapter four presents a case study using our proposed scheme to find the association rules from the supermarket dataset. Chapter five contains the conclusion and future work.

1. 11 Summary

The main purpose of this chapter is to provide the reader a brief description of the research topic which will be conducted in this thesis. The problem statement and the objectives of the study were introduced. The methodology of this research, and the scope and limitation of research were discussed as well.

CHAPTER TWO: LITERATURE REVIEW

Despite the data mining is new field but amount of the works are enormous. In this chapter, we introduce the background knowledge and related work about the data mining and association rules mining in particular. Initially, section 2.1 describes the original of data mining field, the main tasks, and the methods for dealing with these different data mining tasks. The association rules mining task is discussed in section 2.3 and continues throughout until the last section 2.10, digging into depth of the association rules mining task to explore, the main algorithms which are employed to achieve the goal of this task and what are the types of the association rule that could be obtained from the database.

2. 1 Data Mining

Data mining refers to discover knowledge in huge amounts of data. It is a scientific discipline that is concerned with the analysis observational data sets with the objective of finding unsuspected relationships and produces a summary of the data in novel ways that the owner can understand and use (Han et al 2006).

Data mining is the field of study involves the merging of ideas from many domains rather than a pure discipline the four main disciplines (Tan et al 2006), which are contributing to data mining include:

- Statistics: it can provide tools for measuring significance of the given data, estimating probabilities and many other tasks (e. g. linear regression).
- Machine learning: it provides algorithms for inducing knowledge from given data (e. g. SVM).

- Data management and databases: since data mining deals with huge size of data, an efficient way of accessing and maintaining data is necessary.
- Artificial intelligence: it contributes to tasks involving knowledge encoding or search techniques (e. g. neural networks).

2. 2 The Primary Methods of Data Mining

Data mining addresses two basic tasks: verification and discovery. The verification task seeks to verify users' hypotheses. While the discovery task searches for unknown knowledge hidden in the data. In general, discovery task can be further divided into two categories, which are descriptive data mining and predicative data mining.

Descriptive data mining describes the data set in a summery manner and presents interesting general properties of the data. Predictive data mining constructs one or more models to be later used for predicating the behavior of future data sets.

Predictive models are built, or trained, using data for which the value of the response variable is known. This kind of training is sometimes referred to as *supervised learning*, because calculated or estimated values are compared with the already-known results. Some of the essential predictive methods described are classification, regression, estimation, *and* prediction .

By contrast, descriptive methods are sometimes referred to as *unsupervised learning* because there is no already-known result to guide the algorithms. The important descriptive data mining methods explained are clustering and link analysis (association rule discovery and sequence analysis). The data mining tasks (or problems) can be solved by using two approaches: Supervised problem solving and Unsupervised problem solving (Pujari 2001).

❖ **Supervised problem solving** the collection of techniques where analysis uses a well-defined (known) dependent variable. In supervised learning the aim is to use the available data for building a model which describes one particular variable of interest in terms of the rest of the available data ("class prediction"). All regression and classification techniques are supervised. The supervised learning is driven by real business problems and historical data. And the quality of the supervised learning results dependent on the quality of input data.

❖ **Unsupervised problem solving** this term refers to the collection of techniques where groupings of the data are defined without the use of a dependent variable. In unsupervised learning, the goal is to establish some relationship among all the variables ("class discovery"), there is no variable is singled out as the target. Unsupervised learning attempts to find all patterns or similarities among groups of records without the use of a particular target field or set of predefined classes ("class discovery"). Association rules extracting, Segmentation and Cluster analysis are examples. The unsupervised learning approach is useful when trying to get an initial understanding of data and non-obvious patterns can sometimes pop out of a completed data analysis project. The data mining functions is clarified in Figure 2-1.

	Mining function	**Model type**
Supervised	Classification Regression Estimation Prediction	Predictive modeling
Unsupervised	Association rules Sequence analysis Clustering	Descriptive modeling

Figure 2-1. Summary of primitive data mining tasks

A data mining system may contain one or more of the following data mining tasks (Larose 2006):

Classification: In the Classification the emphasis is on the building of models which is able to assign new instances to the one of a set of well-defined classes then the output attribute will be categorical.

Regression: It is a term borrowed from statistics. Except for its outcome, it is basically the same as the classification task. A classifier outputs are discrete values, while the outputs of the regression model are continuous values.

Estimation: Like classification, the purpose of an estimation model is to determine a value for an unknown output, (by given some input data, the coming up value is for some unknown continuous variable such as income, height, or credit-card balance). However, unlike classification, the output attribute(s) for an estimation problem are numeric rather than categorical.

Prediction: Focus in predicating certain events or behavior, based on historical information. In the predication the same process as classification and estimation except that the records are classified according to some predicted future behavior or estimated future value. The output of the predicative model can be categorical or numeric.

Association rules: The purpose of the Association rule is to finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.

Sequence analysis: It is the same as association discovery, except that the time sequence of events is also considered, which is available with database. For instance, "30% of the people who purchase the product X purchase the product Y within 5 months". In sequence analysis the input data is a set of sequences, called data-sequences and each data sequence is an ordered list of transactions (or itemsets), where each transaction is a set of items.

Clustering Analysis: The main objective of clustering is to find high quality clusters within a reasonable time. However, different approaches to clustering often define clusters in different ways. Traditionally clustering techniques are broadly divided into hierarchical and partitioning methods. Partitioning methods can be further divided into distribution-based, density-based and grid-based methods.

There are a number of algorithmic techniques available for each data mining tasks, with features that must be weighed against data characteristics and additional business requirements. Among all the techniques listed above, in this research, we are focusing on the association rules mining technique, which is descriptive mining technique, with transactional database system. This technique was formulated by (Agarwal et al 1993) and is often referred to as market-basket problem.

2. 3 Introduction to Association Rule Mining

Association rules are one of the major techniques of data mining and it is perhaps the most common form of local-pattern discovery in unsupervised learning systems. It is a form of data mining that most closely resembles the process that most people think about when they try to understand the data mining process. Association rule mining finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories (Han et al 2006). The volume of data is increasing dramatically as the data generated by day-to-day activities. Therefore, mining association rules from massive amount of data in the database is interested for many industries which can help in many business decision making processes, such as cross-marketing, Basket data analysis, and promotion assortment. The techniques for discovering association rules from the data have traditionally focused on identifying relationships between items telling some aspect of human behavior, usually buying behavior for determining items that customers buy together. All rules of this type describe a particular local pattern. The group of association rules can be easily interpreted and communicated.

A lot of studies have been done in the area of association rules mining. Agrawal et al. first introduced the association rules mining in (Agrawal et al. 1993, Agrawal et al 1994). Many studies have been conducted to address various conceptual, implementation, and application issues relating to the association rules mining task.

Researcher in conceptual issues focuses on developing a framework to explain the theoretical underpinnings of association rules mining, expand the formulation to handle new types of patterns, and expand the formulation to incorporate attribute types beyond asymmetric binary data.

The reduction of the itemsets scan to obtain fast mining of association rules was discussed in (Wang et al 2004). On the other hand, (Zaki 2004, Xu et al 2007, Cheng et al 2008) proposed to reduce the number of the extracted association rules.

(Pei et al 2004) worked on mining sequential patterns. Quite a few researchers worked on the alternative patterns, such as discussed unexpected patterns in (Padmanabhan et al 2000). Exception patterns studied in (Taniar et al 2008). And negative association discussed in (Wu et al 2004, Sharma et al 2005, Han et al 2006, Dong et al 2007).

Considerable research has been conducted to expand the original association rule formulation to nominal in (Srikant et al 1996), ordinal (Marcus et al 2001), and interval (Miller et al 1997).

Researcher in implementation issues focuses on integrating the mining potential into existing database technology, and developing efficient and scalable mining algorithms.

The SETM algorithm developed by (Houtsma et al 1995) which supports association rule discovery via SQL queries. To approach real-time data-mining, we need to shorten the time to accomplish this task, there are many researchers who propose to tackle the above problem (Han et al 2000, El-Hajj et al 2003). Many researchers have tried to mine association rules from large database. For example, (Lee et al. 2007) proposed an efficient method for mining all frequent inter-transaction patterns. To mine multidimensional quantitative association rules from relational Jinze Li used linked lists in (Li et al 2007).

Researcher in application issues focuses on applying association rules to a variety of application domains. For example, market basket (Chen et al 2005, Yongmei et al 2009), and network intrusion detection (Barbará et al 2001, Dokas et al 2002, Changguo et al 2009).

2.4 Basic Concepts

(Agarwal et al 1993) defined the problem of finding the association rules from the database. This section introduces the basic concepts of frequent pattern mining for discovery of interesting associations and correlations between itemsets in transactional and relational database. Association rule mining can be defined formally as follows:

I= {i_1, i_2, i_3, ..., i_n} is a set of items, such as products like (computer, CD, printer, papers, ...and so on). Let DB be a set of database transactions where each transaction T is a set of items such that T$\subseteq I$. Each transaction is associated with unique identifier, transaction identifier (TID). Let X, Y be a set of items, an association rule is an implication of the form X→Y, where X$\subset I$, Y$\subset I$, and X$\cap$Y=ø. X is called antecedent of rule (or body of the rule), and Y is called the consequent of the rule (or the head of the rule). An itemset containing *i* items is called an *i*-itemset, The rule X→Y holds in the transaction set D with *Support s*, which is the percentage of transactions that contain an itemset. For an itemset to be interesting, its support must be higher than a user-specified minimum. Such itemset are said to be frequent. So we have the support of the rule X→Y as follows:

Support(X→Y) = (X $\cup$ Y) = (number of transactions containing X & Y)/ (total number of transactions)

If you notice that the rule X→Y = Y→X. there is another measure called *Confidence c*, where *c* is the percentage of transactions in D containing X that also contain Y. This is taken to be the conditional probability, P(Y/X), that is:

Confidence (X→Y) = P(Y/X) = Support(X$\cup$Y) / Support(X)

The occurrence frequency (or support count) of an itemset is the number of transaction that contain the itemset. Then the confidence could be defined also as follows:

Confidence (X→Y) = P(Y/X) = Support count(X$\cup$Y) / Support count(X)

It is often desirable to pay attention to only those rules that may have a reasonably large support such rules with high confidence and strong support, are called *strong rules*. The support and confidence are usually referred as interestingness measures of an association rule. Association rule mining is the process of finding all the association rules that pass the condition of min support and min confidence. In order to mine these rules, first the support and confidence values have to be computed for all of the rules and then compare them with the threshold values to prune the rules with low values of either support or confidence. But this process is computationally expensive because there are an exponential number of rules that can be extracted from a dataset (D) according to the following formula: $R = 3^d - 2^{d+1} + 1$

Where d is the number of items in the dataset and R is the number of extracted rules.

However a large number of these rules will be pruned after applying the support and confidence thresholds. Therefore the previous computations will be wasted. To avoid this problem and to improve the performance of the rule discovery algorithm, mining association rules may be decomposed into two phases:

1- Discover the large itemsets, i.e., the sets of items that have transaction support's' above a predetermined minimum threshold.

2- Use the large itemsets to generate the association rules for the database that have confidence 'c' above a predetermined minimum threshold.

The overall performance of mining association rules is determined primarily by the first step. The second step is easy. After the large itemsets are identified, the corresponding association rules can be derived in straightforward manner.

2. 5 Searching Frequent Itemsets

Frequent patterns, such as frequent itemsets, substructures, sequences term-sets, phrase-sets, and sub graphs, generally exist in real-world databases. Identifying frequent itemsets is one of the most important issues faced by the knowledge discovery and data mining community. Frequent itemset mining plays an important role in several data mining fields (Tan et al 2006) as association rules (Agrawal et al 1994), warehousing (Wu 2006), correlations (Brin et al 1998), clustering of high-dimensional biological data (Wang et al 2002), and classification, (Cheng 2008). Given a data set d that contains k items, the number of itemsets that could be generated is $2^k — 1$, excluding the empty set. In order to searching the frequent itemsets, the support of each itemset must be computed by scanning each transaction in the dataset. A brute force approach for doing this will be computationally expensive due to the exponential number of itemsets whose support counts must be determined. There have been a lot of excellent algorithms developed for extracting frequent itemsets in very large databases. The efficiency of algorithm is linked to the size of the database which is amenable to be treated. There are two typical strategies adopted by these algorithms: the first is an effective pruning strategy to reduce the combinatorial search space of candidate itemsets. The second strategy is to use a compressed data representation to facilitate in-core processing of the itemsets. Below, we give an example of each strategy.

2. 5. 1 Apriori Algorithm

The first algorithm for mining all frequent itemsets and strong association rules was the AIS algorithm by (Agrawal et al. 1993). Shortly after that, the algorithm was improved and renamed Apriori. Apriori algorithm is, the most classical and important algorithm for mining frequent itemsets. Apriori is used to find all frequent itemsets in a given database DB. The key idea of Apriori algorithm is to make multiple passes over the database. It employs an iterative approach known as a breadth-first search (level-wise search) through the search space, where k-itemsets are used to explore (k+1)-itemsets. In the beginning, the set of frequent 1-itemsets is found. The set of that contains one item, which satisfy the support threshold, is denoted by L1. In each subsequent pass, we begin with a seed set of itemsets found to be large in the previous pass. This seed set is used for generating new potentially large itemsets, called *candidate itemsets*, and count the actual support for these candidate itemsets during the pass over the data. At the end of the pass, we determine which of the candidate itemsets are actually large (frequent), and they become the seed for the next pass. Therefore, L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. Then, a very significant property called Apriori property is employed to reduce the search space, where the Apriori property is described as "all nonempty subsets of a large itemset must also be large" or "if a set is not large, then its superset can't be large either". This property belongs to a special category of properties called antimonotone in the sense that if a set cannot pass a test, all of its supersets will fail the same test as well.

Specially, Apriori algorithm consists of join and prune steps:

Join step: This step generates new candidate k-itemsets based on joining L_{k-1} with itself which is found in the previous iteration. Let C_k denote candidate k-itemset and L_k be the frequent k-itemset. Let l_1 and l_2 be itemsets in L_k. The notation $l_i[j]$ refers to the jth item in l_i (e. g., l_1 [k-2] refers to the second to the last item in l_1). By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. For the (k-1)-itemset, li, this means that the items are sorted such that $l_i[1] < l_i[2] <\ldots< l_i[k-1]$. The join, $L_{k-1} \otimes L_{k-1}$, is performed, where members of L_{k-1} are joinable if their first (k-2) items are in common. That is, members l_1 and l_2 of L_{k-1} are joined if $(l_1[1]=l_2[1])\wedge(l_1[2]=l_2[2])\wedge\ldots\wedge(l_1[k-2]=l_2[k-2])\wedge(l_1[k-1]=l_2[k-1])$. The condition $l_1[k-1]<l_2$ [k-1] simply ensures that no duplicates are generated. The resulting itemset formed by joining l_1 and l_2 is $l_1[1]$, $l_1[2]$, …, $l_1[k-2]$, $l_1[k-1]$, $l_2[k-1]$. An example for the join step, if L3 = {{bread, milk, cheese}, {bread, milk, coke}}, then C4= L3⊗ L3= {bread, milk, cheese, coke}.

Prune step: This step eliminates some of the candidate k-itemsets using the Apriori property. C_k is a superset of L_k, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in C_k. A scan of the database to determine the count of each candidate in C_k would result in the determination of L_k (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to L_k). C_k, however, can be huge, and so this could involve grave computation. To shrink the size of C_k, the Apriori property is used as follows. Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any (k-1)-subset of candidate k-itemset is not in L_{k-1} then the candidate cannot be frequent either and so can be removed from C_k. In other words, if an itemset is not frequent, its superset is not frequent either. An example, if C4= {{bread, milk, cheese, coke}, {bread, milk, cheese, shoes}}. To find L4, we found the 3-itemset {bread, milk, shoes} is not in L3 or not frequent itemset. So the candidate 4-itemset {break, milk, cheese, shoes} is removed from C4.

For the explanation of the algorithm, we will use the following example to find the frequent itemsets from transactional database DB (see Table 2-1). There are nine transactions in this database, that is, |DB| = 9. We use Figure 2-4 to demonstrate the Apriori algorithm for mining frequent itemsets in DB.

Table 2-1. The transactional database DB

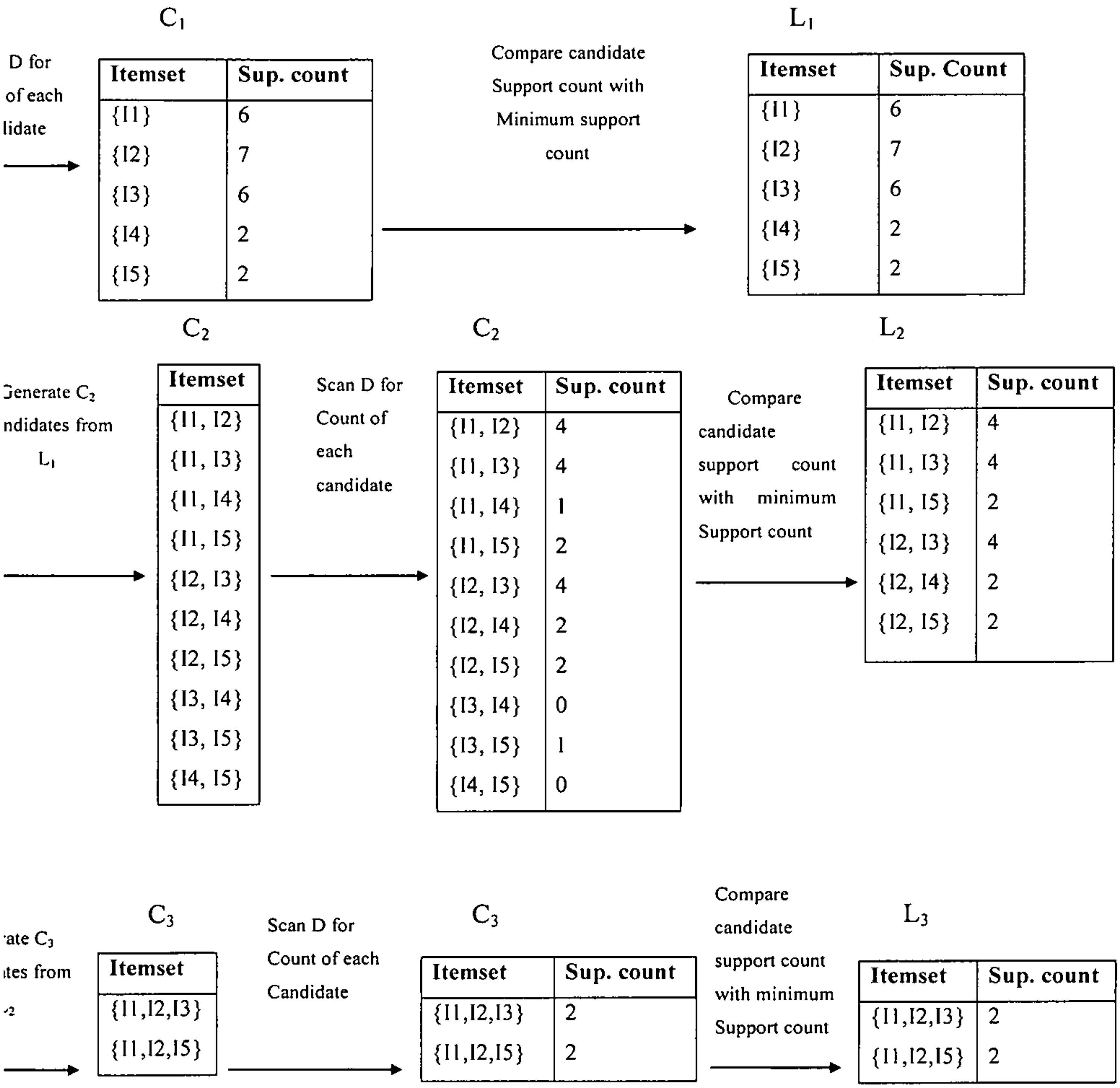
TID	Items
T1	I1, I2, I5
T2	I2, I4
T3	I2, I3
T4	I1, I2, I4
T5	I1, I3
T6	I2, I3
T7	I1, I3
T8	I1, I2, I3, I5
T9	I1, I2, I3

In the first iteration of the algorithm, each item is a member of the set of candidate 1-itemsets, C_1. The algorithm simply scans all of the transactions in order to identify all the individual items (called 1-itemsets) and count the number of occurrences of each individual item.

Assume that the minimum support count required is 2, that is, min_sup = 2. The set of frequent 1-itemsets, L_1, can then be determined. It consists of the candidate 1-itemsets satisfying minimum support. In our example, all of the candidates in C_1 satisfy minimum support (support of the item$\geq$min_sup).

To find out the set of frequent 2-itemsets, L2, the algorithm uses the join L1$\otimes$L1 to generate a candidate set of 2-itemsets which could be potentially frequent, C2. C2 consists of $(2^{|L1|})$ 2-itemsets. Note that no candidates are removed from C2 during the prune step because each subset of the candidates is also frequent.

Next, the transactions in DB are scanned and the support count of each candidate item-set in C2 is accumulated, as shown in the middle table of the second row in Figure 2-2.



D for
of each
lidate

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate
Support count with
Minimum support
count

L_1

Itemset	Sup. Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2
ndidates from
L_1

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for
Count of
each
candidate

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare
candidate
support count
with minimum
Support count

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

ate C_3
ites from
2

C_3

Itemset
{I1,I2,I3}
{I1,I2,I5}

Scan D for
Count of each
Candidate

C_3

Itemset	Sup. count
{I1,I2,I3}	2
{I1,I2,I5}	2

Compare
candidate
support count
with minimum
Support count

L_3

Itemset	Sup. count
{I1,I2,I3}	2
{I1,I2,I5}	2

Figure 2-2. Generation of candidate itemsets and frequent itemsets

The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate 2-itemsets in C2 having minimum support. The generation of the set of candidate 3-itemsets, C3 is detailed in Figure 2-2. From the join step, we first get C3= L2 ⊗ L2 = {{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5}} Based on the Apriori property that all subsets of a frequent itemset must also be frequent, we can determine that the four latter candidates cannot possibly be frequent. We therefore remove them from C3, thereby saving the effort of unnecessarily obtaining their counts during the subsequent scan of D to determine L3. Note that when given a candidate k-itemset, we only need to check if its (k-1)-subsets are frequent since the Apriori algorithm uses a level-wise search strategy. The resulting pruned version of C3 is shown in the first table of the bottom row of Figure 2-2.

The transactions in D are scanned in order to determine L3 consisting of those candidate 3-itemsets in C3 having minimum support (Figure 2-2).

The algorithm uses L3 ⊗ L3 to generate a candidate set of 4-itemsets, C4. Although the join results in {{I1, I2, I3, I5}} this itemset is pruned because its subset {{I2, I3, I5}} is not frequent Thus, C4 =∅, and the algorithm terminates, having found all of the frequent itemsets.

Figure 2-3 shows pseudo-code for the Apriori algorithm and its related procedures. Step 1 of Apriori finds the frequent 1-itemsets, L1. In steps 2 to 10, L_{k-1} is used to generate candidates C_k in order to find L_k for k ≥2. The Apriori_gen procedure generates the candidates and then uses the Apriori property to eliminate those having a subset that is not frequent (step 3). This procedure is described below. Once all of the candidates have been generated, the database is scanned (step 4). For each transaction, a subset function is used to find all subsets of the transaction that are candidates (step 5), and the count for each of these candidates is accumulated (steps 6 and 7). Finally, all of those candidates satisfying minimum support (step 9) form the set of frequent itemsets, L (step 11).

The Apriori_gen procedure performs two kinds of actions, namely, join and prune, as described above. In the join component, L_{k-1} is joined with L_{k-1} to generate potential candidates (steps 1 to 4). The prune component (steps 5 to 7) employs the Apriori property to remove candidates that have a subset that is not frequent. The test for infrequent subsets is shown in procedure has infrequent subset.

```
Algorithm: Apriori
Input:
D: transaction database;
Min_sup: the minimum support threshold
Output: frequent itemsets
Description:
L1 = find_frequent_1-itemsets(DB);
for(k=2;$L_{k-1}$ = φ;k++) {
    Ck = Apriori_gen($L_{k-1}$ );
    for each transaction t ∈ DB { // scan DB for counts
        Ct = subset($C_k$, t); // get the subsets of t that are candidates
        for each candidate c ∈ $C_t$
            c. count++;
    }
    Lk = {c ∈ Ck|c. count ≥ min_sup}
}
return L = $\cup_k L_k$;
procedure Apriori gen($L_{k-1}$:frequent(k-1)-itemsets)
    for each itemset $l_1$ ∈ $L_{k-1}$
        for each itemset $l_2$ ∈ $L_k$
            if ($l_1$[1] = $l_2$[1])^($l_1$[2] = $l_2$[2])^...^($l_1$[k-2]=$l_2$[k-2])^($l_1$[k-1]<$l_2$[k-1])then{
                c = $l_1 \otimes l_2$; // join step: generate candidates
                if has infrequent subset(c, $L_{k-1}$ ) then
                    delete c; // prune step: remove unfruitful candidate
                else add c to Ck;
            }
    return Ck;
procedure has-infrequent-subset(c: candidate k-itemset;
$L_{k-1}$ : frequent (k-1) -itemsets); // use prior knowledge
    for each (k-1) –subset s of c
        if s ∉ $L_{k-1}$ then
            return TRUE;
    return FALSE;
```

Figure 2-3. The pseudo-code of Apriori algorithm (Agrawal et al 1994)

Many other algorithms proposed after the introduction of Apriori keep the same general structure, adding several techniques to optimize certain steps within the algorithm. Partition Algorithm in (Savasere et al 1995, Son et al 2005), partition the data to find candidate itemsets. Dynamic itemset counting in (Brin et al 1997) adds candidate itemsets at different points during a scan. It is used to reduce number of scans on the dataset. Sampling approach in (Toivonen 1996, Mahafzah et al 2009) the basic idea is to mine on a subset of the given data.

2. 5. 2 FP-growth Algorithm

FP-growth algorithm is an efficient method of mining all frequent itemsets without candidate's generation. FP-growth utilizes a combination of the vertical and horizontal database layout to store the database in main memory. Instead of storing the cover for every item in the database, it stores the actual transactions from the database in a trie structure and every item has a linked list going through all transactions that contain that item. This new data structure is denoted by FP-tree (Frequent-Pattern tree) (Han et al 2000). Essentially, all transactions are stored in a trie data structure. Every node additionally stores a counter, which keeps track of the number of transactions that share the branch through that node. Also a link is stored, pointing to the next occurrence of the respective item in the FP-tree, such that all occurrences of an item in the FP-tree are linked together.

Additionally, a header table is stored containing each separate item together with its support and a link to the first occurrence of the item in the FP-tree. In the FP-tree, all items are ordered in support descending order, because in this way, it is hoped that this representation of the database is kept as small as possible since all more frequently occurring items are arranged closer to the root of the FP-tree and thus are more likely to be shared.

The algorithm mine the frequent itemsets by using a divide-and-conquer strategy as follows: FP-growth first compresses the database representing frequent itemset into a frequent-pattern tree, or FP-tree, which retains the itemset association information as well.

The next step is to divide a compressed database into set of conditional databases (a special kind of projected database), each associated with one frequent item. Finally, mine each such database separately.

Particularly, the construction of FP-tree and the mining of FP-tree are the main steps in FP-growth algorithm.

For the explanation of the algorithm, we will use the following example to find the frequent itemsets from transactional database DB (see Table 2-2). First, a scan of the database DB derivers a set of frequent 1-itemsets (L) which also include their support count. The set L is sorted in the order of descending support count, this ordering is important since each path of FP-tree will follow it.

Let the minimum support count be 3, then the set L= {(f, 4), (c, 4), (a, 3), (b, 3), (m, 3), (p, 3)}.

Table 2-2. The transactional database DB

TID	Items
T1	f, a, c, d, g, i, m, p
T2	a, b, c, f, l, m, o
T3	b, f, h, j, o
T4	b, c, k, s, p
T5	a, f, c, e, l, p, m, n

Second, an FP-tree is constructed as follows: The root of the tree, labeled Null, is created. The database DB is scanned for the second time. The items in each transaction are processed in L order, and a branch is created for each transaction.

For example, the scan of the first transaction, "T1: f, a, c, d, g, I, m, p" which contains five items (f, c, a, m, p in L order). Only those items that are in the list of frequent itemsets L, leads to constructions of the first branch of the tree with tree nodes {<f, 1>, <c, 1>, <a, 1>,

<m, 1>, <p, 1>} where <f, 1> is linked as a child of the root. <c, 1> is linked to <f, 1>, <a, 1> is linked to <c, 1>, <m, 1> is linked to <a, 1>, and <p, 1> is linked to <m, 1>.

The second transaction, because it shares items f, c and a, it shares the common prefix {f, c, a} with the previous branch and extends to the new branch {<f, 2>, <c, 2>, <a, 2>, <m, 1>, <p, 1>}, increasing the count of the common prefix by 1. The new intermediate version of FP-tree, after adding two transactions from the database, is given in Figure 2-4. For the remaining transactions can be inserted in the same way (see Figure. 2-5).

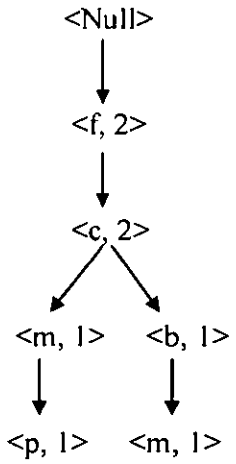


Figure 2-4. FP-tree for two transactions

Item ID	Support Count	Node-link
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

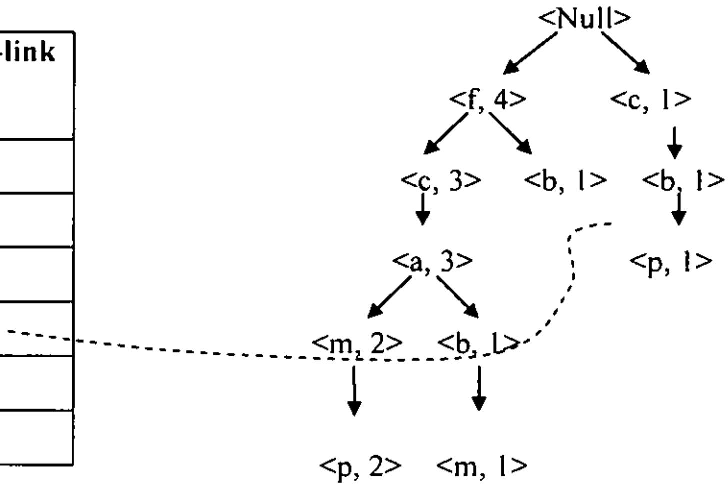


Figure 2-5. Final FP-tree

To ease tree traversal, header table is built so that each item points to its occurrences in the tree via chain of node-link.

Using the compact tree structure (or FP-tree), the FP-growth algorithm mines all the frequent itemsets. The FP-tree is mined as follows. Begin from each frequent-1 pattern (as an initial suffix pattern), construct its conditional pattern base (a "subdatabase" which consists of the set of prefix paths in the FP-tree co-occurring with suffix pattern), then build its (conditional) FP-tree, and do mining recursively on such a tree. The patterns growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree.

In our example, according to L, the complete set of frequent itemsets can be divided into subsets (6 for our example) without overlapping, first, frequent itemsets having items p (as an initial suffix pattern), which is the last item in L, rather than the first item. The reason for starting at the end of the list will become clear as we explain the FP-tree mining process. Second, the itemsets having item m but not p; third, the itemsets that have item b without both m and p; we continue this process to the end. Therefore, the last set will be the large itemsets only with f.

The item p occurs in two branches of the FP-tree of Figure 2-5. The occurrences of p can easily found by starting from the header table of p and following p's node-links. The paths formed by these branches are{<f, 4>, <c, 3>, <a, 3>, <m, 2>, <p, 2>} and {<c, 1>, <b, 1>, <p, 1>} where samples with a frequent item p are {<f, 2>, <c, 2>, <a, 2>, <m, 2>, <p, 2>} and {<c, 1>, <b, 1>, <p, 1>}, which form its conditional pattern base, these samples are the transactions that contain the branch of the tree with the existing of item p. Its conditional FP-tree contains only {<c, 3>}, the other items are not included because its support count is less than 3. The generated frequent itemset that satisfy the minimum support count is {<c, 3>, <p, 3>}, all the other itemsets are below the minimum support count.

The next subsets of frequent itemsets are those with m item and without p. The FP-tree recognizes the paths {<f, 4>, <c, 3>, <a, 3>, <m, 2>}and {<f, 4>, <c, 3>, <a, 3>, <b, 1>, <m, 1>}, or the related accumulated samples {<f, 2>, <c, 2>, <a, 2>, <m, 2>} and {<f, 1>, <c, 1>, <a, 1>, <b, 1>, <m, 1>} . Analyzing the samples we find the frequent itemset {<f, 3>, <c, 3>, <a, 3>, <m, 3>}.

Similar to subset 3 to 6 the same process is done in our example, additional frequent itemsets can be mined. These are itemsets {f, c, a} and {f, c}, but they are already subset of frequent itemsets {f, c, a, m}. Therefore, the final set of frequent itemsets is {{c, p}, {f, c, a, m}}. The pseudo-code of FP-growth algorithm is shown in Figure 2-6.

Algorithm: FP-growth

Input:

DB: transaction database;

Min_sup: the minimum support threshold

Output: frequent itemsets

Description:

1. The FP-tree is constructed in the following steps:

(a) Scan the transaction database DB once. Collect F, the set of frequent items, and their support counts, Sort F in support count descending order as L, the list of frequent items.

(b) Create the root of an FP-tree, and label it as "null. " For each transaction Trans in DB do the following: Select and sort the frequent items in Trans according to the order of L. Let the sorted frequent item list in Trans be [p/P], where p is the first element and P is the remaining list. Call insert tree([p/P], T), which is performed as follows. If T has a child N such that N. item-name=p. item-name, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree(P, N) recursively.

2. The FP-tree is mined by calling FP-growth (FP tree, null), which is implemented as follows.

```
Procedure FP growth (Tree, $\alpha$)
if Tree contains a single path P then
    for each combination (denoted as $\beta$) of the nodes in the path P
        generate pattern $\beta \cup \alpha$ with support_count = minimum support count of nodes in $\beta$;
else for each $a_i$ in the header of Tree {
    generate pattern $\beta$ = ai$\cup\alpha$ with support_count = $a_i$. support_count;
    construct $\beta$'s conditional pattern base and then $\beta$'s conditional FP_tree Tree$\beta$;
    if Tree$\beta \neq \emptyset$ then
        call FP_growth(Tree$\beta$, $\beta$); }
```

Figure 2-6. The pseudo-code of FP-growth algorithm (Han et al 2000)

FP-growth algorithm, its scalable frequent patterns mining method has been proposed as an alternative to the Apriori-based approach. This algorithm is faster than others in the literature, this reported by the authors of this algorithm. Several algorithms implicate the methodology of the FP-growth algorithm. Further improvements of FP-growth mining methods were introduced. (Grahne et al 2005, Gao 2007, Kumar et al. 2007) adapted the similar approach of (Han et al 2000) for mining the frequent itemsets from the transactional database. The authors reported that the performances of these algorithms are more efficient than FP-growth.

2. 6 FP-growth Variations Efficiency Scaling

Several optimization techniques are added to FP-growth algorithm. These algorithms follow the similar approach of FP-growth algorithm (Frequent pattern growth). In the following we will illustrate what are the main optimization ideas in each algorithm.

- **AFOPT Algorithm**

Liu et al in (Liu et al 2003) investigated the algorithmic performance space of the FP-growth algorithm. They specified the problem of conditional databases construction (particularly the number of the conditional databases constructed and the mining cost of each individual conditional database) in FP-growth algorithm, which have direct effect on the performance of the algorithm. They studied the problem of enhancing the FP-growth algorithm from four perspectives to come with the best strategy for mining the frequent itemsets. These perspectives are the item search order (in what order the search space is explored), conditional database representation, conditional database construction strategy and tree traversal strategy.

For the first part of the problem, the number of the conditional databases constructed can differ very much using different items search orders. The dynamic ascending order is able to minimize the number and /or the size of the conditional database constructed in subsequent mining, AFOPT algorithm adopt this kind of items search order which is also used by FP-growth.

For the second part of the problem, the mining cost of each individual conditional database is heavily depends on its representation (tree-based or array-based). AFOPT algorithm use adaptive representation, tree-based structure in the case of dense dataset and array–based representation in the case of sparse dataset. In additions to the conditional database representation the size and the conditional database construction strategy have effect on the mining cost of each individual conditional database, two type of the conditional database construction strategy (physical construction or pseudo-construction).

The dynamic ascending frequency search order can make the subsequent conditional databases shrink rapidly. As a result, it is useful to use the physical construction strategy with the dynamic ascending frequency order. The traversal cost of a tree is minimal using the top-down traversal strategy, AFOPT algorithm uses dynamic ascending frequency order for both the search space exploration and prefix-tree construction, and it uses the top-down traversal strategy. as a summary AFOPT algorithm utilizes dynamic ascending frequency for the item search space, adaptive representation for the conditional database format, physical construction for the conditional database construction, and top-down traversal strategy for the tree traversal, the pseudocode is shown in Figure 2-7 .

```
Algorithm: AFOPT-all

Input:
P is a frequent itemset
$D_p$ is the conditional database of p
min_sup is the minimum support threshold;

Description
Scan $D_p$ count frequent items, $F=\{i_1, i_2, \ldots, i_n\}$;
Sort items in F in ascending order of their frequencies;
For all item $i \in F$ do
    $D_{p \cup \{i\}}=\emptyset$;
For all transactions $t \in D_p$ do
    Remove infrequent items from  t, and sort remaining items according to their frequencies in F;
    Let i the first item of t, insert t into $D_{p \cup \{i\}}$ ;
For all item  i  $\in$ F do
    Out $s=p \cup \{i\}$;
    AFOPT –all (s, $D_s$, min_sup);
    PushRight ($D_s$) ;
```

Figure 2-7. The pseudo-code of AFOPT algorithm (Liu et al 2003)

- **NONORDFP Algorithm**

It is one of FP-growth algorithm variations, it based on a variation of FP-tree, a similar data structure that was used by FP-growth algorithm, but this new version of FP-tree is more compact and there is no need to rebuild it for each conditional step. It is more compact because its node only stores the counters of the item and the parent pointer of the item. This will help to fast allocation, traversal (the data structure is traversal from bottom to up because it is trie), and optionally projection of the tree. There is no need to rebuild the tree because it allows for more recursive steps to be carried out on the same tree (data structure), without need to rebuild it. But there are drawbacks for never rebuild the tree, and then it uses the projection to filter the conditionally infrequent items. The order of the left items cannot be changed to adapt to the conditional frequencies, which is why the name of the new data structure is nonordfp (non order fp-tree).

The running time and the space required for the FP-growth algorithm were the motivation for Nonordfp algorithm. Rácz in (Rácz 2004) dealt with the implementation issues, data structures, memory layout, I/O, and library functions. A compact, memory efficient representation of an FP-tree by using trie data structure, with memory layout that allows faster traversal was introduced, to deal with the running time and space requirement problem. This compact representation of FP-tree allows faster allocation, traversal, and optionally projection. It contains less administrative information about the items in the database (no labels for the items are stored in the node, no header lists and children are required), and allows more recursive steps to be carried out on the same data structure, with no need to rebuild it. The pseudocode is shown in Figures 2-8, 2-9, 2-10.

Algorithm: Nonordfp- core Algorithm

```
Recursion (condition, nextitem, structure, counters):
For citem=nextitem-1 downto 0 do {
    If support of citem <min_supp then {
        Continue at next citem}
    newcounters=aggregate conditional pattern base for condition $\cup$ citem ;
    If projection is beneficial then {
        newstructure=projection of structure to newcounters;
        Recursion (condition $\cup$ citem, citem, newstructure, newcounters)}
    Else
        Recursion (condition $\cup$ citem, citem,  structure,  newcounters)} //end of for loop
```

Figure 2-8. The pseudo-code of NONORDFP algorithm (Rácz 2004)

Algorithm: Nonordfp- Aggregation on the compact trie data structure

cpb-aggregate (item, parents, itemstarts, counters, newcounters, condfreqs)

Input: item is the identifier of the item to add to the current condition, parents and itemstarts describe the current structure of the tree, counters and newcounters hold the current and new conditional counters of the nodes: counters is an itemstarts[item+1] sized array, newcounters is an itemstarts[item] sized array, condfreqs will hold new conditional frequencies of the items. This is the default (dense) aggregation algorithm.

```
Fill newcounters and condfreqs with zeroes;
For n=itemstarts[item] to itemstarts[item+1]-1 do{
    newcounters[parents[n]]=counters[n];}
For citem=item-1 downto 0 do{
    For n=itemstarts[citem] to itemstarts[citem+1]-1 do{
        newcounters[parents[n]]+=newcounters[n];
        condfreqs[citem]+=newcounters[n];}}
```

Figure 2-9. The pseudo-code of NONORDFP algorithm (Rácz 2004)

Algorithm: Nonordfp -Projection of the compact trie data structure
Project(item, parents, itemstarts, newcounters, condfreqs,
newparents, newitemstarts, newnewcounters)

Input: newcounters and condfreqs as computed by the aggregation algorithm,
newparents and newitemstarts will hold the projected structure,
newnewcounters will hold the values of newcounters reordered accordingly.

The array newcounters is reused during the algorithm to store the old position to new position mapping.

```
newcounters[0]=0;                           {node 0 is reserved for the root}
nn=1;                                       {the next free node}
For citem=0 to item-1 do
    newitemstarts[citem]=nn;
    For n=itemstarts[citem] to itemstarts[citem+1]-1 do
        If condfreqs[citem]<min_supp or newcounters[n]==0 then
            newcounters[n]=newcounters[parents[n]]   {skip this node, the new position will be the same as the
                                                       parent's}
        Else
            newnewcounters[nn]=newcounters[n];
            newcounters[n]=nn;                       {save the position mapping}
            newparents[nn]=newcounters[parents[n]];  {retrieve the new position of the parent from the saved
                                                       mapping}
            nn++;
        End if
    End for
End for
newitemstarts[item]=nn;
```

Figure 2-10. The pseudo-code of NONORDFP algorithm (Rácz 2004)

- **FPGROWTH* Algorithm**

Depending on a numerous experiments were done by (Grahne et al 2004), they found that 80% of the CPU time was used for traversing FP-trees. Consequently, they employed the array-based technique (FP-array technique) to reduce the traversal time of the FP-trees. Fpgrowth* algorithm uses FP-tree data structure in combination with the array-based and incorporates various optimization techniques.

In the case of sparse data set the array-based technique works very well, the array save traversal time for all items and the next level of FP-trees can be initialized directly. While in the case of dense data set, the FP-tree is more compact. To deal with this problem they proposed optimizing technique that help the algorithm to estimate if the data set is sparse or dense, by counting the number of the nodes in each level of the tree which done during the construction of each FP-tree. If the data set turns to be dense data set then no need to calculate the array for the next level of the FP-tree. In the case of sparse data set, the calculation of the array for the next FP-tree is required, Figure 2-11 shows the Fpgrowth* algorithm.

```
Fpgrowth*-all algorithm
Input: A conditional FP-tree T
Output: The complete set of all frequent itemset's corresponding to T
If T only contains a single path P
    Then for each subpath Y of P
        Output itemset Y ∪ T. base with count= smallest count of nodes in Y;
Else for each i in T. header do begin
    Output Y=T. base ∪ {i} with i. count;
    If T. FP-array is not NULL
        Construct a new header table for Y's FP-tree from T. FP-array;
    Else construct a new header table from T;
    Construct Y's conditional FP-tree Ty and its FP-array Ay;
    If Ty≠∅
        Call Fpgrowth*(Ty);
```

Figure 2-11. The pseudo-code of Fpgrowth* algorithm (Grahne et al 2004)

2. 7 Searching Infrequent Itemsets

All of the previous studies are about to discover association rules among frequent items. Even though association rule mining among frequent items plays an important role in decision-making, interesting rules (patterns) can still occur among infrequent items. For example, in a supermarket, the expensive diamond necklace and earring are rarely purchased items, and the much inexpensive items like child's clothes/shoes are frequently purchased items. The profits made by selling one piece of diamond would be as selling hundreds of child's clothes. So the manager would like to know what kind of diamond necklace and earring are customers likely to purchase together on a given trip to the store. Therefore, association rule mining among infrequent items can also capture critical information in our world. Therefore, infrequent itemset mining recently has received a lot of attentions such as the researches performed in (Dong et al 2007, Zhou et al 2007).

2. 8 Association Rules Mining

Once the frequent itemsets from transactions in a data set have been found, it is straightforward to generate all frequent and confidence association rules from them (Agrawal et al 1994). The general form of the association rule is:

(Antecedents of the rule) → (Consequent of the rule) (support, confidence)

If Body then Head

Each Frequent K-itemset (K=1, 2, 3, . . .), A, can produce up to 2^k-2 association rules, for instance we have frequent itemset contains 3 items then the number of potential association rules is 6, we subscribe the 2^K by two because we are ignoring rules that have empty antecedents or consequents ($\emptyset \rightarrow$A) or (A$\rightarrow\emptyset$) those rules always hold confidence equal to 100%. An association rules can be generated by partitioning the itemset A into two non-empty subsets, X and X-Y, such that X→X-Y, (where strong association rules satisfy both minimum support and minimum confidence). Note that for the support of the rules its already satisfied because they are generated from frequent itemsets and for the confidence of the association rules to compute it does not require additional scans of the transaction data set (that is because the support of the antecedent and the consequent of the association rules have already counted in the stage of finding frequent itemsets).

$$\text{Support } (X \rightarrow Y) = P(X \cup Y)$$

$$\text{Confidence } (X \rightarrow Y) = P(Y|X) = \frac{\text{Support\_count } (X \cup Y)}{\text{Support\_count } (X)}$$

As another important property of measuring the support and the confidence of the association rules is to work as brute-force approach, because the number of the extracted rules can be very huge, more specifically, the total number of the possible rules, R, extracted from a data set that contains d items is:

$$R=\sum_{k=1}^{d1}[(k^{d})\sum_{j=1}^{d-k}(j^{d-k})] == 3^{d}-2^{d+1}+1$$

Theorem 2.1: if a rule X→Y-X does not satisfy the confidence threshold, then any rule X'→Y-X', where X'⊂X, must not satisfy the confidence threshold as well.

An example to clear the idea of theorem 2.1, we have the frequent itemset X= {a, b, c, d}, the lattice structure for the association rules generated from the frequent itemset is shown in Figure 2-12.

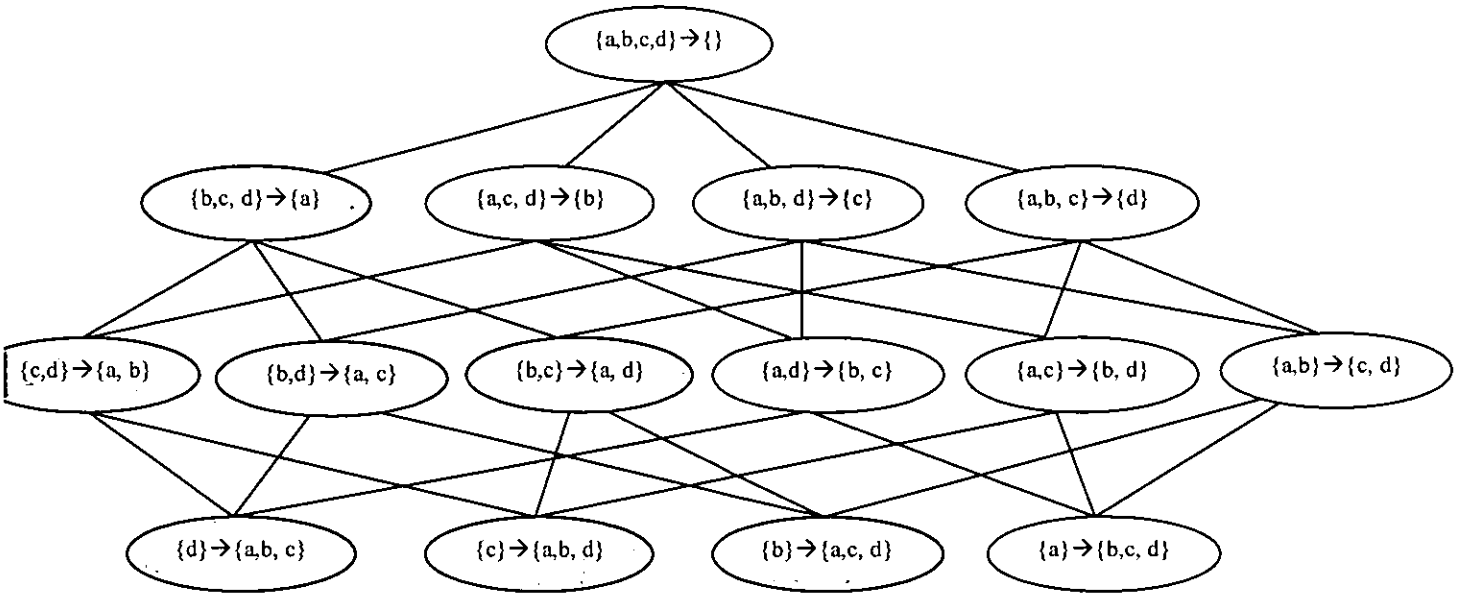


Figure 2-12. Lattice structure for the association rules

According to the theorem 2.1, any node of the lattice structure has low confidence value then the entire sub graph spanned by the node pruned directly. For instance, assume the node that contains {b, c, d}→{a} has low confidence value. Then all the nodes spanned by this node will prune directly, as shown in the lattice structure all the highlight nodes are pruned.

A pseudocode for rule generation step is shown in Figure 2-13 and Figure 2-14. This algorithm adopts level-wise approach for extracting the association rules, where each level corresponds to the number of items that belong to the rule consequent (head of the rule). In the beginning the algorithm extracts all the high-confidence rules that have only one item in the rule consequent. Then these rules are used to generate the new candidate rules. For instance, if the itemset Y= {a, b, c, d}, and at the first level, we obtain {a, c, d}→{b} and {a, b, d}→{c} as a high confidence rules, then the candidate rule will be {a, d}→{b, c} is generated by merging the consequents of both rules, as shown in line 8. For the confidence calculation, there is no need to make additional pass over the data set to compute it. Instead, we determine it by using the support counts computed during the frequent itemsets mining.

Rule generation of the Apriori algorithm

Input: frequent itemset

Output: Association rules

```
For each frequent K-itemset $f_k$, $K \geq 2$ do    { $f_k$ :the frequent itemset}
    $H_1 = \{i | i \in f_k\}$;                      {1-item consequents of the rule}
    Call Apriori-generate-rules ($f_k$, $H_1$);     {Apriori-generate-rules(): is a function}
End for
```

Figure 2-13. Apriori rule generation algorithm (Agrawal et al 1994)

Apriori-generate-rules (f_k, H_m)

```
$k = |f_k|$;                                         {the size of the itemset}
$m = |H_m|$;                                         {the size of the consequent}
If $k > m+1$ then
    $H_{m+1}$ =Apriori-gen ($H_m$);                  {candidate head generation}
    For each $h_{m+1} \in H_{m+1}$ do
        Confidence=support ($f_k$)/support($f_k$-$h_{m+1}$) ;   {calculate the confidence}
        If Confidence $\geq$ min_confidence then
            Output the rule ($f_k - h_{m+1}$) $\rightarrow h_{m+1}$ ;
        Else
            Delete $h_{m+1}$ from $H_{m+1}$ ;
        End if
    End for
    Call Apriori-generate-rules ($f_k$, $H_{m+1}$)   {recursion}
End if
```

Figure 2-14. Apriori rule generation algorithm (Agrawal et al 1994)

Fortunately, if the number of the association rules is not too large, then the time needed to finding all such rules consists mainly of the time that was needed to find all frequent itemsets. To the best of our knowledge, Since the proposal of Apriori association rules algorithm which generates the association rules from frequent itemset there is no significant optimization have been done anymore and almost all the researches have been focused on the way of mining the frequent itemsets

2. 9 Types of Association Rules Mining

In fact there are many kinds of association rules, frequent patterns, and correlations relationships, Association rules can be classified in various ways (Han et al 2006), based on the following criteria:

❖ Based on the types of values handled in the rule (categorical data, numerical data):

The categories can include mining Boolean association rules and Quantitative association rules, as follows:

Boolean association rule: if a rule concerns associations between the absence and the presence of items (0 refer for absence and 1 refer for presence of the item) (Wur et al. 1999), for instance:

Printer→Papers (support=20%, confidence=80%),

it is an example of Boolean association rule.

Quantitative association rule: if a rule describes association between quantitative items or attributes, in such rules, quantitative values for items or attributes are partitioned into intervals then use any algorithm for finding Boolean Association Rules (Tsai et al 2001), for instance:

Income(X, "10K...40K") ^ Age(X, "25...50")→Buys(new laptop)(support = 8%, confidence = 70%)

Where x is a variable representing a customer, is an example of quantitative association rules, quantitative attributes are discretized.

- *Based on the levels of abstractions involved in the rule set:*

Some methods for association rule mining can be find rules at differing levels of abstraction as follows:

Single-level association rules: rules do not refer items or attribute at different level (Agarwal et al 1993). The approach for finding this type of rules is by using any algorithm for finding Boolean Association Rules on the single level of items' abstractions.

Multilevel association rules: Multi-Level Association Rule from Items often form hierarchies in Transaction Database (Computer: desktop (IBM, Dell), laptop (Toshiba, Sony)) (Han et al. 1995), the following rules are multilevel association rules.

Buys (X, "computer") →Buys(X, "Canon printer")
Buys (X, "laptop") →Buys (X, "Canon printer")

In the previous rules the items bought are referenced at different levels of abstraction (e. g., "computer" is higher-level abstraction of "laptop"). In another word we can define the level of the rule depends on the brand of the items that are from the same category (See Figure 2-15).

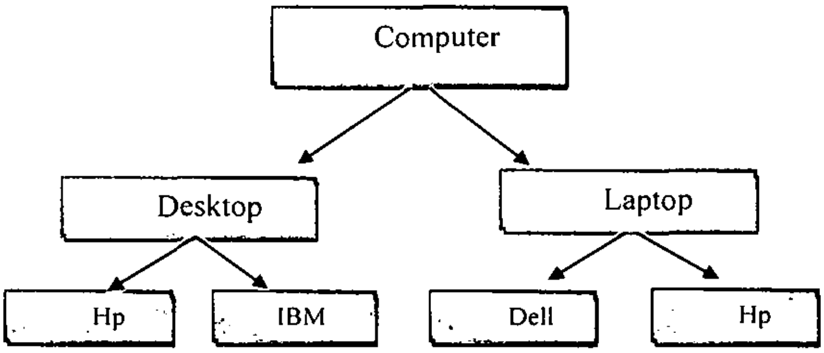


Figure 2-15. The multilevel of the abstraction

- Based on the dimensions of data involved in the rule:

Single-dimensional association rule: items or attributes in an association rule refer to only one dimension, Single-Dimensional Association Rule from Transaction Database (Agarwal et al 1993). For instance:

Buys (X, "computer") →Buys (X, "antivirus_software")

Where X represents customer, is a single-dimensional rule since it refers to only one dimension, Buys.

Multidimensional association rule: a rule refers to two or more dimensions, Multi-Dimensional Association Rule from Data Warehouse and Relational Data Base (Xu et al. 2006), for example, the rule

Age (X, "25...40") & Income(X, "30k...70k") →Buys (X, "new laptop")

Where X represents customer, is a Multidimensional association rule since it involves dimensions Age, Income, and Buys. The Multidimensional association rule categorizes as the repeated of the predicates in the rules as follow:

- Inter-dimension association rules (*no repeated predicates*)

  Age (X, "20-26") ∧ Occupation(X, "student") → Buys(X, "coke")

- hybrid-dimension association rules (*repeated predicates)*

  Age (X, "20-26") ∧ Buys(X, "popcorn") → Buys(X, "coke")

- Based on the kinds of patterns to be mined: many kinds of association rule can be mined from different kinds of data sets.

Frequent itemset mining: that is, the mining of frequent itemsets (set of items) from transactional or relational data sets (Agarwal et al. 1993). However, other kinds of frequent patterns can be found from other kinds of data sets.

Sequential pattern mining: search for frequent subsequences in a sequence data set (Agrawal et al. 1995, Pei et al 2004). For example with Sequential pattern mining, we can study the order in which items are frequently purchased. For instance, customers may tend to first buy a PC, followed by a printer, and then a memory card.

Structured pattern mining: searches for frequent sub structures in a structured data set (Han et al. 2004). Notice that structure is a general concept that covers many different kinds of structural forms, such as graph, lattices, trees, sequences, sets, single items, or combinations of such structures. Single items are the simplest form of structure. Each element of an itemset may contain a subsequence, sub tree, and so on. And such containment relationships can be defining recursively. Therefore, structured patterns mining can be considered as the most general form of frequent pattern mining.

Choosing the type of the rules is mainly based on the type of the application and the data. Association rules are good for complete data and discrete values, complete support from data. Approximate association rules are good for database contains missing or noisy data and small variations (Gerardo et al. 2004), partial support from data (Nayak et al. 2001), fuzzy association rules-from fuzzified data (Chen et al. 2002).

2. 10 Association Analysis to Correlation Analysis

Association rules, and the support-confidence framework used to mine them, are well suited to the market-basket problem. Most association rule mining algorithms employ a support-confidence framework for the discovery of interesting rules. Although these two parameters (minimum support: is the minimum number of occurrences of some set of attributes in a dataset (referred to as *itemsets*), and minimum confidence: Confidence is an indication of the support for an AR in a rule set, i.e. how "confident" we are about the validity of a rule. Confidence is expressed as a percentage and is calculated by dividing the support for the union of the antecedent and consequent of an AR by the support of just its antecedent) prune many associations discovered, many rules that are not interesting to the user may still be produced. The bigger problem of the support-confidence framework does not work well when correlation is appropriate measure. In order to ameliorate the criterion, a correlation measure is used to augment the support-confidence framework for association rules; this generates correlation rules of the form:

(Antecedents of the rule) → (Consequent of the rule) (Support, confidence, correlation)

Several objective measurements were introduced in the literature, among those three popular objective measurements which are correlation, cosine, and interest (Tan et al 2006); Table 2-3 shows the measurement and its range of values.

Table 2-3. Objective measurements

Correlation measurement	**Range**
Correlation Coefficient = $\frac{P(AB)-P(A)P(B)}{\sqrt{P(A)P(B)P(\bar{A})P(\bar{B})}}$	The value of correlation ranges from -1 (perfect negative correlation) to +1 perfect positive correlation, if it is 0 then they are independent.
Cosine = $\frac{P(AB)}{\sqrt{P(A)P(B)}}$	The value of correlation ranges from 0 (negative correlation) to 1 (positive correlation) if it is $\sqrt{P(A,B)}$ then they are independent.
Interest= $\frac{P(AB)}{P(A)P(B)}$	The value of correlation ranges from <1 (negative correlation) to >1 (positive correlation), if it is 1 then they are independent.

For the correlation calculation, there is no need to make additional pass over the data set to compute it. Instead, we determine it by using the support counts computed during the frequent itemsets mining.

2. 11 Summary

This chapter has introduced the fundamental information that helps the reader to understand the basic concepts and terms presented in the rest of this thesis. Initially, section 2.1 presents the original of data mining field, the main tasks, and the methods for dealing with these different data mining tasks. Section 2.4 shows the basic concepts of mining frequent itemsets and association rules. In section 2.5 we have talked about searching for the frequent itemsets and describe the approach of two algorithms (Apriori and FP-growth). Section 2.6 present descriptions of notable algorithms, and their characteristics. Section 2.8 and section 2.9 present the association rules generation method and the types of the associations rules could be find from the frequent itemsets. Finally section 2.10 introduces the idea of moving from extracting the association rules to extracting the correlation rules.

CHAPTER THREE: METHODOLOGY

3. 1 Introduction

Management of large supermarket has to consider many important elements in relation to the items that should be displayed. Among these elements are the following; what items we should sale together? How to design the coupons? How to organize merchandise on the shelves in order to increase the earnings and maximize profit?. Analysis of historical data is quite common used approach in this regard in order to improve quality of such important decisions. Until recently, only global data about the cumulative sales during some time period was available on the computer. However, progress in bar-code technology has made it possible to store the so called basket data that stores items purchased on a per-transaction basis. In this chapter, we consider the problem of "mining" a large collection of basket data type transactions for finding association rules between sets of items, we describe the main methodologies for analyzing the transactional dataset. Section 3.2 we set the business goal of using the data mining. Section 3.3 and 3.4; describe the processes of preparing the data to make it proper for model building. In section 3.5, we describe the model building stage in which we are using our new scheme for analyze the data set, we describe the main architecture of our new scheme and how we choose the main components for this scheme. Finally, in section 3.6 and 3.7, we talk about the post-processing of association rules and the Interpretation of the results.

This chapter includes several ideas which are all about explaining the core contribution of the research beside the base of this research. Moreover, this chapter provides all the sequence steps that have been followed in order to satisfy the research objectives.

- **The Research Design**

The research design is the plan, structure, and strategy of investigation conceived so as to obtain answers to research questions and to variance with economy in procedure. It is the conceptual structure within research is conducted and it constitutes the blueprint for the collection, measurement and analysis of data. As such, the design includes an outline of what the researcher will do from setting the objectives and its operational implications to the final analysis of data. Descriptive research design has been used in this study, which is consist of the process of collecting data, analysis it, presents its results, and presents the conclusion of this analysis.

- **The Purpose of Research**

The main objective of the research is to develop and propose a new scheme for mining the association rules out of transactional data set. The proposed scheme is based on two approaches: nonordfp approach and Apriori rule generation approach. The proposed scheme is more efficient than Apriori algorithm and FP-growth algorithm, as it is based on two of the most efficient approaches. To achieve the research objective successfully, a series of sequence progresses and analysis steps have been adopted. Figure 3-1 depicts the methodologies to extract the association rules from the transactional data set using the new scheme.

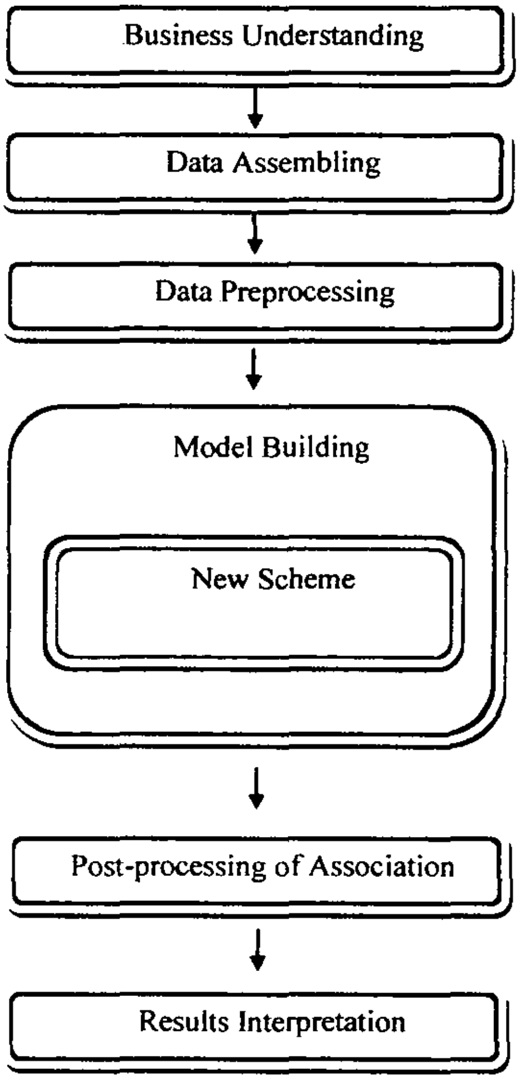


Figure 3-1. Methodologies to extract the association rules

Figure 3-1 illustrates different phases to extract the association rules from the transactional database using our scheme, starting with business understanding, and continues throughout data Assembling, data preprocessing, model building (using the new scheme), post-processing of association rules, and results interpretation. This new scheme will be used to generate the association rules in a real business case study.

3. 2 Business Understanding

The concern of this stage is to identify the problem area and describe the problem in general terms. In another word, the enterprise decision makers need to formulate goals that the data mining process is expected to achieve. Then the first step in the methodology is to clearly defined business problem. The business analyst specifies the problem in the business.

3. 3 Data Assembling

Data mining required access to data. The data may be represented as volumes of records in several database files or the data may contain only few hundred records in a single file. In order to build effective model a data mining algorithm must be presented with thousands or millions of instances. Then the second step in our methodology is to collect the data which could come from many resources (OLAP, Data warehouse, relational database, and flat file). For instance, in the case study which is presented in chapter four the data was collected for six months (1 December 2007 until 31 May 2008) from relation database.

3. 4 Data Preprocessing

Applying data preprocessing techniques before mining, can substantially enhance the overall quality of the patterns mined and/or the time required for the actual mining, Low-quality data will lead to low-quality mining results. The preparation of data set is one of the most critical steps in a data mining process. This stage is concerned with selecting data, and mapping it.

Selecting data refers for removing unnecessary information. When the data is drawn from different sources, it is possible that the same information is represented in different sources in different format. For instance, in the case study the recorded transactions are all the transactions made by someone holding one of the loyalty cards. Each card carries a code for identifies feature about the owner, including important personal characteristics such as sex, birth date, number of children, profession and education. Our aim is to select only the transactions data on products (total amount of transactions being collected equal 88162, total number of products available in the shop is 16,470 items and average transaction size is 13 items).

Mapping the data, it is the process of transfer the values of the selected variables. In our scheme we are going to deal with numeric attribute. Using the numeric attributes will reduce the consumption of the memory. Therefore we need to map or to eliminate the nominal attributes from the dataset. Consequently, this stage reconfigures the data to ensure consistent format, as there is possibility of inconsistent formats.

3. 5 Model Building

This stage is concerned with extraction of patterns for the data. The core of this research is mainly focused on model building. This phase concerns various view points and different aspects that should be given attention in order to yield sufficient results.

It starts with examine most of the existing frequent itemset algorithms, to study the performance of each algorithm on the same set of data and determine the best algorithm based on those comparisons in (section 3.5.1.1 and section 3.5.1.2) we found that nonordfp algorithm outperforms the other algorithms if we compare for both the time and the memory consumption. The other part of our new scheme is Apriori rule generation. A distinct feature of this scheme is that the generation of the association rules is efficient, another important feature is that no need to read the data base more than two times.

Those two approaches, Nonordfp approach and Apriori rule generation approach, were employed by our scheme to pull the association rules out of the transactional database. Testing this scheme will be taken place in the real business case study as is shown in chapter four. And performance evaluation of this scheme is presented in chapter five.

3.5.1 Efficient and Scalable Frequent Itemset Mining Methods

There are several algorithms for mining the frequent itemsets. Those algorithms can be classified and Apriori-like algorithms (candidate generate-and-test strategy) and FP-growth-like algorithms (divide-and-conquer strategy). In this research we are focusing on FP-growth-like algorithms to find the frequent itemsets. Many variations of the FP-growth algorithm have been proposed which focus on improving the efficiency of the original algorithm. To investigate which of these algorithms (namely, AFOPT algorithm, Nonordfp algorithm and Fpgrowth* algorithm) is the best for our scheme, we carried out many of experiments by using different existing data sets.

Data set: The data is challenging due to the number of characteristics which are the number of the records, and the sparseness of the data (each records contains only small portion of items). In our experiments we chose different dataset with different properties, to prove the efficiency of the algorithms, Table 3-1 shows the datasets and the characteristics.

Table 3-1. The Datasets

Data set	#Items	Avg. Length	#Trans	Type	Size
T10I4D100K	1000	10	100, 000	Sparse	3. 93 MB
T40I10D100K	1000	40	100, 000	Sparse	14. 8 MB
Mushroom	119	23	8, 124	Dense	557 KB
Connect4	150	43	67,557	Dense	8. 89 MB

Table 3-2. Meaning of the synthetic dataset parameters

T	Average length of the transaction
I	Average size of frequent itemsets
D	Number of transactions
K	Thousand

Mushroom and connect4 data sets are real data. While T10I4D100k and T40I10D100K are synthetic data set Table 3-2 explains the symbols of the synthetic data. Mushroom and Connect4 are real-world data set that publicly available in the FIMI repository.

As shown in the table, the data sets have different properties (number of the items: as the number of the items increases, more space will be needed to store the support counts of items; the average length of the transaction: for dense data sets, the average length of the transaction can be very large, this affects on the complexity of the algorithm ; the number of the transactions in the data set: the run time increases with a large number of transactions, the sparseness of the data, the size of the data set), which have the great effect on the performance and efficiency of the algorithms.

- **Running Time Comparison**

The running time is real time, system time and user time. Figure 3-2, Figure 3-3, Figure 3-4, and Figure 3-5 depict the time needed in seconds for each one of the algorithms. Different support values (The minimum support values to be chosen depends a bit on the number of records in the data set) were taken to evaluate the important of the frequent itemsets obtained by the algorithm (the support is an important measure because the itemset that has very low support values may occur simply by chance. A low support itemset is also likely to be uninteresting from business perspective because it may not profitable to promote items that customers seldom purchase together. Another benefit of the support measure is that it has an attractive property (anti-monotone property) that can be exploited for the efficient discovery of association rules).

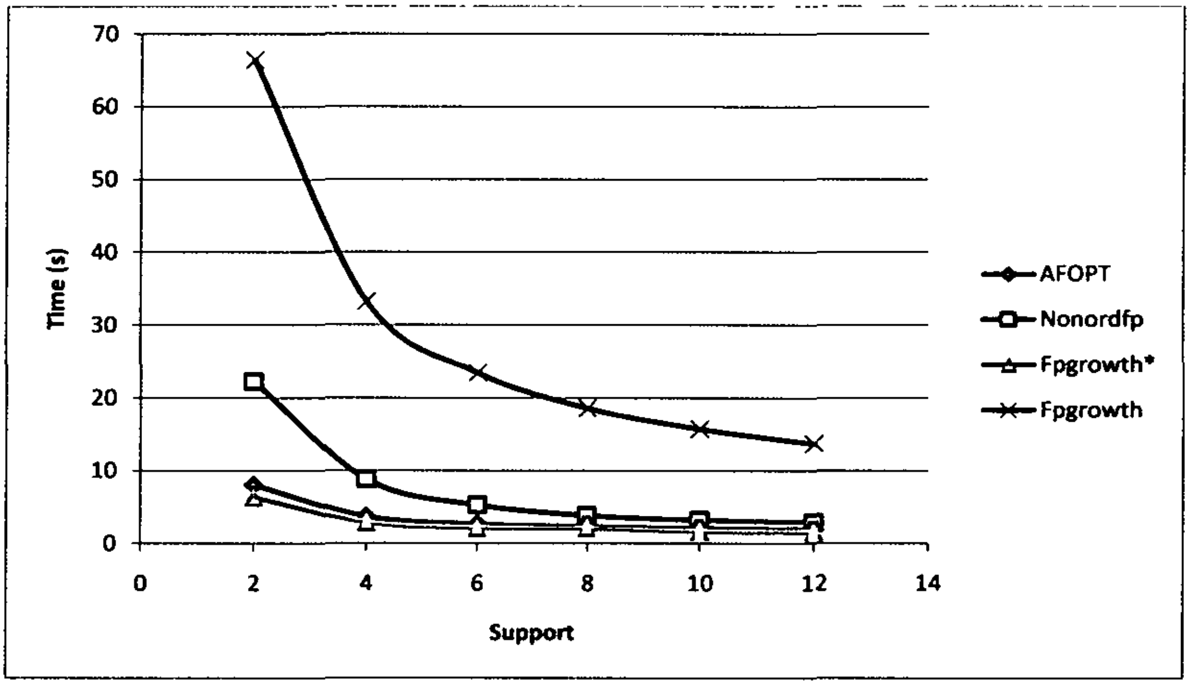


Figure 3-2. Execution time at various support levels on T10I4D100k

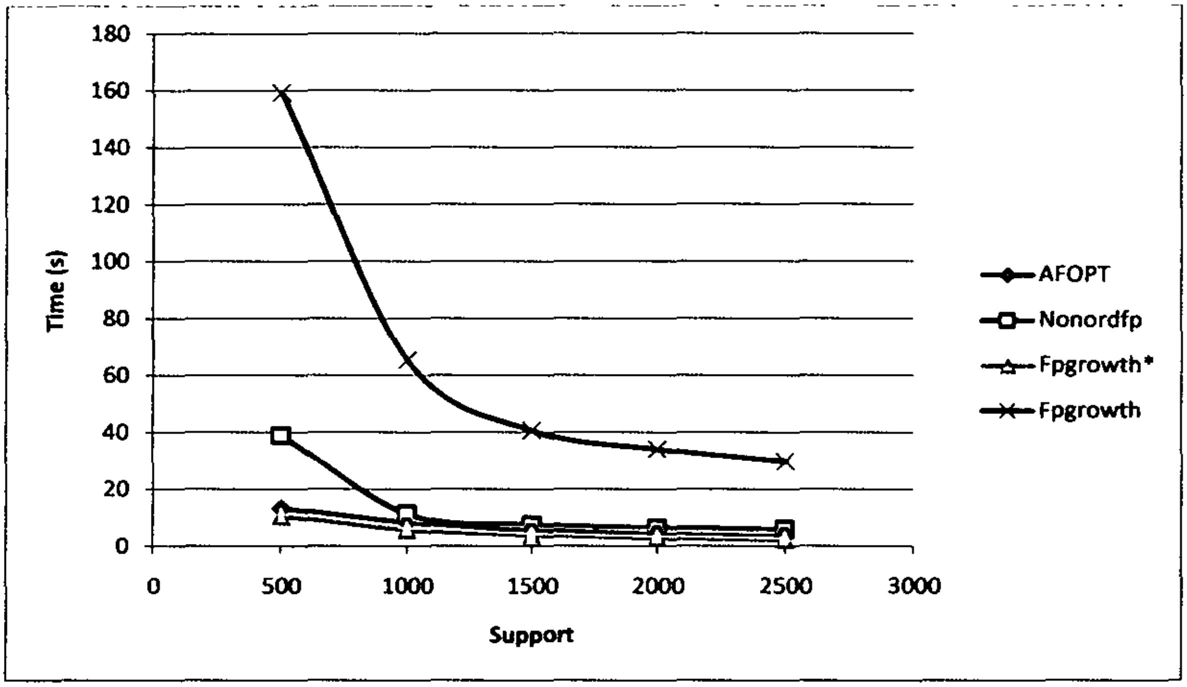


Figure 3-3. Execution time at various support levels on T40I10D100K

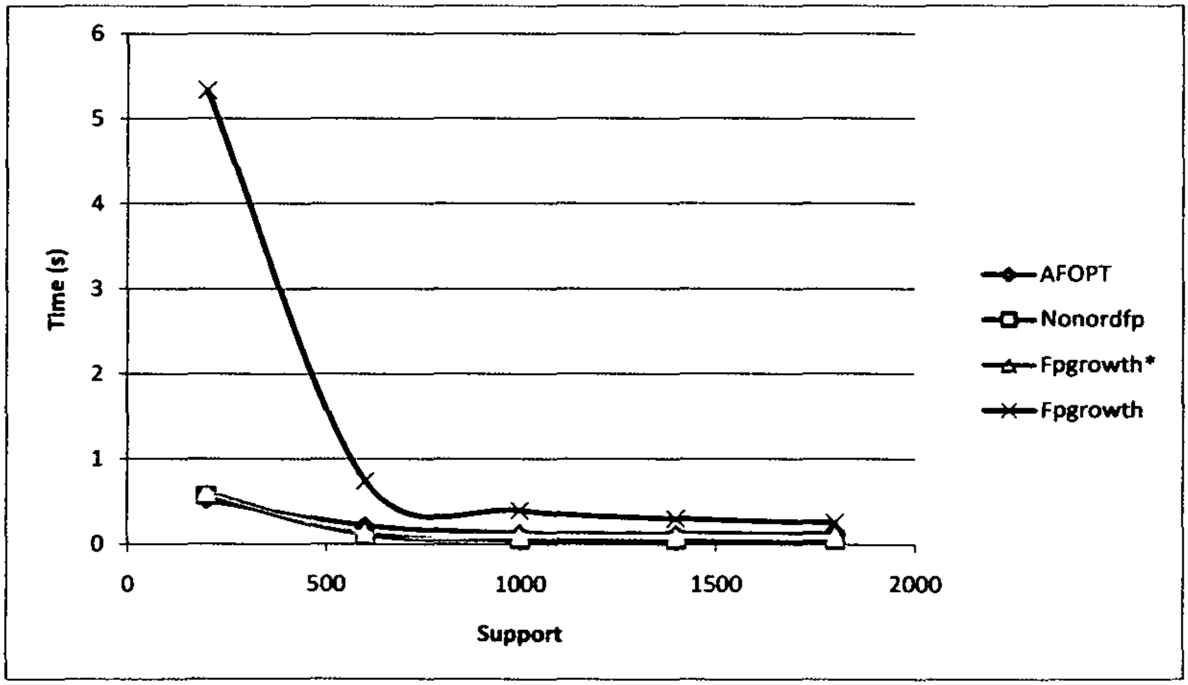


Figure 3-4. Execution time at various support levels on Mushroom

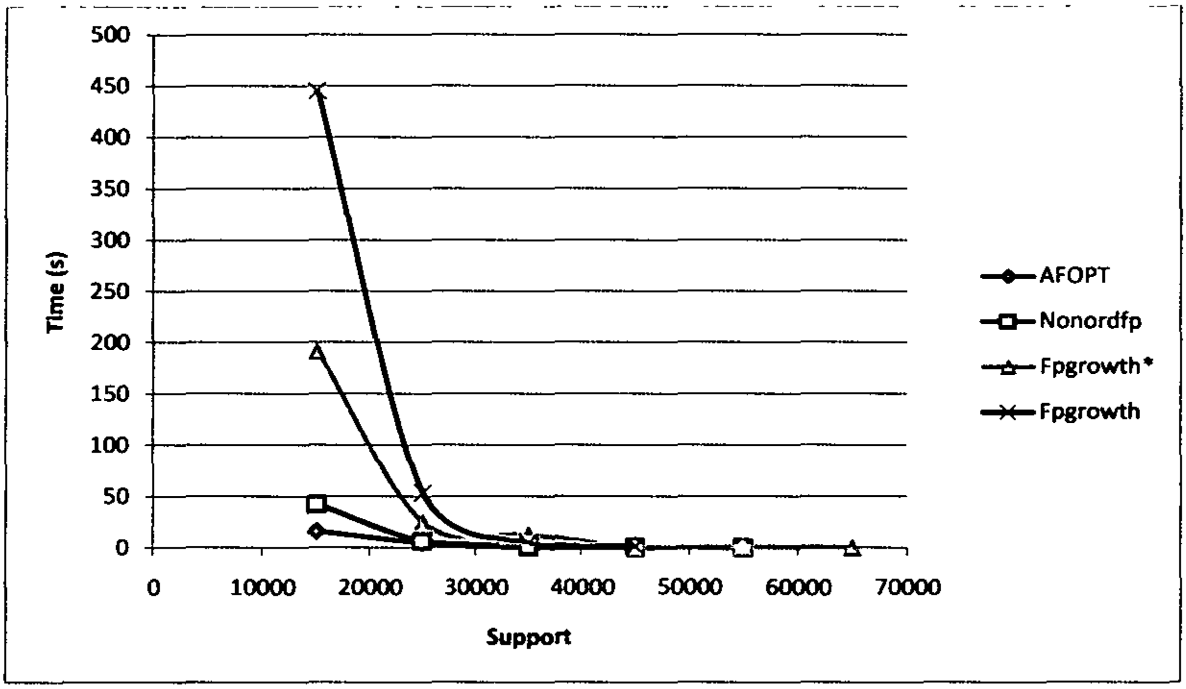


Figure 3-5. Execution time at various support levels on Connect4

It is clear that with the T10I4D100k data set Fpgrowth* algorithm outperforms all the other algorithms. On T40I10D100K data set there is obvious performance competition among both Fpgrowth* algorithm and AFOPT algorithm. The running times for the AFOPT algorithm, Nonordfp algorithm, and Fpgrowth* algorithm are near in the case of mushroom data set. For the connect4 data set, we should mention that some algorithms had problem, segmentation fault, with some values of support due to the huge number of the frequent itemsets satisfy those thresholds values and some took long time to find the frequent itemsets.

- **Memory Consumption Comparison**

In this section, we calculate the total number of memory consumption for each algorithm. All the experiments are done on the same sets of data. As demonstrated in Figure 3-6, Figure 3-7, Figure 3-8, and Figure 3-9, the support values and the amount of memory for each one. We observe that, Nonordfp algorithm remains stable over the whole range of support values on T10I4D100k. The stability in memory consumption is also obvious for Fpgrowth* algorithm and AFOPT algorithm for the high values of support.

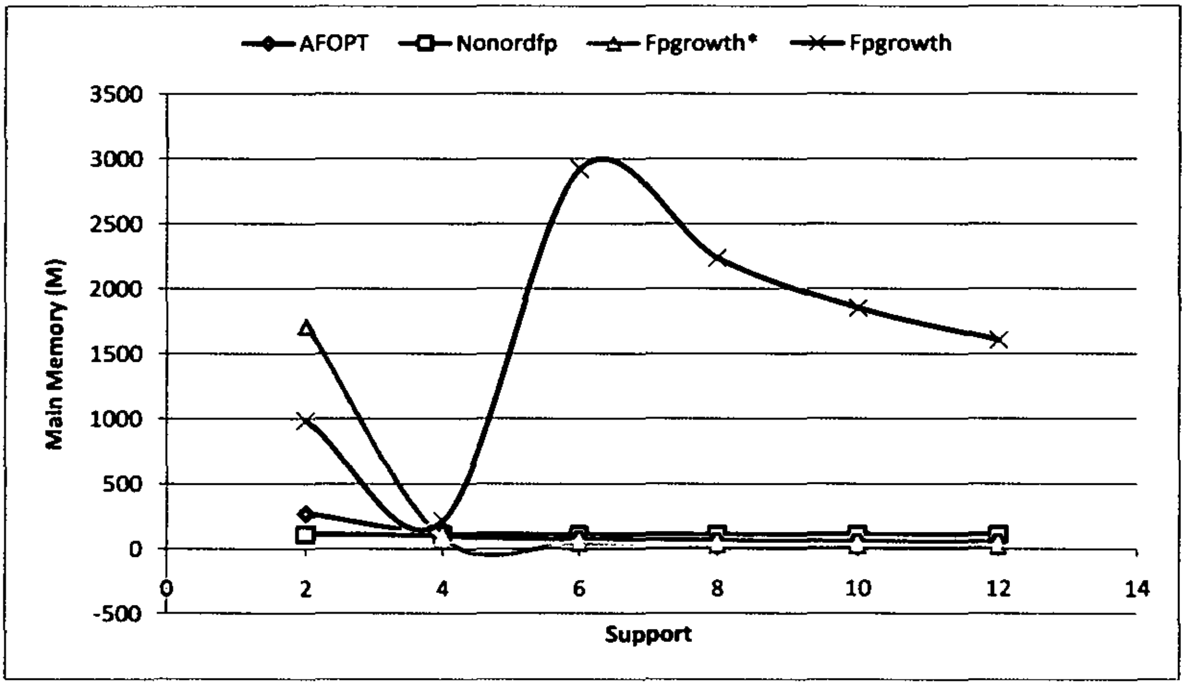


Figure 3-6. Memory usage on T10I4D100k

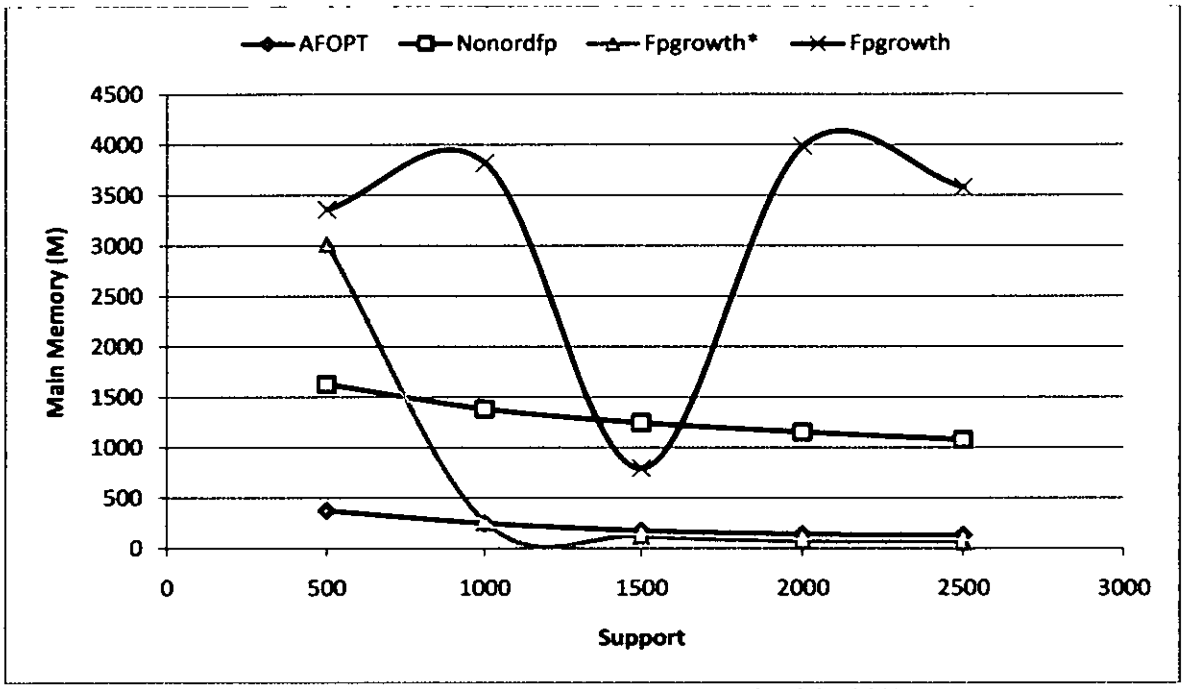


Figure 3-7. Memory usages on T40I10D100K

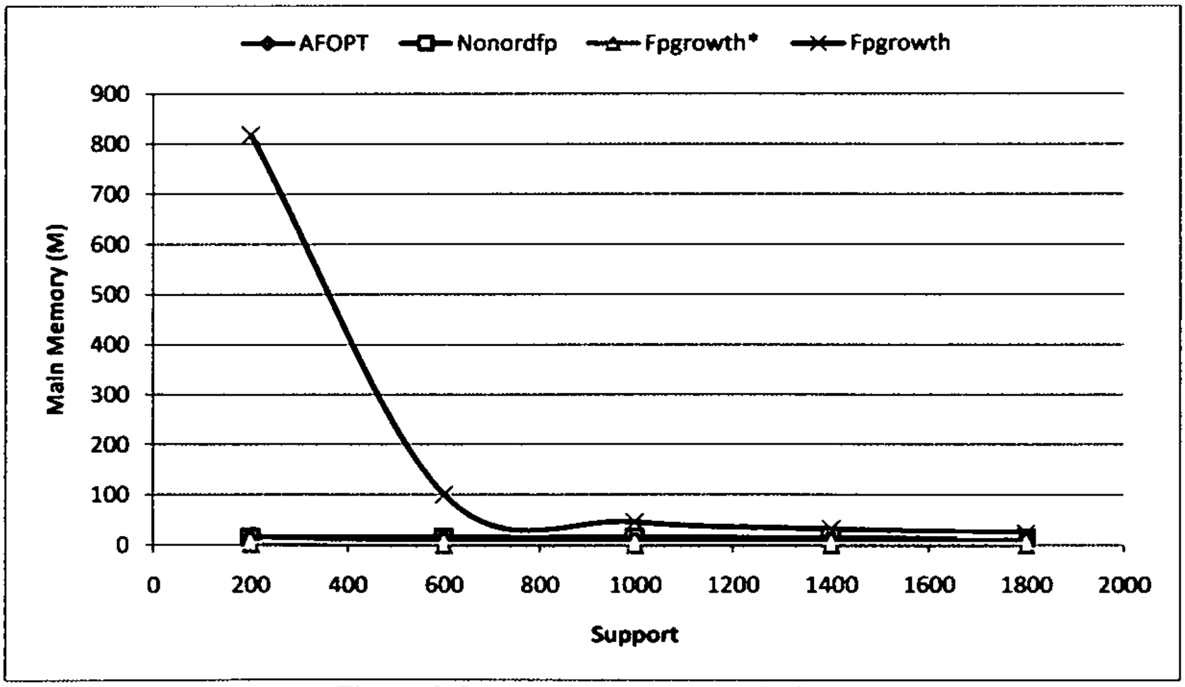


Figure 3-8. Memory usages on Mushroom

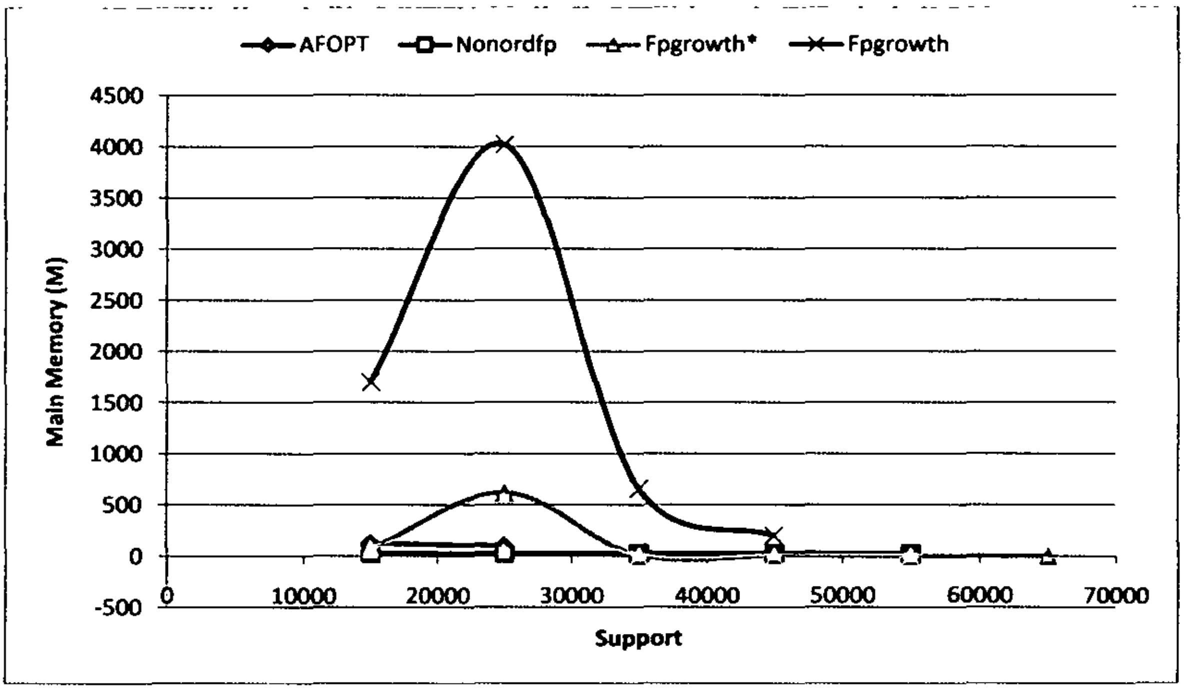


Figure 3-9. Memory usages on Connect4

The behavior of FP-growth algorithm is unstable for the consumption of the memory. The instability for the consumption happens due to the conditional databases construction. AFOPT algorithm keeps its stable consumption of the memory on T40I10D100K. It further confirmed the fact that AFOPT algorithm is stable with sparse data sets. In Figure 3-8 the competition between the three algorithms is clear, the memory usage is competitive. On Connect4 data set, Fpgrowth* shows stability in the case of the high support thresholds while Nonordfp algorithm remains stable for the low values of support. As conclusion for these comparisons we found that nonordfp algorithm outperforms the other algorithms if we compare for both the time and the memory consumption.

3.5.2 Rule Generation

The process of generating the rules is straight forward, because there is no need to read the database again. In our research we are interesting to mine association rules with single dimension, single level from frequent itemset in the transactional databases, which is based on the type of the available data. Depending on preliminary test of time and memory consumption we found that Apriori rule generation approach (Agrawal et al 1994) is stable and consume a little amount of time and memory during the process of rule generation. Figure 3-10 shows the running time by using different level of confidence for generating the association rules from the existing data sets.

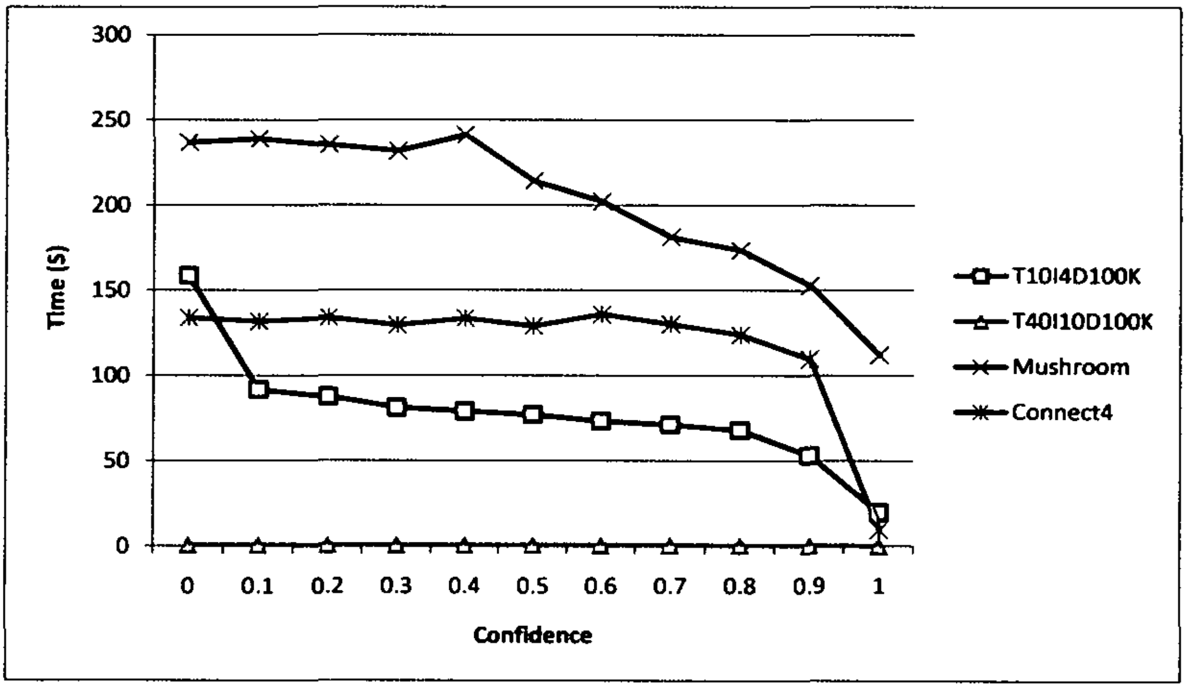


Figure 3-10. Running time of rule generation process

The behavior of the rule generation using the Apriori rules shows that the size of the data set has affect on the time. With the increase of the size the running time for generation rules will increase, this is because with the increasing of the size the time for processing will increase. Moving from the low level of confidence to high level the behavior is changed gradually, because with increasing the level of the confidence more rules are trimmed. The other processes running by the systems have affected on the inconsistent in the behavior of all of these data sets.

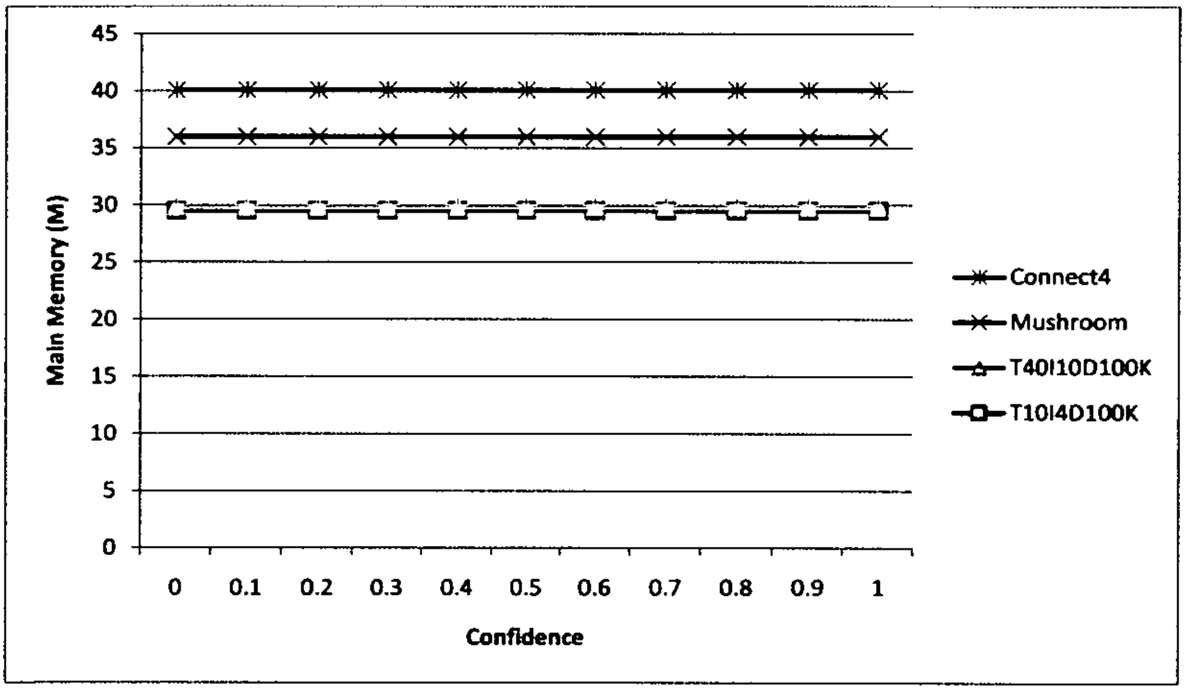


Figure 3-11. Memory consumption of rule generation process

Figure 3-11 depicts the amount of the memory utilization (the value of the heap peak) during the process of rule generation. It is clear that for all data sets the consumption of memory is stable, because for every itemset in the data set the algorithm finds the subsets of it without the need to read the data set.

3.5.3 Implementation of New Scheme

Basically, the new scheme integrates nonordfp approach and Apriori rule generation approach. Thus the main architecture of our scheme is:

Input of scheme:

- The set of the data (D)
- The support value (S)
- The confidence value(C)

Output of scheme:

- Set of association rules

```
Step 1:
  Compute the frequency of itemsets in D;
  Compute the support (sup) of itemset, //where sup= frequency (X∪Y)/|D|;
   If(sup(itemsets)>=S)
     {Generate the frequent itemsets (FI);}
Step 2:
  For each frequent itemsets (FI)
   {Find the subset itemsets x and y;}
Step 3:
  Compute the confidence (conf) of x===>y, //where conf=frequency (X∪Y)/frequency(X);
  For each x and y
    If(conf(x===>y) >=C)
    {  Compute the correlation (cor) of x===>y,
       /* where cor is one of three measurements

         • Correlation Coefficient $\frac{P(xy)-P(x)P(y)}{\sqrt{P(x)P(y)P(x-1)P(y-1)}}$
         • Cosine =$\frac{P(xy)}{\sqrt{P(x)P(y)}}$
         • Interest=$\frac{P(xy)}{P(x)P(y)}$

       */
      Output x===>y {sup(x,y) , conf(x,y),cor(x,y)};
    }
```

3. 6 Post-processing of Association Rules

This stage depends on the result of the previous ones. Actually it is about measure the interestingness of the items in the obtained model or pattern. Most likely the most significant problem with association rules, which so far remains largely unsettled, is the 'interestingness' of association rules. Indeed, the main strength of association rule mining is that, since it discovers all association rules that exist in a database, it can reveal valuable and unexpected information. These strengths, however, are also its weakness; i.e. the number of discovered rules can be huge, hundreds or even thousands of rules, which makes manual examination of those rules practically infeasible. In other words, association rule results sometimes create a new data mining problem of the second order. This makes post-processing of these rules very significant, i.e. we need good methods to reduce the number of association rules to the most interesting ones. The reasons for this problem of interestingness can be found in the limitations of the support-confidence framework, adopted by almost all.

After learning system induces models from the data their evaluation should take place, there are several measurement for this purpose, support, confidence, correlation, cosine, interest (see section 2.10). All objective measure of interestingness for association rules is based on the statistical notion of correlation between the items in the antecedent and the consequent of the rule. The idea is to construct a contingency table from the association rule results and test the interdependence between the antecedent and the consequent of the rule.

We utilize three popular objective measurements which are correlation, cosine, and interest (Tan et al 2006) to see which is the most suitable for our data set. Therefore, a good strategy is to perform the correlation coefficient analysis first, and when the result shows that they are weakly positively/negatively correlated, other analyses can be performed to assist in obtaining a more complete picture.

3. 7 Interpretation and Explanation of the Results

The patterns obtained in the stage of model building and refined in the stage of post-processing of association rules are converted into knowledge, which in turn, is used to support the decision-making.

After the rules are mined out of the database, the rules are used to understand the problem better. To summarize the obtain rules there are four methods which are: by ranking the rules by their supports value help the decision maker to know what the most ubiquitous rules are. By ranking the rules by their confidence value highly confidence value imply strong relationships. By considering the correlation to obtain a relative measure of a rule's interestingness. By summarizing all the rules that have certain value for consequent; it can be used to understand what is the associated with the consequent and perhaps what affects the consequent.

Now we may use the acquired knowledge directly for predication or in an expert system shell as a knowledge base. If the knowledge discovery process is performed for an end-user, we usually document the derived results. Another possibility is to visualize the knowledge, or to transform it to an understandable form for the user-end. In this stage the main concern is to summarize the obtain rules and present them to the decision makers.

3. 8 Summary

In this chapter, we proposed new scheme for extracting association rules from transactional datasets. The methodology for analyzing the transactional data by using our new scheme consists of six phases. First step of this methodology is to specify the problem which is going to be solved by analyzing the data. The next step is to assemble the data sets into one data set to be used for finding the patterns in the data set. The next step of our methodology is to map the data to numeric values to make it works with the new scheme, to reduce the amount of the memory taken by the nominal values. The scheme employees the nonordfp approach, which has been chosen according to its time and memory efficiency. Nonordfp approach is used in this scheme to find the frequent itemset. Once the frequent itemsets from transactions in a data set have been found, it is straightforward to generate all frequent and confidence association rules from them. Level-wise approach is utilized by the scheme to generate the association rules. Next step is to evaluate the obtained rules by the correlation measurements to eliminate the un-interestingness rules. Final step is to summarize the obtain rules to support the decision-making, the summarization can be by sorting the obtained rules depending on the values of its support, confidence, correlation ,and unified consequent of the rules .

CHAPTER FOUR: MARKET BASKET ANALYSIS CASE STUDY

The retail industry is a most important application area for data mining, since it collects enormous amounts of data on sales, customer shopping history, service, and goods transportation, consumption. Progress in bar code technology has made it possible for retail organizations to collect and store massive amounts of sales data, referred as the basket data. Such market basket databases consist of a large number of transaction records. Each record lists all items bought by a customer on a single purchase trip. Using market basket analysis is a key factor of success in the competition of supermarket retailers. Market basket analysis provides manager with knowledge of customers and their purchasing behaviour which brings potentially huge added value for their business. Recent marketing research has suggested that in-store enviromental stimuli, such as shelf-space allocation, and product display, have a great influence upon consumer buying behavior and may induce substantial demand. This chapter presents a market basket analysis case study, to verify the efficiency of our new scheme.

4. 1 Objective of the Market Basket Analysis

Retailing is an industry with high level of competition. It is a customer based industry which depends on how it could be aware of what the customers' needs and requirements are. The very first groceries displayed their products in an industrial approach which have produced the present day grocery store layouts based on "sectors" as fruits, vegetables, magazines, CDs, and so on. This approach is a company oriented and it fails to respond to the needs of the time-pressured consumer. In the new market, satisfying the customer needs is one of the important tasks for the retailers. This required from company to move from the traditional company-oriented to customer-oriented which concern about the buying behavior of the customer.

The store layout and the promotional campaign are huge tasks for retail managers. The complexity of these tasks lies on the relationships between categories on sale as well as on the impact that it produces on the consumer spatial behavior and in-store traffic.

One possibility to do so is to make the store layout construction and the promotional campaign through the introduction of market basket analysis. Market basket analysis has the objective of individuating products, or groups of products, that tend to occur together (are associated) in buying transactions (baskets). The knowledge obtained from a market basket analysis can be very valuable, and it can be employed by a supermarket to redesign the layout of the store to increase the profit through placing interdependencies products near to each other and to satisfy customers through saving time and personalized the store layout. Another strategy, Items that are associated can be put near to each other; it increases the sales of other items due to complementarily effects. If the customers see them, it has higher probability that they will purchase them together.

The knowledge obtained from a market basket analysis can be also used to improve the efficiency of a promotional campaign: products that are associated should not be put on promotion for the same periods. By promoting just one of the association products, it should be possible to increase the sales of that product and accompanying sales increases for the associated products.

From a marketing perspective, the research is motivated by the fact that some recent trends in retailing pose important challenges to retailers in order to stay competitive. Indeed, the rise of large retail stores and the fact that customers are getting used to self-service resulted in a loss of personalized customer service and creates new challenges to gain and keep customer loyalty, for instance through personalization.

Indeed, as a result of the trend for one-stop-shopping, consumers typically make interdependent purchases in multiple product categories and failing to consider those interdependencies may lead to marketing actions with disappointing results.

The database usually considered in a market basket analysis consists of all the transactions made in a certain sale period and in certain sale location. Consumers can appear more than once in the database. In fact, consumers will appear in the database whenever they carry out a transaction at a sales location. A number of recent techniques in data mining (association rules) provide excellent opportunities to take such interdependencies among products into account. This case study applies data mining method, namely association rules mining to understand the association between buying behaviors.

4. 2 Description of the Data

For the purpose of this study, an empirical data set is kindly provided from anonymous retail supermarket store. The data collected over six months, started by the first of December 2007 until the end of May 2008. The considered period of the data consists of approximately six months.

The recorded transactions are all the transactions made by someone holding one of the loyalty cards. Each card carries a code for identifies feature about the owner, including important personal characteristics such as sex, birth date, number of children, profession and education. The card allows the analyst to follow the buying behavior of its owner: how many times they go to the supermarket in the given period, what they buy, whether they follow the promotions, etc.

The total amount of receipts being collected equals 88162. The total number of products available in the shop is 16,470 items and the average transaction size is 13 items.

In total, 5,133 customers have purchased at least one product in the supermarket during the data collection period. The second step in our methodology is to assemble the data which we have in the format of a relational database. We first converted the data into horizontal database layout <TID, {item1, item 2,... , item n}>, to be suited for our scheme. Table 4-1 and Table 4-2 contain the description of the database tables used. We used the concatenation of the fields TARH, KAS_NO, FS_NO as the TID of the itemset. Our aim here is to consider only transactions data on products, in order to investigate the association between these products.

Table 4-1. Table Fs_Baslik used in the datasets

Field name	Field Description
TARH	Date of transaction
KAS_NO	ID of the cashier
FS_NO	ID of the receipt
MUSTERI_NO	ID of the customer

Table 4-2. Table Fs_Detay used in the datasets

Field Name	Field Description
TARH	Date of transaction
KAS_NO	ID of the cashier
FS_NO	ID of the receipt
FS_SIRANO	The place of the item in the receipt
MALA_NO	ID of the item
MIKTER	The amount of the item

Figure 4-1 shows the average number of different items bought per shopping visit. The average number of different products bought per shopping visit equals 13 and most customers buy between 7 and 11 items per shopping visit.

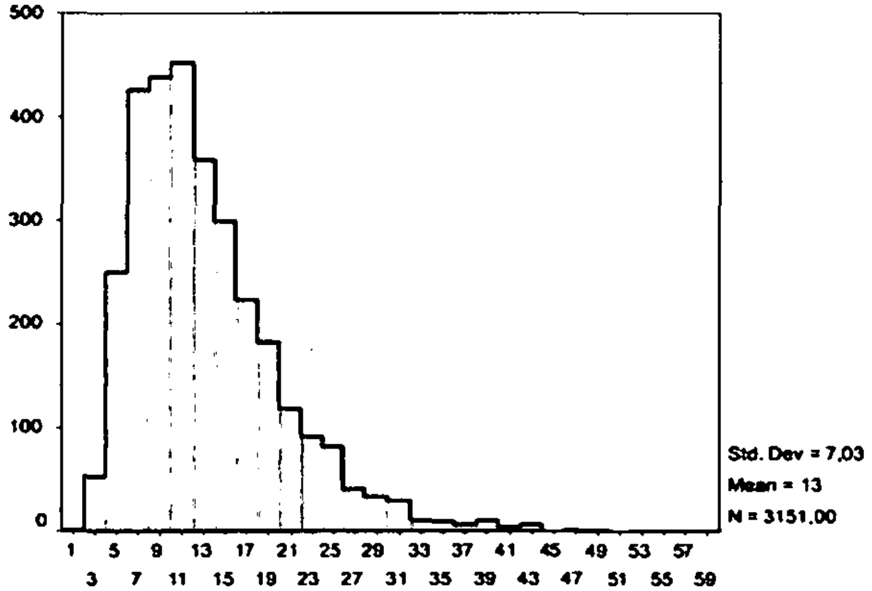


Figure 4-1. Average number of different items bought per visit

In data mining, a key problem that arises in any of masse collection of data is confidentiality issues. The need for privacy can be motivated by business interests. The data we are using in this thesis is confidential therefore it is mined blindly, i.e. not knowing what we are mining. Before the data can be used for market basket analysis, mapping the data is crucial, that because, the numeric attributes are used by our scheme. Therefore, we need to map or to eliminate the nominal attributes from the dataset. Table 4-3 shows the overall position associated with the name of the product.

Table 4-3. A sample of the mapped data set

Overall Position	**Description**	**Category**
1	Coca Cola Regular	Soft drinks
2	Dessert vacuum Douwe Eghberts	Coffee
4	Water still Spa	Water
6	Dash Scoops	Washing powder
12	Sandwiches	Fresh sandwiches
16	Red port	Appetizer drinks
21	Mayonaise egg D. L.	Mayonaise
23	Bo French bread	Bake-off products
25	Fresh eggs	Eggs
27	Yakult	Milk
28	Multi-grain bread	Bread
58	Dreft household liquid	Dish washing
70	Calgon	Cleaning products
86	Salty crisps Smiths	Crisps
90	Double Lait chocolate C. d'Or	Chocolate
100	Fruit basket	Fruit
102	Baby tissues pampers	Baby care
118	Daycreme nivea	Beauty
127	Batteries cigarette 1. 5 V Duracell	Electricity
130	frying oil VDM	oils
165	Herring fillets Korenbloem	Refrigerated salads
165	Herring fillets Korenbloem	Refrigerated salads
167	Effi Minarine	Margarine spread
174	Toilet paper Scottex	Toilet paper
303	Backerbsen	Flour products
637	Deo spray Dove	Deodorant
919	Whiskas cocktail	Cat food
...	...	...

4. 3 Model Building

The most common way to analyze of market basket data is to use association rules, a local data mining method. This local models (patterns) look at selected parts of the data set (subset of variables or subset of observations). In our scheme there are two steps for obtain the association rules

4. 3. 1 Find the Large Itemsets

For finding the frequent itemsets in our scheme we are using the nonordfp approach on the prepared data.

We chose the minimum support 0.00187 to be threshold for the itemsets which is equal to absolute support count 164, choosing the right support threshold for mining the data set is quite tricky. In our analysis we are using small value for two reasons, the first we do not want to lose interesting patterns even if it is happened seldom, and the second reason, the computational efficiency of the chosen algorithm allow us to use this low value of support. This resulted in 3029 frequent itemset sets of size [1] to [5], see table 4-4. The table shows the majority of the frequent itemsets are of size 1 and 2. The amount of the time needed to generate these frequent itemsets is equal 0. 684 seconds.

Table 4-4. The number of the frequent itemset sets for different sizes

The size of the frequent itemset	The number of the frequent itemsets
1-itemsets	1041
2-itemsets	1285
3-itemsets	586
4-itemsets	110
5-itemsets	7

Tables 4-5, 4-6, 4-7, 4-8, 4-9 show portions of the frequent itemsets. The support can be used to assess the importance of a rule in terms of its frequency in the database. The highest frequency found from this data set:

In the 1-itemsets is 50675 for the itemset {39}, the whole output of frequent 1-itemsets is shown in appendix 1. Note that the minimum frequency in the tables is 164 which is the pre-defined threshold.

In the case of 2-itemsets the highest frequency is 29142 for the itemset {39, 48}, the whole output of frequent 2- itemsets is shown in appendix 2.

For the 3-itemset, the highest frequency is 7366 for the itemset {39, 48, 41}, the whole output of frequent 3- itemsets is shown in appendix 3.

For the 4-itemsets, the highest frequency is 1991 for the itemset {38, 41, 48, 39}, the whole output of frequent 4- itemsets is shown in appendix 4.

Finally, in the case of 5-itemsets, the highest frequency is 448 for the itemset {32, 38, 41, 48, 39}.

Table 4-5. The frequent 1-itemsets

Itemsets (frequency)	Itemsets (frequency)	Itemsets (frequency)
892 (164)	848 (170)	5289 (176)
991 (164)	2182 (170)	0 (177)
2327 (164)	2876 (170)	2202 (177)
2853 (164)	3762 (170)	2471 (177)
3338 (164)	4552 (170)	3402 (177)
3420 (164)	11560 (170)	4113 (177)
3539 (164)	12491 (170)	1290 (178)
4389 (164)	958 (171)	1805 (178)
4886 (164)	2026 (171)	1905 (178)
58 (165)	2720 (171)	2493 (178)
1439 (165)	5346 (171)	3312 (178)
1568 (165)	10598 (171)	4143 (178)
2084 (165)	1612 (172)	4221 (178)
2269 (165)	2424 (172)	5323 (178)
2721 (165)	4424 (172)	6403 (178)
3297 (165)	444 (173)	368 (179)
4340 (165)	986 (173)	572 (179)
6404 (165)	1781 (173)	1166 (179)
12996 (165)	2129 (173)	2822 (179)
265 (166)	2514 (173)	4643 (179)
460 (166)	3271 (173)	10128 (179)
770 (166)	4771 (173)	1615 (180)
1981 (166)	5178 (173)	2708 (180)
4080 (166)	364 (174)	2840 (180)
4226 (166)	445 (174)	3240 (180)
10441 (166)	683 (174)	4859 (180)
12921 (166)	1178 (174)	10481 (180)
1130 (167)	1215 (174)	228 (181)
1180 (167)	1727 (174)	289 (181)
1354 (167)	7646 (174)	1184 (181)
2122 (167)	467 (175)	1588 (181)
2919 (167)	791 (175)	2800 (181)
3742 (167)	877 (175)	2812 (181)
4548 (167)	1896 (175)	3988 (181)
3272 (168)	1951 (175)	4913 (181)
3314 (168)	4310 (175)	13556 (181)
3840 (168)	236 (176)	184 (182)
2408 (169)	1736 (176)	781 (182)
2820 (169)	2210 (176)	3161 (182)
2967 (169)	2654 (176)	4342 (182)
3973 (169)	3236 (176)	4744 (182)
5339 (169)	4843 (176)	421 (183)

Table 4-6. The frequent 2-itemsets

Itemsets (frequency)	Itemsets (frequency)	Itemsets (frequency)
3005 38 (195)	414 39 (188)	2573 39 (178)
4975 39 (164)	673 39 (172)	4945 48 (177)
2805 38 (212)	3012 39 (165)	10420 48 (178)
2113 39 (171)	347 48 (177)	10420 39 (183)
1996 39 (174)	1629 (264)	128 48 (170)
10074 39 (165)	1629 48 (165)	169 39 (177)
887 48 (164)	1786 (264)	250 48 (178)
1476 39 (168)	1786 48 (168)	1188 38 (256)
3311 1819 (180)	1786 39 (206)	1188 39 (176)
491 48 (182)	2763 39 (164)	684 48 (177)
491 39 (186)	3635 48 (166)	684 39 (168)
708 39 (178)	3635 39 (168)	1291 39 (170)
1104 39 (167)	1183 48 (179)	1294 48 (166)
1591 39 (169)	1938 39 (169)	1294 39 (167)
767 48 (164)	116 39 (166)	4070 48 (175)
2998 48 (169)	1010 48 (165)	4070 39 (177)
2998 39 (168)	1010 39 (193)	1269 39 (194)
256 39 (188)	2167 48 (181)	3279 39 (181)
597 39 (209)	2167 39 (181)	611 39 (165)
795 1819 (167)	285 48 (185)	764 39 (194)
1804 39 (164)	2856 48 (176)	2344 48 (180)
924 48 (164)	2856 39 (170)	2344 39 (183)
925 39 (165)	12473 48 (166)	1976 48 (177)
1143 39 (178)	12473 39 (172)	1976 39 (184)
5728 39 (191)	1602 48 (165)	2625 48 (166)
576 39 (170)	1602 39 (176)	1280 48 (165)
167 48 (164)	1616 39 (169)	1280 39 (169)
1026 39 (168)	4685 48 (178)	1379 309 (166)
1163 48 (177)	4685 39 (166)	1379 39 (196)
1966 39 (184)	10605 48 (179)	1564 48 (182)
1899 39 (174)	10605 39 (191)	234 39 (185)
4307 48 (168)	95 39 (189)	412 48 (181)
489 48 (165)	2053 48 (176)	412 39 (188)
492 39 (164)	1232 48 (179)	2635 39 (178)
552 48 (174)	5124 48 (170)	2965 48 (179)
757 39 (184)	5248 39 (175)	2965 39 (179)
1831 38 (252)	2997 39 (164)	10551 48 (168)
2259 39 (175)	165 48 (195)	10551 39 (178)
2915 39 (164)	615 48 (170)	662 39 (169)
4344 39 (184)	805 48 (171)	3668 48 (180)
536 48 (164)	805 39 (193)	3668 39 (174)
932 39 (166)	3315 39 (175)	82 48 (177)
308 39 (170)	1741 39 (194)	82 39 (165)

Table 4-7. The frequent 3-itemsets

Itemsets (frequency)	Itemsets (frequency)	Itemsets (frequency)
	2775 48 39 (167)	281 38 39 (274)
1590 38 39 (193)	1066 48 39 (169)	281 48 39 (193)
504 38 39 (170)	2673 48 39 (190)	831 48 39 (237)
2792 48 39 (175)	248 48 39 (200)	1659 48 39 (194)
1529 38 39 (202)	415 48 39 (179)	2051 48 39 (222)
1638 48 39 (164)	8985 48 39 (168)	2135 48 39 (228)
3904 38 48 (193)	1557 48 39 (166)	2184 48 39 (210)
3904 38 39 (268)	10656 48 39 (167)	80 48 39 (208)
3904 48 39 (179)	10491 48 39 (176)	1020 48 39 (190)
1027 48 39 (165)	10653 48 39 (169)	432 48 39 (190)
16431 16430 48 (174)	1543 48 39 (189)	136 48 39 (219)
16431 16430 39 (186)	2056 48 39 (180)	1867 48 39 (187)
640 48 39 (169)	1277 48 39 (166)	1677 48 39 (211)
1046 32 48 (171)	2241 48 39 (214)	1479 48 39 (197)
734 48 39 (229)	3799 48 39 (206)	10444 48 39 (221)
730 48 39 (167)	1513 48 39 (196)	178 48 39(194)
8691 48 39 (213)	1000 48 39 (164)	408 48 39 (198)
978 48 39 (164)	1486 48 39 (181)	14933 48 39 (205)
1619 48 39 (174)	1121 48 39 (168)	418 48 39 (200)
370 38 48 (191)	1126 48 39 (165)	2080 48 39 (187)
370 38 39 (241)	2523 48 39 (187)	2399 48 39 (194)
2115 48 39 (168)	514 48 39 (201)	2879 48 39 (199)
390 38 48 (197)	547 48 39 (171)	808 48 39 (181)
390 38 39 (241)	830 48 39 (196)	1987 48 39 (217)
390 48 39 (165)	9501 48 39 (170)	2329 48 39 (214)
2015 48 39 (178)	2987 48 39 (186)	345 48 39 (212)
3502 48 39 (164)	591 48 39 (210)	793 48 39 (218)
715 48 39 (167)	1062 48 39 (168)	856 48 39 (192)
10490 48 39 (176)	2505 48 39 (201)	2168 48 39 (205)
727 48 39 (184)	319 48 39 (194)	52 48 39 (239)
1257 48 39 (193)	384 48 39 (175)	798 48 39 (230)
2353 48 39 (166)	13189 48 39 (173)	1714 48 39 (193)
1003 48 39 (182)	979 48 39 (222)	426 48 39 (250)
12935 48 39 (214)	365 48 39 (197)	2284 48 39 (226)
840 38 48 (202)	47 38 48 (269)	53 48 39 (253)
840 38 39 (245)	47 38 39 (313)	1355 48 39 (194)
1313 48 39 (179)	47 48 39 (216)	571 48 39 (240)
12951 48 39 (183)	490 48 39 (193)	398 48 39 (224)
246 48 39 (182)	1404 48 39 (202)	94 48 39 (209)
4698 48 39 (187)	1113 48 39 (171)	855 48 39 (228)
1103 48 39 (189)	1585 48 39 (191)	910 48 39 (222)
1481 48 39 (188)	1704 48 39 (180)	4883 48 39 (200)
14099 48 39 (172)	281 38 48 (259)	261 48 39 (244)

Table 4-8. The frequent 4-itemsets

Itemsets (frequency)	Itemsets (frequency)	Itemsets (frequency)
3904 38 48 39 (173)	824 41 48 39 (172)	147 38 48 39 (176)
47 38 48 39 (212)	592 41 48 39 (170)	1327 41 48 39 (260)
281 38 48 39 (184)	16010 41 48 39 (261)	1327 38 48 39 (170)
790 38 48 39 (235)	9 41 48 39 (191)	438 41 48 39 (265)
56 38 48 39 (219)	185 41 48 39 (168)	438 32 48 39 (177)
8978 41 48 39 (172)	1146 41 48 39 (260)	438 38 48 39 (164)
105 38 48 39 (290)	255 41 48 39 (230)	413 41 48 39 (228)
16011 16010 41 39 (190)	255 32 48 39 (189)	413 38 48 39 (170)
16011 16010 48 39 (269)	255 38 48 39 (170)	271 225 48 39 (183)
55 38 48 39 (243)	533 41 48 39 (215)	271 41 48 39 (272)
371 38 48 39 (294)	79 41 48 39 (270)	271 32 48 39 (188)
175 41 48 39 (166)	79 32 48 39 (174)	271 38 48 39 (188)
13041 41 48 39 (219)	2238 225 48 39 (196)	475 41 48 39 (372)
37 41 38 48 (172)	2238 41 48 39 (322)	475 32 48 39 (234)
37 41 38 39 (202)	2238 38 48 39 (203)	475 38 48 39 (243)
37 38 48 39 (433)	270 271 41 39 (179)	101 41 48 39 (264)
15832 41 48 39 (220)	270 271 48 39 (284)	101 32 48 39 (182)
16217 41 48 39 (234)	270 41 48 39 (258)	101 38 48 39 (181)
286 41 38 39 (203)	270 32 48 39 (193)	310 89 48 39 (167)
286 38 48 39 (458)	270 38 48 39 (196)	310 41 48 39 (486)
604 41 48 39 (171)	147 41 48 39 (234)	310 32 48 39 (307)

Table 4-9. The frequent 5-itemsets

Itemsets (frequency)
110 41 38 48 39 (346)
110 32 38 48 39 (201)
36 41 38 48 39 (334)
36 32 38 48 39 (191)
170 41 38 48 39 (413)
170 32 38 48 39 (213)
41 32 38 48 39 (448)

4. 3. 2 Extracting the Association Rules

According to the frequent itemsets (which are a special types of pattern) in tables 4-4, 4-5, 4-6, 4-7, 4-8, and by using minimum confidence 0.8, for finding association rules. The *reason for choosing 0.8 as minimum confidence threshold is, the higher the confidence of the* association rules A→B, the greater probability that if a customer buys products in A, it will also buy product B. Figure 4-2 shows the number of the obtained rules according to different value of confidence.

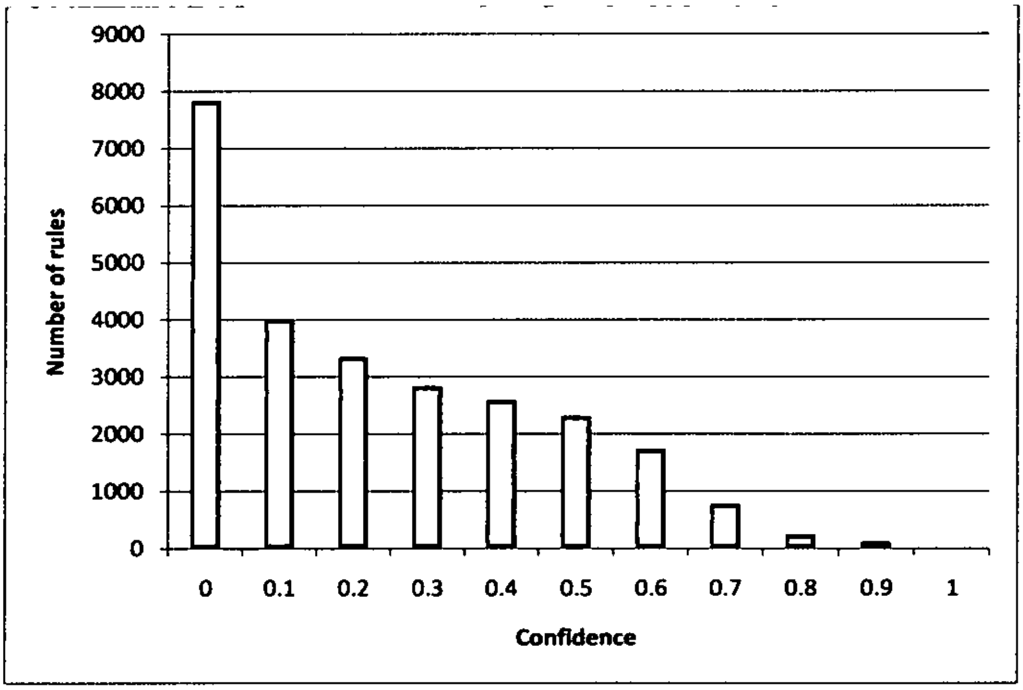


Figure 4-2. Number of rules vs. different confidence values

Table 4-10 shows random sample of the association rules with different number of primitive patterns, i.e. 1, 2, 3, 4. The total number of the association rules obtained by using 0.8 confidence thresholds is 249 rules.

As shown in the Table 4-10 the support for the rule "if item number 16011, then item number 16010" is

Support(X→Y) = (X ∪ Y) = (number of transactions containing X & Y)/ (total number of transactions)

Support (16011 ====>> 16010) = 651/88162 = 0. 0074

That number indicates low support for the rule. This means these two products are bought together only occasionally. A support of 0.0074 means that only 0. 74% of the transactions considered will have both 16011 and 16010 in the basket. The support of an association rules is symmetric; the support of the rule (16011 ====>> 16010) is the same of the support of rule (16010====>>16011).

The confidence of a rule, even when calculated for an association, where order does not matter, depends on the body and head of the rule:

Confidence (X→Y) = P(Y/X) = Support(X∪Y) / Support(X)

Confidence (16011 ====>> 16010) = 651/669= 0. 973094

And

Confidence (16010 ====>>16011) = 651/1316= 0. 494681

The confidence corresponds to the conditional frequency of the rule's body. In the first case it indicates the proportion, among those that buy 16011 of those that also buy 16010. In the second case it indicates the proportion, among those that buy 16010, of those that also buy 16011. Using the same way we can calculate the support and the confidence for the rules of order 3, 4 and 5.

Table 4-10. Set of order two association rules with support and confidence measures

Rule	Frequency	Confidence
16011 ====>> 16010	651	0. 973094
16431 ====>> 16430	348	0. 991453
16430 ====>> 16431	348	0. 816901
1590 ====>> 38	281	0. 959044
1046 ====>> 32	339	0. 928767
734 ====>> 48	329	0. 898907
16430 & 39 ====>> 16431	186	0. 841629
16431 & 48 ====>> 16430	174	0. 994286
1344 & 41 ====>> 39	213	0. 825581
225 & 41 ====>> 39	726	0. 827822
110 & 32 ====>> 38	443	0. 986637
2958 & 39 ====>> 48	570	0. 870229
16217 & 41 & 48 ====>> 39	234	0. 815331
36 & 41 & 48 ====>> 38 & 39	334	0. 860825
225 & 32 & 48 ====>> 39	298	0. 814208
604 & 41 & 48 ====>> 39	171	0. 859297
16011& 41 & 39 ====>> 16010	190	0. 979381
1327 & 41 & 48 ====>> 39	260	0. 846906
36 & 32 & 48 & 39 ====>> 38	191	0. 97449
170 & 41 & 38 & 48 ====>> 39	413	0. 853306

4. 4 Evaluate the Association Rules

Existing association rule mining algorithms employ a support and confidence measures for the discovery of interesting rules. Although these two parameters (minimum support and confidence thresholds) prune many associations discovered, many rules that are not interesting to the user may still be produced. The best demonstration for that problem, is with the following example, we want to study the purchase of tea and coffee (see Table 4-11). In this example we are interested in analyzing the relationship between people who drink tea and coffee. Let Tea refer to the transactions containing Tea, and –Tea refer to those not containing Tea. Let Coffee refer to the transactions containing Coffee and –Coffee refer to those not containing Coffee. The information given in this table can be used to evaluate the association {Coffee}→{Tea}.

Use "support-confidence" framework, say, a minimum support of 30% and a minimum confidence of 60%. Rule {Coffee} →{Tea} has support 40% and confidence 67%, which are reasonably high, is discovered as a valid rule. However, "{Coffee}→{Tea}" is misleading since the probability of purchasing Tea is 75%, which is even larger than 67%. In fact, Coffee and Tea are negatively correlated since the purchase of one of these items actually decreases the likelihood of purchasing the other.

Table 4-11. Purchase of coffee and tea among a group of 1000 people

To From	Coffee	-Coffee	Σ row
Tea	400	350	750
-Tea	200	50	250
Σcol	600	400	1000

The above example indicates the weakness of support-confidence framework. Association rules mined using a support-confidence framework are useful for many applications. However, the support-confidence framework can be misleading if the occurrence of antecedent does not imply the occurrence of consequent. In our scheme we consider an alternative framework for finding interesting relationships between data itemsets based on correlation. Correlation measurement is used to augment the support- confidence framework for association rules which is related to the observed frequency of the rules.

An objective measurement of interestingness for association rules is based on the statistical notion of correlation between the items in the antecedent and the consequent of the rule. The idea is to construct a contingency table from the association rule results and test the interdependence between the antecedent and the consequent of the rule.

We utilize three popular objective measurements which are correlation, cosine, and interest (Tan et al 2006) to see which is the most suitable for our data set. Therefore, a good strategy is to perform the correlation coefficient analysis first, and when the result shows that they are weakly positively/negatively correlated, other analyses can be performed to assist in obtaining a more complete picture.

$$\text{Correlation Coefficient} = \frac{P(AB)-P(A)P(B)}{\sqrt{P(A)P(B)P(\bar{A})P(\bar{B})}}$$

$$\text{Cosine} = \frac{P(AB)}{\sqrt{P(A)P(B)}}$$

$$\text{Interest} = \frac{P(AB)}{P(A)P(B)}$$

The values of the three measurements are shown in table 4-12. The rules are sorted in ascending order of the two columns (support, confidence).

$$\text{Correlation Coefficient (3005 ====>> 38)} = \frac{0.001801}{\sqrt{0.000338}} = 0.\ 0979646$$

$$\text{Cosine (3005 ====>> 38)} = \frac{195}{\sqrt{3197180}} \quad = 0.\ 109056$$

$$\text{Interest (3005 ====>> 38)} = \frac{0.951219}{0.176902} = 5.\ 37711$$

Table 4-12. Comparison of three correlation measures

Rule	Frequency	Confidence	Correlation	Cosine	Interest
3005 ====>> 38	195	0.951219	0.0979646	0.109056	5.37711
597 ====>> 39	209	0.832669	0.0278721	0.058602	1.44864
2805 ====>> 38	212	0.954955	0.102447	0.113934	5.39823
1956 ====>> 48	243	0.804636	0.0383462	0.068121	1.6836
504 ====>> 38	245	0.822148	0.0984773	0.113645	4.64749
1831 ====>> 38	252	0.969231	0.112928	0.125143	5.47893
1188 ====>> 38	256	0.927536	0.110238	0.12339	5.24323
3904 ====>> 39	275	0.806452	0.029199	0.0661544	1.40303
1590 ====>> 38	281	0.959044	0.118361	0.131452	5.42134
1529 ====>> 38	295	0.916149	0.117295	0.13164	5.17886
734 ====>> 48	329	0.898907	0.054415	0.0837787	1.88085
3904 ====>> 38	333	0.97654	0.130581	0.144398	5.52024
1046 ====>> 32	339	0.928767	0.129281	0.14408	5.39869
390 ====>> 38	354	0.941489	0.131134	0.146185	5.32211
370 ====>> 38	362	0.965333	0.135043	0.149688	5.45689
840 ====>> 38	379	0.969309	0.138602	0.153477	5.47937
281 ====>> 38	432	0.953642	0.146289	0.162528	5.39081
47 ====>> 38	432	0.972973	0.148425	0.164167	5.50008
790 ====>> 38	508	0.971319	0.160827	0.177871	5.49073
56 ====>> 38	514	0.960748	0.160508	0.177942	5.43097
1135 ====>> 48	541	0.811094	0.0582355	0.10205	1.69711
105 ====>> 38	643	0.978691	0.182068	0.200873	5.5324
55 ====>> 38	657	0.933239	0.177832	0.198277	5.27547
371 ====>> 38	767	0.980818	0.199304	0.219627	5.54443
2958 ====>> 48	779	0.861726	0.0782057	0.126221	1.80305
37 ====>> 38	1046	0.973929	0.231955	0.255578	5.50549
286 ====>> 38	1116	0.943364	0.234253	0.259816	5.33271
110 ====>> 38	2725	0.975304	0.378526	0.412807	5.51326
36 ====>> 38	2790	0.950273	0.376173	0.412306	5.37176
170 ====>> 38	3031	0.978057	0.400743	0.435982	5.52882

Because there is no measurement constantly superior to others in all application domains we tested the three measurements namely (interest, correlation coefficient, and cosine). We found that interest is not convenient for the data set in this case study due to the conflict information of the interestingness for the obtained rules. In the case of rule (225&48===>39) and (3904===>39) there were conflict in the values of the interest measurement; both of the rules have different values of support but similar values of interest. Therefore our strategy was to perform the correlation coefficient analysis first, and when the result shows that they are weakly positively/negatively correlated, other analyses can be performed to assist in obtaining a more complete picture.

4. 5 Results

After the rules are mined out of the database, the rules are used to understand the business problem better, in our case study the obtained rules generally could be used for designing the store layout and promotional campaign.

Table 4-13 presents the association rules with the highest support out of 249 possible rules. The support can be used to assess the importance of a rule in terms of its frequency in the database. For each rule it shows the support, the confidence, the correlation, the cosine and the interest. Ranking the rules by their supports value help the decision maker to know what the most ubiquitous rules are. As shown in the table the rule with highest support is {41 & 48 ====>> 39} which appears almost in 8.3%. This is followed by {170 ====>> 38}, {36 ====>> 38}, and {110 ====>> 38}, all occurring in about 3.0 % of the transactions.

Table 4-14 presents the association rules with the highest confidence out of 249 possible rules. The confidence can be used to investigate possible dependences between variables. From Table 4-14 we can see, for example, that {37 & 41 & 48 ====>> 38} has a confidence equal to 1. This means that if a transaction contains {37 & 41 & 48}, it will also contain {38} about 100% of the time. The highly confidence value imply strong relationships that can be exploited. This rule followed by {110 & 41 & 48 & 39 ====>> 38}, to the end of the table, with confidence values about 0.9. Notice that the confidence value of {37 & 41 & 48 ====>> 38} is 100%, however the support of this rule is only 0. 195.

Finally, to obtain a relative measure of a rule's interestingness we can also consider the correlation. The correlation can be used to measure the distance from the situation of independence. Table 4-15 reports the rules with the highest correlation out of the 249 possible rules. Notice that {16431 ====>> 16430} and {16430 ====>> 16431} come first, both with a correlation about 0. 899533, which means they are positively correlated.

For the three tables (4-13, 4-14, 4-15) the items {38}, {39}, {48}, {16010}, {16430}, and {16431} are the only heads selected. We chose the highest five rules, which are ordered by the support, confidence, and correlation. Those rules are {41 & 48 ====>> 39}, {170 ====>> 38}, {36 ====>> 38}, {110 ====>> 38}, and {170 & 39 ====>> 38}. The whole set of obtained the association rules is presented in Appendix 5.

In Table 4-16 it contains all the rules that have certain value for consequent; it can be used to understand what is the associated with the consequent and perhaps what affects the consequent. For instance, it might be useful to know all of the interesting rules that have "item number 39" in their consequent. These may will be the rules that affect the purchases of item 39 and that a store owner may what to put close to the item 39 in order to increase the sale of both items. Or it may be used to determine what the items to place are in the promotional campaign.

Table 4-13. Association rules with highest support

Rule	Frequency	Confidence	Correlation	Cosine	Interest
41 48 ====>> 39	7366	0. 816811	0. 165248	0. 344572	1. 42105
170 ====>> 38	3031	0. 978057	0. 400743	0. 435982	5. 52882
36 ====>> 38	2790	0. 950273	0. 376173	0. 412306	5. 37176
110 ====>> 38	2725	0. 975304	0. 378526	0. 412807	5. 51326
170 39 ====>> 38	2019	0. 980573	0. 325691	0. 356288	5. 54304
41 38 48 ====>> 39	1991	0. 838669	0. 088791	0. 181524	1. 45908
36 39 ====>> 38	1945	0. 954836	0. 313532	0. 345078	5. 39755
110 39 ====>> 38	1740	0. 989198	0. 303733	0. 332208	5. 5918
170 48 ====>> 38	1538	0. 987797	0. 284935	0. 312108	5. 58388
225 48 ====>> 39	1400	0. 806452	0. 0664115	0. 149264	1. 40303
110 48 ====>> 38	1361	0. 986232	0. 26746	0. 293367	5. 57503
36 48 ====>> 38	1360	0. 960452	0. 262351	0. 289401	5. 4293
170 48 39 ====>> 38	1193	0. 989221	0. 250703	0. 275081	5. 59193
286 ====>> 38	1116	0. 943364	0. 234253	0. 259816	5. 33271
36 48 39 ====>> 38	1080	0. 967742	0. 234668	0. 258872	5. 47051
37 ====>> 38	1046	0. 973929	0. 231955	0. 255578	5. 50549
110 48 39 ====>> 38	1031	0. 994214	0. 233676	0. 256367	5. 62015
170 41 ====>> 38	794	0. 986335	0. 203629	0. 224087	5. 57562
2238 48 ====>> 39	788	0. 825131	0. 0529901	0. 113273	1. 43552
2958 ====>> 48	779	0. 861726	0. 0782057	0. 126221	1. 80305
371 ====>> 38	767	0. 980818	0. 199304	0. 219627	5. 54443
286 39 ====>> 38	728	0. 970667	0. 192684	0. 21286	5. 48704
225 41 ====>> 39	726	0. 827822	0. 051303	0. 108903	1. 44021
37 39 ====>> 38	684	0. 967468	0. 186279	0. 205987	5. 46896
36 41 ====>> 38	671	0. 958571	0. 183261	0. 20308	5. 41867
110 41 ====>> 38	666	0. 983752	0. 186007	0. 204962	5. 56101
55 ====>> 38	657	0. 933239	0. 177832	0. 198277	5. 27547
16011 ====>> 16010	651	0. 973094	0. 690949	0. 693809	65. 1899
105 ====>> 38	643	0. 978691	0. 182068	0. 200873	5. 5324
310 41 ====>> 39	625	0. 869263	0. 0540114	0. 103543	1. 5123
89 41 ====>> 39	619	0. 845628	0. 0501272	0. 101634	1. 47118
49 39 ====>> 48	617	0. 803385	0. 0610786	0. 108463	1. 68098
170 41 39 ====>> 38	615	0. 985577	0. 178927	0. 197141	5. 57133
89 41 ====>> 48	606	0. 827869	0. 0641025	0. 109118	1. 73221
286 48 ====>> 38	581	0. 98308	0. 173561	0. 191371	5. 55721
89 32 ====>> 48	573	0. 803647	0. 0588796	0. 104541	1. 68153
36 41 ====>> 39	572	0. 817143	0. 0438555	0. 0960396	1. 42163
2958 39 ====>> 48	570	0. 870229	0. 0679475	0. 108501	1. 82084
37 48 ====>> 38	557	0. 985841	0. 170256	0. 18764	5. 57282
36 41 38 ====>> 39	553	0. 824143	0. 0441704	0. 0948346	1. 43381
36 41 39 ====>> 38	553	0. 966783	0. 167279	0. 185149	5. 46509
65 41 48 ====>> 39	547	0. 825038	0. 0440618	0. 0943699	1. 43536
1135 ====>> 48	541	0. 811094	0. 0582355	0. 10205	1. 69711
170 32 ====>> 38	532	0. 985185	0. 166289	0. 183319	5. 56911
371 39 ====>> 38	526	0. 988722	0. 165767	0. 182609	5. 58911

Table 4-14. Association rules with highest confidence

Rule	Frequency	Confidence	Correlation	Cosine	Interest
37 41 48 ====>> 38	172	1	0. 0953691	0. 105017	5. 65286
110 41 48 39 ====>> 38	346	0. 997118	0. 135119	0. 148732	5. 63657
371 48 39 ====>> 38	294	0. 99661	0. 12447	0. 137066	5. 6337
105 48 39 ====>> 38	290	0. 996564	0. 123614	0. 136127	5. 63343
37 41 ====>> 38	246	0. 995951	0. 113772	0. 125337	5. 62997
37 41 39 ====>> 38	202	0. 995074	0. 103006	0. 113526	5. 62501
110 32 48 39 ====>> 38	201	0. 995049	0. 102748	0. 113244	5. 62488
37 32 ====>> 38	185	0. 994624	0. 0985345	0. 10862	5. 62247
16431 48 ====>> 16430	174	0. 994286	0. 636346	0. 637273	205. 77
110 48 39 ====>> 38	1031	0. 994214	0. 233676	0. 256367	5. 62015
371 41 ====>> 38	171	0. 994186	0. 0946954	0. 104406	5. 61999
110 32 39 ====>> 38	284	0. 993007	0. 122012	0. 134471	5. 61333
110 41 39 ====>> 38	511	0. 992233	0. 163786	0. 180306	5. 60895
16431 ====>> 16430	348	0. 991453	0. 899533	0. 899955	205. 184
170 32 48 39 ====>> 38	213	0. 990698	0. 105447	0. 11632	5. 60028
110 41 48 ====>> 38	419	0. 990544	0. 148052	0. 163131	5. 59941
170 32 48 ====>> 38	305	0. 99026	0. 126207	0. 139161	5. 5978
790 41 ====>> 38	202	0. 990196	0. 102644	0. 113248	5. 59744
170 48 39 ====>> 38	1193	0. 989221	0. 250703	0. 275081	5. 59193
110 39 ====>> 38	1740	0. 989198	0. 303733	0. 332208	5. 5918
371 48 ====>> 38	365	0. 98916	0. 138002	0. 15215	5. 59158
110 32 48 ====>> 38	264	0. 988764	0. 117264	0. 129372	5. 58934
371 39 ====>> 38	526	0. 988722	0. 165767	0. 182609	5. 58911
170 89 ====>> 38	169	0. 988304	0. 0937397	0. 103486	5. 58674
170 48 ====>> 38	1538	0. 987797	0. 284935	0. 312108	5. 58388
286 48 39 ====>> 38	458	0. 987069	0. 154436	0. 170255	5. 57976
105 39 ====>> 38	449	0. 986813	0. 152874	0. 168552	5. 57832
110 32 ====>> 38	443	0. 986637	0. 151825	0. 167407	5. 57732
105 48 ====>> 38	367	0. 986559	0. 138121	0. 152366	5. 57688
170 41 ====>> 38	794	0. 986335	0. 203629	0. 224087	5. 57562
110 48 ====>> 38	1361	0. 986232	0. 26746	0. 293367	5. 57503
37 48 ====>> 38	557	0. 985841	0. 170256	0. 18764	5. 57282
170 41 48 39 ====>> 38	413	0. 98568	0. 146467	0. 161561	5. 57191
170 41 39 ====>> 38	615	0. 985577	0. 178927	0. 197141	5. 57133
170 32 ====>> 38	532	0. 985185	0. 166289	0. 183319	5. 56911
16431 39 ====>> 16430	186	0. 984127	0. 654572	0. 655507	203. 668
37 48 39 ====>> 38	433	0. 984091	0. 149815	0. 165293	5. 56293
110 41 ====>> 38	666	0. 983752	0. 186007	0. 204962	5. 56101
170 41 48 ====>> 38	484	0. 98374	0. 158399	0. 174725	5. 56094
790 48 39 ====>> 38	235	0. 983264	0. 110176	0. 12172	5. 55825
286 48 ====>> 38	581	0. 98308	0. 173561	0. 191371	5. 55721
170 32 39 ====>> 38	326	0. 981928	0. 129708	0. 143266	5. 5507
47 48 ====>> 38	269	0. 981752	0. 11777	0. 130128	5. 54971
47 48 39 ====>> 38	212	0. 981481	0. 104495	0. 115505	5. 54818
36 170 ====>> 38	210	0. 981308	0. 103987	0. 114949	5. 5472

Table 4-15. Association rules with highest correlation

Rule	Frequency	Confidence	Correlation	Cosine	Interest
16431 ====>> 16430	348	0. 991453	0. 899533	0. 899955	205. 184
16430 ====>> 16431	348	0. 816901	0. 899533	0. 899955	205. 184
16011 ====>> 16010	651	0. 973094	0. 690949	0. 693809	65. 1899
16430 39 ====>> 16431	186	0. 841629	0. 66683	0. 667826	211. 395
16431 39 ====>> 16430	186	0. 984127	0. 654572	0. 655507	203. 668
16431 48 ====>> 16430	174	0. 994286	0. 636346	0. 637273	205. 77
16011 39 ====>> 16010	419	0. 981265	0. 555949	0. 558949	65. 7373
16011 48 ====>> 16010	362	0. 967914	0. 512961	0. 515994	64. 8429
16011 48 39 ====>> 16010	269	0. 978182	0. 444349	0. 447155	65. 5307
16011 41 ====>> 16010	236	0. 967213	0. 413716	0. 416475	64. 7959
170 ====>> 38	3031	0. 978057	0. 400743	0. 435982	5. 52882
110 ====>> 38	2725	0. 975304	0. 378526	0. 412807	5. 51326
36 ====>> 38	2790	0. 950273	0. 376173	0. 412306	5. 37176
16011 41 39 ====>> 16010	190	0. 979381	0. 373507	0. 376032	65. 6111
170 39 ====>> 38	2019	0. 980573	0. 325691	0. 356288	5. 54304
36 39 ====>> 38	1945	0. 954836	0. 313532	0. 345078	5. 39755
110 39 ====>> 38	1740	0. 989198	0. 303733	0. 332208	5. 5918
170 48 ====>> 38	1538	0. 987797	0. 284935	0. 312108	5. 58388
110 48 ====>> 38	1361	0. 986232	0. 26746	0. 293367	5. 57503
36 48 ====>> 38	1360	0. 960452	0. 262351	0. 289401	5. 4293
170 48 39 ====>> 38	1193	0. 989221	0. 250703	0. 275081	5. 59193
36 48 39 ====>> 38	1080	0. 967742	0. 234668	0. 258872	5. 47051
286 ====>> 38	1116	0. 943364	0. 234253	0. 259816	5. 33271
110 48 39 ====>> 38	1031	0. 994214	0. 233676	0. 256367	5. 62015
37 ====>> 38	1046	0. 973929	0. 231955	0. 255578	5. 50549
170 41 ====>> 38	794	0. 986335	0. 203629	0. 224087	5. 57562
371 ====>> 38	767	0. 980818	0. 199304	0. 219627	5. 54443
286 39 ====>> 38	728	0. 970667	0. 192684	0. 21286	5. 48704
37 39 ====>> 38	684	0. 967468	0. 186279	0. 205987	5. 46896
110 41 ====>> 38	666	0. 983752	0. 186007	0. 204962	5. 56101
36 41 ====>> 38	671	0. 958571	0. 183261	0. 20308	5. 41867
105 ====>> 38	643	0. 978691	0. 182068	0. 200873	5. 5324
170 41 39 ====>> 38	615	0. 985577	0. 178927	0. 197141	5. 57133
55 ====>> 38	657	0. 933239	0. 177832	0. 198277	5. 27547
286 48 ====>> 38	581	0. 98308	0. 173561	0. 191371	5. 55721
37 48 ====>> 38	557	0. 985841	0. 170256	0. 18764	5. 57282
170 41 48 ====>> 38 39	413	0. 839431	0. 168084	0. 183064	7. 15378
36 41 39 ====>> 38	553	0. 966783	0. 167279	0. 185149	5. 46509
170 32 ====>> 38	532	0. 985185	0. 166289	0. 183319	5. 56911
371 39 ====>> 38	526	0. 988722	0. 165767	0. 182609	5. 58911
41 48 ====>> 39	7366	0. 816811	0. 165248	0. 344572	1. 42105
110 41 39 ====>> 38	511	0. 992233	0. 163786	0. 180306	5. 60895
790 ====>> 38	508	0. 971319	0. 160827	0. 177871	5. 49073
56 ====>> 38	514	0. 960748	0. 160508	0. 177942	5. 43097
170 41 48 ====>> 38	484	0. 98374	0. 158399	0. 174725	5. 56094

Table 4-16. Target the consequent

Rule Antecedent		Rule Consequent
41 48	====>>	39
41 38 48	====>>	39
225 48	====>>	39
2238 48	====>>	39
225 41	====>>	39
170	====>>	38
36	====>>	38
110	====>>	38
170 39	====>>	38
36 39	====>>	38
36 41 48	====>>	38 39
37 41	====>>	38 39
110 41 48	====>>	38 39
3904 48	====>>	38 39
2958	====>>	48
49 39	====>>	48
89 41	====>>	48
89 32	====>>	48
2958 39	====>>	48
1046	====>>	32
1046 48	====>>	32
16011	====>>	16010
16011 39	====>>	16010
16011 41	====>>	16010
16011 48	====>>	16010
16011 48 39	====>>	16010
16431	====>>	16430
16431 39	====>>	16430
16431 48	====>>	16430
16430	====>>	16431

4. 6 Summary

In this chapter, the association between buying behaviors was carefully studied. The main objective is to track the most important buying patterns, where the patterns mean a group ŏf products bought together by customer. In this case study we considered support, confidence, and correlation as main measures for validating a set of association rules. The needs of the user will govern which of these three is the best one for selecting a set of rules. The support can be used to assess the importance of a rule in terms of its frequency in the database; the confidence can be used to investigate possible dependences between variables; and the correlation can be used to measure the distance from the situation of independence.

Ultimately, a set of rules has to be assessed on its ability to meet the analysis objectives. Here the objectives are primarily to reorganize the layout of a sales outlet and to plan promotions so as to increase revenues. Once the associations have been identified, it is possible to organize promotions within the outlet so the products that are put on offer at the same time are products which are not associated. Correspondingly, by putting one product on promotion; we also increase the sales of the associated products.

At the beginning of this chapter, we presented the main objective of this case study, which was to analysis customer buying behavior. The data set used in this case study was confidential; we did not have the chance to make comments on the data. We only could blindly mine and find the association rules. If we had more knowledge on the data then we could drive more results. The data consisted of items and the amount of items sold at each transaction. Since we did not know what item was the amount did not mean anything to us. Also some of the amounts were fractions i.e. in kilogram and some of them were in numbers so we did not have a way of using the amount of the item in the mining of the data. We had to take the data as binary i.e. it exists in this transaction or not. Also since the data was confidential we did not have information on the prices of the items to prioritize the items.

The next step was to find the frequent itemsets; this resulted in 3029 frequent itemset sets of size [1] to [5]. In the 1-itemsets is 50675 for the itemset {39}. Note for the 2-itemsets the highest frequency is 29142 for the itemset {48, 39}. While, for the 3-itemset, the highest frequency is 7366 for the itemset {41, 48, 39}. The highest frequency for 4-itemsets is 1991 for the itemset {41, 38, 48, 39}. And finally, in the case of 5-itemsets, the highest frequency is 448 for the itemset {41, 32, 38, 48, 39}.

The number of the association rules from the frequent itemsets is 249 possible rules. We chose the highest five rules, which are ordered by the support, confidence, and correlation. Those rules are {41 & 48 ====>> 39}, {170 ====>> 38}, {36 ====>> 38}, {110 ====>> 38}, and {170 & 39 ====>> 38}.

CHAPTER FIVE: PERFORMANCE STUDY

This chapter demonstrates the experiments that we have performed to evaluate the new scheme. For the evaluation purpose we have conducted several experiments using our market basket data and the existing data set (namely, Mushroom data set). Those experiments performed on computer with Core 2 Duo 2.00 GHZ CPU, 2.00 GB memory and hard disk 160 GB [Serial ATA-150 - 5400.0 rpm]. The operating system is ubuntu 8.10 and the G++ compiler [Ver. 3.4]. All the algorithms were developed by C++ language. The runtime includes both system time and user time, and was measured by Time (Linux command). The memory consumption was measured by Memusage (Linux command).And for the unit of measuring the time and the memory are second and megabyte respectively.

5.1 Benefit Analysis

As the result of the experimental study, we revealed the performance statistics capabilities of our proposed scheme against FP-growth and Apriori approach.

5.1.1 Comparison on Collected Data

In this section we visualize, evaluate and compare the results obtained from simulation on the supermarket dataset. The run time and the memory consumption were calculated simply by adding the mining time and memory to the generating of the association rules time and memory respectively. The experimental result of time is shown in Figure 5-1 reveals that the proposed scheme outperforms the FP-growth and the Apriori approach.

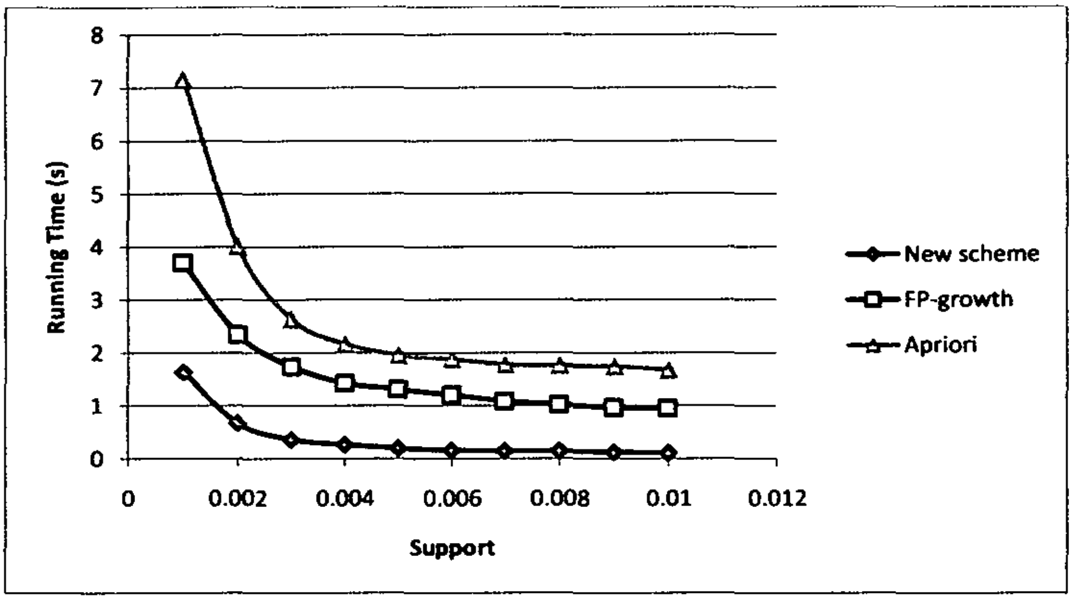


Figure 5-1. The execution time at various support levels on the supermarket dataset

The experiment in Figure 5-2 shows the memory consumption measure by the megabyte. The experiment revealed that our proposed scheme took the lead at low levels of support comparing with FP-growth and Apriori algorithm. At high levels support the performances of our proposed scheme and Apriori are near, this result can be explained by sparseness of our data set. Apriori exhibits a better mining performance on our relatively sparse supermarket data set in comparison to FP-growth. By "relatively sparse" means the data sets containing more enough zero entries (unmarked fields or items), in other words, the ratio number of fields / number of elements for the data database is smaller. Since the Apriori algorithm stores and processes only the non-zero entries, it takes the advantage of pruning most of the infrequent items during the first few passes.

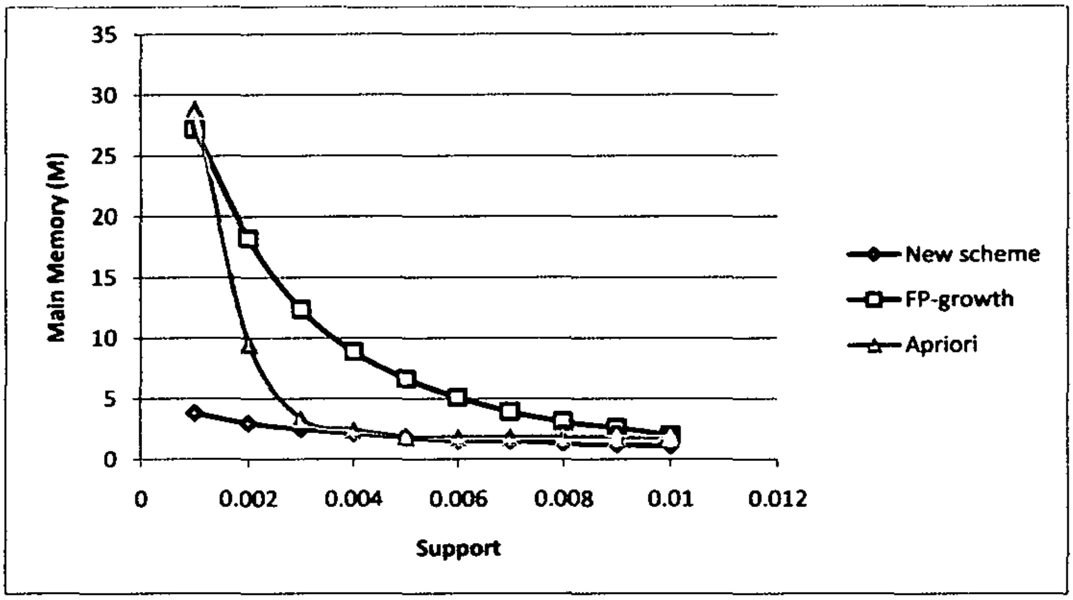


Figure 5-2. The memory usage at various support levels on the supermarket dataset

5.1.2 Comparison on Existing Data

In this section we compare the new scheme with FP-growth and apriori on existing data set namely mushroom data set. Using the same simulation environment we tested the performance of our scheme against the other two (FP-growth and apriori). Figure 5-3 depicts the running time for generating the association rules by using the proposed scheme, FP-growth, and the Apriori approach. Different values of support were utilized because it has different size of frequent itemsets which in turn have a huge effect on the performance. From Figure 5-3, it is clear that the performances of our proposed scheme and FP-growth are near.

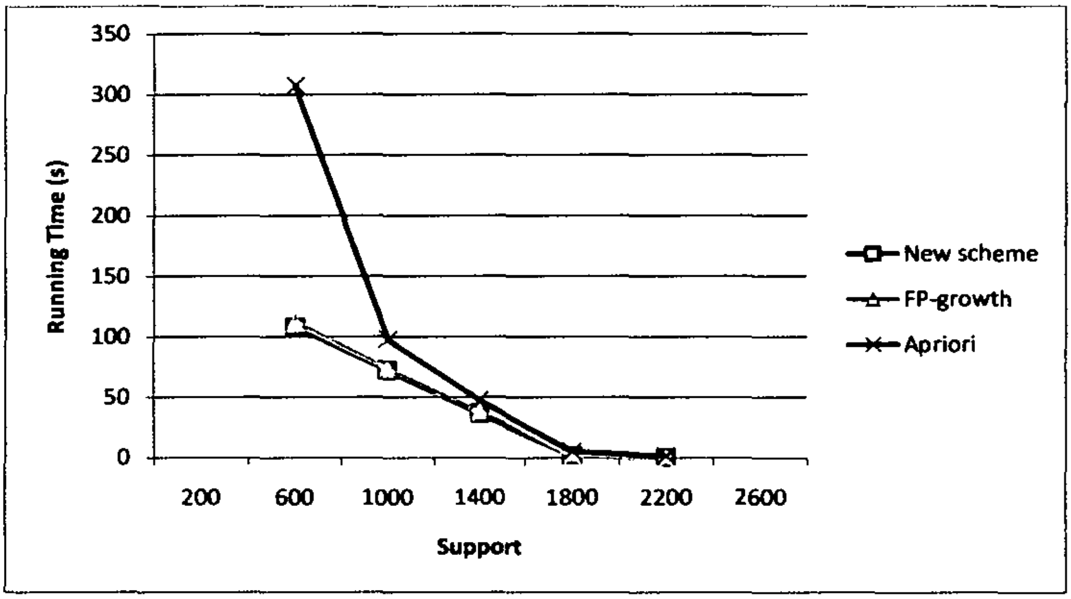


Figure 5-3. The execution time at various support levels on Mushroom dataset

Figure 5-4 shows the results we obtained from the experimentation for the memory consumption on the mushroom data set. Mushroom data set contains characteristics of different species of mushrooms, which contains 8124 transactions and average length of items 23. From the graph, the new scheme performed well over the other two (FP-growth and Apriori) for all level of support values. For the high level of the support the performance of FP-growth draw near to the performance of the new scheme. The explanation for the running time and memory consumption for both the new scheme and the FP-growth is that, are using a compressed data representation to facilitate in-core processing of the itemsets.

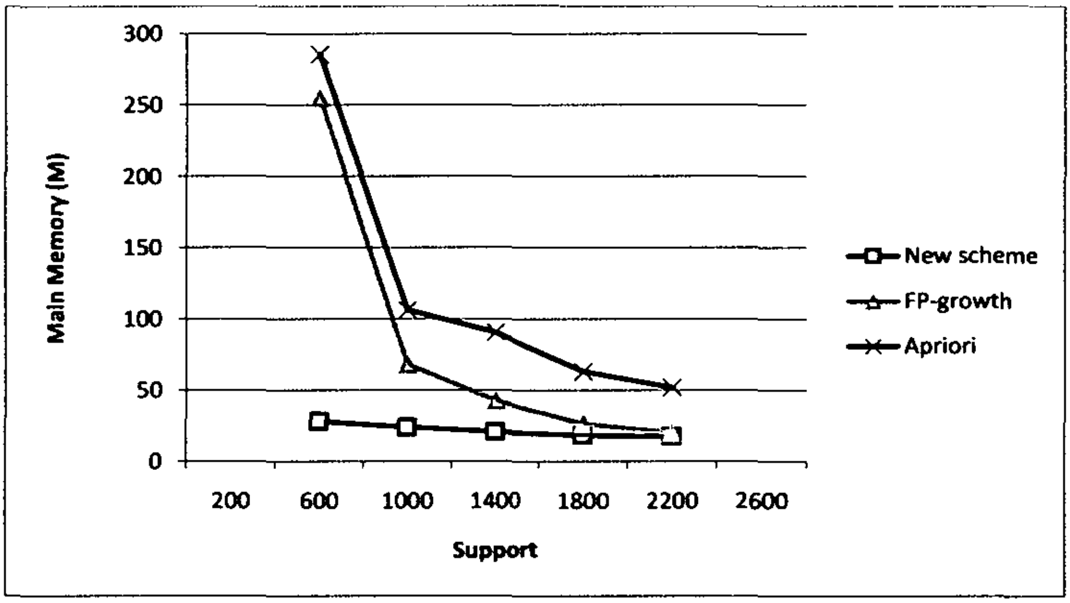


Figure 5-4. The memory usage at various support levels on Mushroom dataset

5.2 Summary

In this chapter we have performed several experiments to evaluate the performance of our scheme against FP-growth and Apriori, for generating the association rules. To perform the experiments different values of support were set because with different value of support the number of the frequent itemsets is different, and the running time and the memory consumptions are affected by the value of the support.

For both data sets the running time of our new scheme outperformed Apriori. Whereas the memory consumption for the new scheme and the Apriori on the collected data set were near for middle and high support values. While on the mushroom data set our scheme outperformed the Apriori for all the level of support.

The running time of our scheme performed well over the FP-growth on the collected data set. While, on the mushroom data set there were high competition among both for all level of supports. For the memory consumption, our scheme outperformed FP-growth on the collected data set for all level of support. In the case of the mushroom data set, the memory consumption of our scheme exceed in performance FP-growth for the low and middle value of support. While the performance of both became near with high level of support. Finally, the result of the simulation shows the efficiency of our new scheme on both data sets.

CHAPTER SIX: CONCLUSION AND FUTURE RESEARCH

The association rules mining is somewhat young discipline with broad and diverse applications, there is still nontrivial gap between general principles of association rules mining and its applications. Nearly all of the previous studies were using apriori approach for extracting the association rules, which is inefficient approach. The goal of this research was to find a scheme for pulling the rules out of the transactional data sets which considers the time, the memory consumption, and the interestingness of the rules. This chapter summarizes the work has been done in this thesis first, and then the future trends are given.

6. 1 Conclusions

In this thesis, we considered the following factors for creating our new scheme, which are the time, the memory consumption, and the interestingness of the rules. For the first two factors are affected by the approach for finding the frequent itemsets and the approach for generating the association rules. Experiments with synthetic as well as real-life data are performed to compare the performance of the existing algorithms. From the analysis of the existing algorithms for mining frequent itemsets we found that nonordfp approach is the most efficient among the other approaches (FP-growth, AFOPT, FP-growth*). According to our observations, the performances of the algorithms are strongly depending on the support levels and the features of the data sets (the nature and the size of the data sets). Therefore we employed it in our scheme to guarantee the save of the time and the memory in the case of sparse and dense data sets. In addition, to employee the nonordfp approach in our scheme,

We used the Apriori rule generation approach to generate the association rules as well. The selection of this approach was for two reasons; the first reason, we are interesting to mine association rules with single dimension, single level from frequent itemset in the transactional databases, which is based on the type of the available data. And the second reason, depending on preliminary test of time and memory consumption we found that

Apriori rule generation approach is stable and consume a little amount of time and memory during the process of rule generation.

For the last factor of our new scheme basicly depends on the type of the data set. That is because we used objective measurements which is affected by the charasitesctic of the data.Therefore the choosing of the measurement was determined after specifying the data set.

To achieve the second objective of this research an empirical data set is kindly provided from anonymous retail supermarket store. The data collected over six months, started by the first of December 2007 until the end of May 2008. This data set was employed to verify the effectiveness of the new scheme via a case study. The problem of that case study was how to redesign the store layout and the promotional campaign from the company-oriented to customer-oriented approach. Therefore the objective was to analysis the customer buying behavior by using the daily buying records. Toward this objective the new scheme was employed to extract the association rules.

To implement our scheme on this case study, different phases to extract the association rules from the transactional database using our scheme were specified, starting with business understanding, and continues throughout data Assembling, data preprocessing, model building (using the new scheme), post-processing of association rules, and results interpretation. For business interest the data used in the thesis was confidential so we mined data blindly without knowing what we mined.

The first phase was specified by the business analysis; this was how to redesign the store layout and the promotional campaign from the company-oriented to customer-oriented approach. The second phase in our methodology was to assemble the data which we had in the format of a relational database. We first converted the data into horizontal database layout <TID, {item1, item 2,... , item n}>, to be suited for our scheme. In data preprocessing phase, we selected the interesting attributes of the data set, therefore considered only transactions data on products, in order to investigate the association between these products.

Model building phase used the new scheme for analyzing the supermarket data set. The input thresholds for the scheme were the support value and the confidence value, which were

0.00187 and 0.8 respectively. In our analysis we used small value of support for two reasons, the first we did not want to lose interesting patterns even if it is happened seldom, and the second reason, the computational efficiency of the chosen algorithm allowed us to use this low value of support. And the reasons behind set high level of confidence were the higher the confidence of the association rules A→B, the greater probability that if a customer buys products in A, it will also buy product B. the second reason was, with high level of the confidence the number of the obtained rules was manageable.

Post-processing of association rules phase required us to specify the objective measurement to measure the interestingness of the obtained rules. Because there is no measure that is consistently better than others in all application domains. We tested the most common measurements that have been used in the literature, namely (interest, correlation coefficient, and cosine). We found that interest is not convenient for the data set in this case study due to the conflict information of the interestingness for the obtained rules. Therefore our strategy was to perform the correlation coefficient analysis first, and when the result shows that they are weakly positively/negatively correlated, other analyses can be performed to assist in obtaining a more complete picture.

Results interpretation phase was concerned about summarize the rules and presented it in a simple way. After the rules are mined out of the database, the rules are used to understand the problem better. To summarize the obtain rules we followed four methods which are: by ranking the rules by their supports value help the decision maker to know what the most ubiquitous rules are. By ranking the rules by their confidence value highly confidence value imply strong relationships. By considering the correlation to obtain a relative measure of a rule's interestingness. By summarizing all the rules that have certain value for consequent; it can be used to understand what is the associated with the consequent and perhaps what affects the consequent.

As a result of the analyzing the supermarket data set our findings were: The single item which is sold the most came out to be item number 39. It has 50675 frequencies over the whole data set. In the case of two itemsets the highest frequency is 29142 for the itemset {39, 48}. And we found that the highest five rules, which are ordered by the support, confidence, and correlation. Those rules are

```
{41 & 48 ====>> 39   , support=7366, confidence=0.816811, correlation=0.165248},
{170 ====>> 38       , support=3031, confidence=0.978057, correlation=0.400743},
{36 ====>> 38        , support=2790, confidence=0.950273, correlation=0.376173},
{110 ====>> 38       , support=2725, confidence=0.975304, correlation=0.378526},
{170 & 39 ====>> 38  , support=2019, confidence=0.980573, correlation=0.325691}.
```

Several experiments have been performed to evaluate the performance of our scheme against FP-growth and Apriori, for generating the association rules. To perform the experiments different values of support were set because with different value of support the number of the frequent itemsets is different, and the running time and the memory consumptions are affected by the value of the support.

For both data sets the running time of our new scheme outperformed Apriori. Whereas the memory consumption for the new scheme and the Apriori on the collected data set were near for middle and high support values. While on the mushroom data set our scheme outperformed the Apriori for all the level of support.

The running time of our scheme performed well over the FP-growth on the collected data set. While, on the mushroom data set there were high competition among both for all level of supports. For the memory consumption, our scheme outperformed FP-growth on the collected data set for all level of support. In the case of the mushroom data set, the memory consumption of our scheme exceed in performance FP-growth for the low and middle value of support. While the performance of both became near with high level of support. Finally, the result of the simulation showed the efficiency of our new scheme on both data sets.

The main contributions of this research:

We can summarize the main contribution of this research as follows:

- New scheme is devised to generate association rules among co-occurrence items.
- New interestingness measure is used in retailer industry application to capture rules of strong interest.
- Market basket analysis case study using the new scheme.

6. 2 Future Trends

There are a number of future research directions based on the work presented in this thesis.

- Using constraints can further reduce the size of itemsets generated and improve mining efficiency.
- In our scheme we used binary variables, using other type of variables can further develop our scheme.
- This scheme was applied in retailer industry application, trying other industry is an interesting field for future work.

PUBLICATIONS

1. **Aiman Moyaid said** , P D D. Dominic, Azween B Abdullah, "A Comparative Study of FP-growth Variations", International Journal of Computer Science and Network Security, Seoul, Korea, Vol.9,Page:266, May 2009.
2. **Aiman Moyaid Said,** P D D. Dominic, Azween B Abdullah, "Improving Mining Efficiency: "A New Scheme for Extracting Association Rules", Second International Conference on Computing & Informatics, Kuala Lumpur, Malaysia, Page: 268, June 2009.
3. **Aiman Moyaid Said,** P D D. Dominic, Azween B Abdullah, "The Trends of Optimization on Frequent Pattern Growth", International Conference on Software Engineering and Computer Systems, Pahang, Malaysia, Page:392 ,October 2009.

REFERENCES

Agrawal R. , Imielinski T. , and Swami A. N. , (1993), 'Mining association rules between sets of items in large databases', In Proceedings ACM SIGMOD International Conference on Management of Data, Vol. 22, No. 2, of SIGMOD Record, Washington, Page(s): 207–216.

Agrawal R. and Srikant R., (1994), 'Fast algorithms for mining association rules'. In Proceedings 20th International Conference on Very Large Data Bases (VLDB'94), Page(s): 487-499.

Agrawal R. and Srikant R. ,(1995), 'Mining sequential patterns' . In Proceeding of the 11th International Conference on Data Engineering, Taipei, Taiwan, Page(s):3-14.

Barbará D., Couto J., Jajodia S. and Wu N., (2001), 'ADAM: a testbed for exploring the use of data mining in intrusion detection', ACM SIGMOD Record, Vol. 30(4), Page(s):15-24.

Brin S. Motwani R., Ullman J. and 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, Vol. 6, No. 2: New York, Page(s): 255 - 264.

Brin S. , Motwani R. , and Silverstein C. , (1998), 'Beyond market baskets: Generalizing association rules to correlations', Data Mining and Knowledge Discovery Journal, Vol. 2: Page(s): 39-68.

Changguo Y., Qin Z., Jingwei Z., Nianzhong W., Xiaorong Z., Tailei W.,(2009), 'Improvement of association rules mining algorithm in wireless network intrusion detection', In Proceedings of International Conference on Computational Intelligence and Natural Computing, Vol. 2, Page(s):413-416.

Chen G. and Wei Q., (2002), 'Fuzzy association rules and the extended mining algorithms', Information Sciences—Informatics and Computer Science: An International Journal, Vol. 147 No.1-4, Page(s): 201-228.

Chen Y.-L., Tang K., Shen R.-J., and Hu Y.-H., (2005) , 'Market basket analysis in a multiple store environment', Decision Support Systems, Vol. 40, No. 2, Page(s):339-354.

Cheng H., (2008), 'Towards accurate and efficient classification: a discriminative and frequent pattern-based approach', PH.D. Dissertation, University of Illinois at Urbana-Champaign.

Cheng J., Ke Y., and Ng W., (2008), 'Effective elimination of redundant association rules', Data Mining and Knowledge Discovery Journal, Vol. 16, Page(s): 221–249.

Dokas P., Ertoz L., Kumar V., Lazarevic A., Srivastava J., and Tan P.N. ,(2002),'Data mining for network intrusion detection', In Proceeding NSF Workshop on Nest Generation Data Mining. , Page(s): 21-30.

Dong X. , Niu Z. , Shi X. , Zhang X. and Zhu D. , (2007), 'Mining both positive and negative association rules from frequent and infrequent itemsets', Springer Berlin / Heidelberg, Vol. 4632/2007, Page(s): 122-133.

Dong X, Zheng Z. and Niu Z. , (2007), 'Mining infrequent itemsets based on multiple level minimum supports', In Proceeding of the Second International Conference on Innovative Computing, Information and Control (ICICIC), Page(s): 528.

Dong-Peng Y. , Lun L. Jin-Lin R. , and Chao Z. , (2008), 'Applications of data mining methods in the evaluation of client credibility', IOS Press, Vol. 177, Page(s):35-43.

El-Hajj M. and Zaïane O. R., (2003), 'COFI-tree mining: a new approach to pattern growth with reduced candidacy generation', Workshop on Frequent Itemset Mining Implementations (FIMI'03) in conjunction with IEEE-ICDM.

Fayyad U. M., Piatetsky-Shapiro G. and Smyth, P. , (1996), 'From data mining to knowledge discovery in databases, AI Magazine', Vol. 17, No. 3, Page(s): 37-54.

Fayyad U. M. , Piatetsky-Shapiro G. , and Piatetsky-Shapiro P. X. , (1996), 'From data mining to knowledge discovery: an Overview', Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press, Page(s): 1-36.

Gao J. , (2007), 'Realization of new association rule mining algorithm', In Proceeding of International Conference on Computational Intelligence and Security, IEEE CNF. Page(s):201 – 204.

Gerardo B. D., Lee J., Lee J., Park M., and Lee M.,(2004), 'The association rule algorithm with missing data in data mining', Springer-Verlag Berlin Heidelberg, ICCSA 2004, LNCS 3043, Page(s): 97–105.

Giuffrida G., Cantone V. , and Tribulato G. , (2008), 'An apriori based approach to improve on-line advertising performance', IOS Press, Vol. 177, Page(s): 53–63.

Gordon L., (2008), 'Leading practices in market basket analysis: how top retailers are using market basket analysis to win margin and market share', [On- line]. Available: www. irgintl. com/pdf2/1. pdf.

Grahne G. , and Zhu J. , (2005), 'Fast algorithm for frequent itemset mining using fp-trees', IEEE Transactions on Knowledge and Data Engineer (TKDE Journal), Vol. 17, Issue 10, Page(s): 1347 - 1362.

Grahne O. and Zhu J. , (2004), 'Efficiently using prefix-trees in mining frequent itemsets', In Proceeding of the IEEE ICDM Workshop on Frequent Itemset Mining (FIMI04), Brighton.

Han J. , and Fu Y., (1995), 'Discovery of multiple-level association rules from large databases', In Proceedings of the 21th International Conference on Very Large Data Bases, Page(s): 420-431.

Han J., Pei J. and Yin Y., (2000), 'Mining frequent patterns without candidate generations', In Proceeding of the ACM SIGMOD, Page(s): 1–12.

Han, J. , and Kamber, M. , (2006), 'Data mining concepts and techniques', Elsevier Inc., Second Edition, San Francisco.

Han J., and Beheshti M., (2006) , 'Discovering both positive and negative fuzzy association rules in large transaction databases', Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.10, No.3 Page(s):287-294.

Han J.-W. , Pei J., and Yan X.-F., (2004), 'From sequential pattern mining to structured pattern mining: a pattern-growth approach', Journal of Computer Science and Technology, Vol.19, No. 3, Page(s): 257-279.

Houtsma M. and Swami A., (1995), 'Set-oriented mining for association rules in relational database', In Proceeding of the 11th International Conference on Data Engineering, Pages: 25-33.

Kumar A. V. S. and Wahidabanu R. S. D., (2007), 'Discovery of frequent itemsets: frequent item tree-based approach', ITB Journal of Information and Communication Technology, Vol. 1C, Page(s): 42-55.

Larose D. T., (2006), 'Data mining methods and models', John Wiley & Sons, Inc. Hoboken, New Jersey.

Lee A. J. T., and Wang C.-S., (2007), 'An efficient algorithm for mining frequent inter-transaction patterns', Information Sciences: an International Journal, Vol. 177, Issue 17, Page(s): 3453-3476.

Li J. and Ye X. , (2007), 'Study on linked list-based algorithm for metarule-guided mining of multidimensional quantitative association rules', In Proceeding of 3rd International Conference on Natural Computation, IEEE CNF, Vol. 1, Page(s):300 - 304.

Liu, G., Lu, H., Yu, J. X., Wang, W., and Xiao, X., (2003), 'AFOPT: An efficient implementation of pattern growth approach', In Proceeding IEEE ICDM'03 Workshop (FIMI'03), Florida.

Luo D., Cao L., Luo C., Zhang C., and Wang W., (2008), 'Towards business interestingness in actionable knowledge discovery', IOS Press, Vol. 177, Page(s): 101–111.

Mahafzah B., Al-Badarneh A., and Zakaria M., (2009), 'A new sampling technique for association rule mining', Journal of Information Science, Vol. 35, No. 3, Page(s): 358-376.

Marcus A. , Maletic J. I., and Lin K.-I., (2001) , 'Ordinal association rules for error identification in data sets' , In Proceeding of the 10th International Conference on Information and Knowledge Management ,Pages:589-591.

Miller R. J. and Yang Y., (1997), 'Association rules over interval data', In Proceeding of ACM-SIGMOD International Conference on Management of Data, Pages: 452-461.

Nayak J. R. and Cook D. J., (2001), 'Approximate association rule mining', In Proceedings of the Florida Artificial Intelligence Research Symposium (FLAIRS-14), Key West.

Ni J. , Cao L. , and Zhang C. , (2008), 'Evolutionary optimization of trading strategies', IOS Press, Vol. 177, Page(s): 13–26.

Padmanabhan B. and Tuzhilin A. , (2000), 'Small is beautiful: Discovering the minimal set of unexpected patterns', In Proceeding of 6th ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD), Page(s): 54-63.

Pei J. , Han J. , Mortazavi-Asl B. , Wang J. , Pinto H. , Chen Q. , Dayal U. and Hsu M. , (2004), 'Mining sequential patterns by pattern-growth: the PrefixSpan approach', Knowledge and Data Engineering Journal, IEEE Transactions on Vol. 16, Issue 11, Page(s): 1424 – 1440.

Pujari A. K., (2001), 'Data mining techniques', Universities Press (India) Private Limited.

Rácz B., (2004), 'Nonordfp: An FP-growth variation without rebuilding the FP-tree', 2nd International Workshop on Frequent Itemset Mining Implementations (FIMI04), Brighton.

Savasere A. , Omiecinski E. and Navathe S. , (1995), 'An efficient algorithm for mining association rules in large databases', In Proceedings of 21th International Conference on Very Large Data Bases, Page(s): 432-444.

Sharif M. N. A., Bahari M., Bakri A. and Zakaria N. H., (2005), 'Using a priori for supporting e-commerce system', Journal of Information Technology Impact, Vol. 5, No. 3, Page(s): 129-138.

Sharma L.K., Vyas O.P., Tiwary U. S. and Vyas R. , (2005), 'A novel approach of multilevel positive and negative association rule mining for spatial databases' , Springer-Verlag Berlin Heidelberg, MLDM 2005, LNAI 3587, Page(s): 620 – 629.

Son N. Nguyen and Maria E. Orlowska, (2005), 'Improvements in the data partitioning approach for frequent itemsets mining', Springer-Verlag Berlin Heidelberg, Vol. 3721/2005,Page(s): 625-633.

Srikant R., and Agrawal R. ,(1996),' Mining quantitative association rules in large relational tables', In Proceeding of Association for Computing Machinery-Special Interest Group on Management of Data (ACM SIGMOD), Page(s): 1-12.

Tan P.-N., Steinbach M., and Kumar V., (2006), 'Introduction to data mining', Addison Wesley.

Taniar D., Rahayu W., Lee V., and Daly O. , (2008), 'Exception rules in association rule mining', Applied Mathematics and Computation Journal, Vol. 205, Issue 2, Page(s): 735-750.

Toivonen H., (1996), 'Sampling large databases for association rules', In Proceeding of International Conference Very Large Data Bases (VLDB'96), Page(s): 134–145,Bombay, India.

Tsai F. S., Chen Y., and Chan K. L., (2008), 'Probabilistic latent semantic analysis for search and mining of corporate blogs', IOS Press, Vol. 177, Page(s): 65–75.

Tsai P. S. M., Chen C.-M., (2001),'Mining quantitative association rules in a large database of sales transactions' , Journal of Information Science and Engineering – JISE, Vol. 17, No. 4 : Page(s): 667-681.

Wang Y. and HU X., (2004), 'A fast algorithm for mining association rules based on concept lattice', In Proceeding of International Conference on Machine Learning and Cybernetics Vol. 3, Page(s): 1687 - 1691.

Wang H., Wang W., Yang J. and Yu P. S.,(2002), 'Clustering by pattern similarity in large data sets', In Proceedings of the 2002 ACM SIGMOD International Conference on Management of data ,Page(s): 418-427.

Wu X., Zhang C. and Zhang S., (2004), 'Efficient mining of both positive and negative association rules', ACM Transactions on Information Systems, (TOIS Journal), Page(s): 381-405.

Wu C. , (2006), 'Applying frequent itemset mining to identify a small itemset that satisfies a large percentage of orders in a warehouse', Computers and Operations Research Journal, Page(s): Vol. 33: Page(s): 3161-3170.

Wur S.-Y., Leu Y., (1999), 'An effective boolean algorithm for mining association rules in large databases', In Proceedings of the 6th International Conference on Database Systems for Advanced Applications (DASFAA '99), Page(s):179.

Xu Y. and Li Y. , (2007), 'Generating concise association rules', In Proceeding of the 16th ACM Conference on information and knowledge management, Page(s): 781-790.

Xu W. , and Wang R., (2006), 'A novel algorithm of mining multidimensional association rules', Springer Berlin / Heidelberg, Volume 344/2006, Page(s): 771-777.

Yongmei L. and Yong G.,(2009), 'Application in market basket research based on fp-growth algorithm', In Proceeding of WRI World Congress on Computer Science and Information Engineering, Vol. 4, Page(s): 112-115.

Zaki M. J., (2004), 'Mining non-redundant association rules', Data Mining and Knowledge Discovery: An International Journal, DMKDJ'04, Vol. 9, Issue 3, Page(s): 223-248.

Zhang C. and Zhang S., (2002), 'Association rule mining. Models and algorithms', Springer Berlin / Heidelberg, Vol. 2307.

Zhou L. and Yau S., (2007), 'Association rule and quantitative association rule mining among infrequent items', In Proceeding of the 8th International Workshop on Multimedia Data Mining : (associated with the ACM SIGKDD 2007), Article No. 9.

APPENDIX 1: Whole set of frequent 1-itemsets

Itemsets (frequency)	Itemsets (frequency)	Itemsets (frequency)
892 (164)	848 (170)	5289 (176)
991 (164)	2182 (170)	0 (177)
2327 (164)	2876 (170)	2202 (177)
2853 (164)	3762 (170)	2471 (177)
3338 (164)	4552 (170)	3402 (177)
3420 (164)	11560 (170)	4113 (177)
3539 (164)	12491 (170)	1290 (178)
4389 (164)	958 (171)	1805 (178)
4886 (164)	2026 (171)	1905 (178)
58 (165)	2720 (171)	2493 (178)
1439 (165)	5346 (171)	3312 (178)
1568 (165)	10598 (171)	4143 (178)
2084 (165)	1612 (172)	4221 (178)
2269 (165)	2424 (172)	5323 (178)
2721 (165)	4424 (172)	6403 (178)
3297 (165)	444 (173)	368 (179)
4340 (165)	986 (173)	572 (179)
6404 (165)	1781 (173)	1166 (179)
12996 (165)	2129 (173)	2822 (179)
265 (166)	2514 (173)	4643 (179)
460 (166)	3271 (173)	10128 (179)
770 (166)	4771 (173)	1615 (180)
1981 (166)	5178 (173)	2708 (180)
4080 (166)	364 (174)	2840 (180)
4226 (166)	445 (174)	3240 (180)
10441 (166)	683 (174)	4859 (180)
12921 (166)	1178 (174)	10481 (180)
1130 (167)	1215 (174)	228 (181)
1180 (167)	1727 (174)	289 (181)
1354 (167)	7646 (174)	1184 (181)
2122 (167)	467 (175)	1588 (181)
2919 (167)	791 (175)	2800 (181)
3742 (167)	877 (175)	2812 (181)
4548 (167)	1896 (175)	3988 (181)
3272 (168)	1951 (175)	4913 (181)
3314 (168)	4310 (175)	13556 (181)
3840 (168)	236 (176)	184 (182)
2408 (169)	1736 (176)	781 (182)
2820 (169)	2210 (176)	3161 (182)
2967 (169)	2654 (176)	4342 (182)
3973 (169)	3236 (176)	4744 (182)
5339 (169)	4843 (176)	421 (183)

1640 (183)
2417 (183)
2861 (183)
3343 (183)
3855 (183)
4472 (183)
5976 (183)
1603 (184)
1972 (184)
2810 (184)
4360 (184)
6190 (184)
621 (185)
636 (185)
698 (185)
1884 (185)
2058 (185)
2669 (185)
4357 (185)
5127 (185)
207 (186)
2476 (186)
2519 (186)
2907 (186)
4111 (186)
4386 (186)
13049 (186)
129 (187)
807 (187)
966 (187)
981 (187)
15798 (187)
1430 (188)
1791 (188)
2064 (188)
67 (189)
584 (189)
874 (189)
973 (189)
4127 (189)
4251 (189)
14634 (189)
763 (190)
1428 (190)
2793 (190)
5651 (190)
197 (191)
813 (191)
1055 (191)
1598 (191)
1776 (191)
1811 (191)
2855 (191)
64 (192)
573 (192)
1115 (192)
1315 (192)
1517 (192)
1739 (192)
1837 (192)
2004 (192)
3086 (192)
3467 (192)
4144 (192)
13056 (192)
325 (193)
695 (193)
731 (193)
1192 (193)
1390 (193)
2240 (193)
2378 (193)
3202 (193)
85 (194)
331 (194)
593 (194)
1433 (194)
1592 (194)
2842 (194)
6630 (194)
51 (195)
274 (195)
1642 (195)
1697 (195)
2891 (195)
3504 (195)
10423 (195)
44 (196)
1408 (196)
1696 (196)
1929 (196)
10493 (196)
11794 (196)
59 (197)
968 (197)
1159 (197)
1537 (197)
2480 (197)
5108 (197)
83 (198)
3276 (198)
3324 (198)
3896 (198)
5968 (198)
828 (199)
2239 (199)
3551 (199)
6602 (199)
22 (200)
127 (200)
755 (200)
2150 (200)
1023 (201)
1821 (201)
3537 (201)
3638 (201)
243 (202)
485 (202)
500 (202)
555 (202)
1063 (202)
2263 (202)
3185 (202)
9617 (202)
11254 (202)
566 (203)
1773 (203)
2389 (203)
10474 (203)
13060 (203)
193 (204)
314 (204)
2469 (204)
3535 (204)
861 (205)
3005 (205)
3047 (205)
5358 (205)
211 (206)
434 (207)
713 (207)

1796 (207)
2351 (207)
2117 (208)
2621 (208)
939 (209)
1380 (209)
1771 (209)
1772 (209)
2097 (209)
3412 (209)
4105 (209)
382 (210)
706 (210)
711 (210)
1060 (210)
1233 (210)
2537 (210)
2554 (210)
3756 (210)
40 (211)
141 (211)
596 (211)
558 (212)
1016 (212)
1342 (212)
2663 (212)
5767 (212)
739 (213)
1387 (213)
2506 (213)
4891 (213)
11840 (213)
1161 (214)
1343 (214)
1789 (214)
2002 (214)
3055 (214)
3319 (214)
6118 (214)
8730 (214)
68 (215)
2413 (215)
2788 (215)
3021 (215)
3186 (215)
5782 (215)
10611 (215)

639 (216)
4975 (216)
11747 (216)
10939 (217)
442 (218)
1025 (218)
1459 (218)
2286 (218)
43 (219)
1275 (219)
2011 (219)
3735 (219)
3808 (219)
4201 (219)
13043 (220)
402 (221)
493 (221)
1538 (221)
2147 (221)
3966 (221)
583 (222)
880 (222)
1593 (222)
1678 (222)
1744 (222)
2805 (222)
4720 (222)
199 (223)
2599 (223)
337 (224)
529 (224)
660 (224)
2394 (224)
3006 (224)
9843 (224)
171 (225)
525 (225)
896 (225)
1964 (225)
9669 (225)
1345 (226)
1818 (226)
2113 (226)
3724 (227)
222 (228)
406 (228)
650 (228)

1527 (228)
1652 (228)
3149 (228)
10818 (228)
1815 (229)
2706 (229)
4940 (229)
220 (230)
1012 (230)
2052 (230)
2746 (230)
3292 (230)
9001 (230)
10074 (230)
13182 (230)
447 (231)
486 (231)
851 (231)
2728 (231)
2983 (231)
3221 (231)
98 (232)
151 (232)
1071 (232)
1072 (232)
3553 (232)
282 (233)
2874 (233)
2990 (233)
6128 (233)
108 (234)
849 (234)
4630 (234)
399 (235)
440 (235)
599 (235)
759 (235)
887 (235)
1085 (235)
12677 (235)
4170 (236)
565 (237)
911 (237)
1249 (237)
1476 (237)
2103 (237)
4316 (237)

2188 (238)
9304 (238)
200 (239)
392 (239)
606 (239)
1395 (239)
4209 (239)
926 (240)
1994 (240)
3683 (240)
1784 (241)
3311 (241)
2634 (242)
760 (243)
1332 (243)
2528 (243)
3404 (243)
4339 (243)
104 (244)
708 (244)
491 (244)
1378 (244)
1594 (244)
2350 (244)
3291 (244)
3321 (244)
15345 (244)
133 (245)
69 (246)
651 (247)
1104 (247)
1160 (247)
1591 (247)
350 (248)
388 (248)
767 (248)
2276 (248)
2998 (248)
50 (249)
1318 (249)
1609 (249)
2177 (249)
2815 (249)
77 (250)
256 (250)
2441 (250)
597 (251)
795 (251)
1804 (251)
1819 (251)
2186 (251)
2561 (251)
924 (252)
1897 (252)
2406 (252)
3102 (252)
925 (253)
1143 (253)
1372 (253)
1497 (253)
2723 (253)
10568 (253)
1595 (254)
5728 (254)
576 (255)
987 (255)
2843 (255)
167 (256)
1026 (256)
1163 (256)
3451 (256)
206 (257)
837 (257)
1966 (257)
2225 (257)
647 (258)
1899 (258)
3189 (258)
4307 (258)
189 (259)
489 (259)
492 (259)
552 (259)
612 (259)
3160 (259)
10513 (259)
550 (260)
757 (260)
1013 (260)
1831 (260)
2259 (260)
2420 (260)
2915 (260)
587 (261)
749 (261)
753 (261)
1014 (261)
2212 (261)
4344 (261)
205 (262)
344 (262)
536 (262)
932 (262)
3151 (262)
3462 (262)
308 (263)
414 (263)
673 (263)
3012 (263)
347 (264)
1629 (264)
1786 (264)
2988 (264)
3635 (264)
1938 (265)
10448 (265)
1 (266)
116 (266)
1010 (266)
1903 (266)
2167 (266)
2370 (266)
8867 (267)
285 (268)
2856 (268)
12473 (268)
563 (270)
1616 (270)
4685 (270)
10605 (270)
95 (271)
1096 (271)
2053 (271)
1602 (270)
244 (272)
1232 (272)
5124 (272)
5248 (272)
2997 (273)
165 (274)
615 (274)

805 (274)
3315 (274)
1741 (275)
2573 (275)
4945 (275)
10420 (275)
128 (276)
169 (276)
250 (276)
1188 (276)
684 (277)
1291 (277)
4070 (278)
1269 (279)
3279 (279)
611 (280)
764 (280)
2344 (280)
2596 (280)
1976 (281)
2625 (281)
511 (283)
1280 (283)
1379 (283)
1564 (283)
234 (284)
412 (284)
2635 (285)
2965 (285)
10551 (285)
662 (286)
3194 (286)
3668 (286)
82 (287)
718 (287)
907 (287)
1001 (287)
4899 (287)
423 (288)
2388 (288)
4642 (288)
5405 (288)
1282 (289)
3294 (289)
5729 (289)
5729 (289)
938 (290)
9579 (291)
1763 (292)
168 (293)
1590 (293)
2164 (293)
2464 (293)
8068 (293)
5074 (294)
5 (295)
543 (295)
1080 (295)
1172 (295)
682 (296)
682 (296)
1842 (296)
2440 (296)
4994 (296)
2054 (297)
5152 (297)
504 (298)
1281 (298)
4779 (299)
166 (300)
697 (300)
1939 (300)
2375 (300)
2425 (300)
3316 (300)
927 (301)
2364 (301)
2749 (301)
14386 (301)
989 (302)
1956 (302)
10451 (302)
4207 (303)
12981 (303)
213 (304)
320 (305)
278 (306)
1991 (306)
227 (307)
397 (307)
235 (309)
309 (309)
2792 (309)
1034 (310)
2524 (310)
1333 (311)
1655 (311)
10943 (311)
3759 (312)
1808 (314)
204 (316)
534 (316)
2191 (316)
3317 (316)
262 (317)
622 (317)
2629 (318)
66 (319)
342 (319)
501 (319)
2075 (319)
215 (320)
433 (321)
921 (322)
1529 (322)
75 (324)
1209 (324)
2495 (324)
3964 (324)
4524 (324)
13032 (324)
652 (325)
1196 (325)
12982 (325)
13174 (325)
203 (327)
1698 (327)
2215 (327)
751 (328)
15578 (328)
769 (329)
4393 (329)
164 (330)
3827 (330)
13334 (330)
1185 (332)
2091 (332)
96 (334)
984 (334)
2894 (334)
346 (335)

643 (336)
195 (337)
1081 (337)
1360 (337)
4650 (337)
10525 (338)
653 (339)
1444 (339)
1638 (339)
2325 (339)
2633 (339)
7975 (339)
10442 (340)
1425 (341)
12933 (342)
1308 (343)
2468 (343)
993 (344)
15 (345)
2065 (345)
497 (346)
610 (346)
1027 (346)
2254 (347)
16431 (351)
424 (352)
829 (352)
3510 (352)
2626 (353)
768 (354)
640 (359)
5051 (363)
1046 (365)
232 (366)
734 (366)
7205 (366)
730 (369)
785 (369)
817 (369)
2991 (369)
8691 (369)
978 (371)
1973 (371)
2552 (371)
1619 (372)
1562 (374)
2187 (374)

370 (375)
390 (376)
903 (376)
2015 (376)
1082 (379)
1930 (379)
3502 (381)
715 (382)
10490 (382)
727 (383)
1257 (383)
2353 (383)
1003 (387)
2515 (388)
12935 (390)
840 (391)
1214 (391)
1859 (391)
1313 (392)
2492 (392)
341 (396)
13443 (396)
2208 (399)
1361 (400)
1052 (401)
12951 (403)
176 (404)
246 (405)
1103 (407)
12 (409)
1481 (409)
3361 (409)
14099 (409)
2775 (410)
1066 (411)
2673 (411)
248 (412)
415 (412)
694 (412)
842 (412)
8985 (414)
1557 (415)
10656 (415)
12943 (417)
10491 (418)
10653 (418)
1543 (419)

2056 (419)
351 (420)
1277 (420)
2241 (420)
3799 (421)
9555 (421)
1513 (422)
1000 (423)
1486 (423)
1121 (425)
1126 (425)
2343 (425)
2523 (426)
16430 (426)
514 (427)
547 (428)
830 (428)
1158 (428)
9501 (430)
2987 (431)
591 (432)
1062 (433)
2505 (433)
319 (435)
384 (437)
352 (438)
13189 (438)
365 (442)
47 (444)
490 (445)
1404 (446)
1113 (448)
1585 (450)
1704 (450)
281 (453)
831 (454)
1659 (459)
2051 (461)
2135 (461)
2184 (462)
80 (464)
1020 (465)
432 (466)
136 (467)
1867 (467)
1677 (468)
1479 (472)

10444 (473)
12959 (476)
178 (479)
408 (479)
14933 (479)
418 (481)
2080 (481)
2399 (481)
2879 (481)
808 (482)
1987 (482)
2329 (488)
345 (490)
793 (494)
856 (494)
2168 (499)
52 (501)
798 (503)
1714 (507)
426 (508)
2284 (508)
53 (509)
1355 (512)
571 (513)
398 (515)
94 (516)
855 (519)
910 (519)
4883 (521)
261 (523)
790 (523)
1693 (524)
297 (525)
623 (525)
2118 (525)
704 (526)
208 (529)
56 (535)
12946 (535)
1417 (536)
30 (540)
961 (540)
62 (543)
913 (543)
916 (543)
155 (546)
2 (549)

1043 (551)
1239 (569)
2383 (569)
878 (570)
2437 (571)
1872 (572)
581 (574)
885 (576)
267 (584)
374 (584)
1067 (584)
4336 (589)
1715 (590)
947 (591)
846 (595)
150 (597)
2199 (597)
1144 (599)
420 (600)
1783 (600)
675 (601)
425 (606)
8978 (606)
441 (609)
209 (611)
1986 (614)
15618 (614)
10446 (618)
251 (621)
2046 (625)
809 (626)
865 (632)
279 (633)
340 (643)
789 (648)
272 (650)
3616 (654)
105 (657)
186 (660)
1198 (660)
1135 (667)
16011 (669)
396 (674)
379 (677)
259 (684)
1814 (688)
449 (690)

260 (691)
407 (693)
464 (695)
269 (701)
55 (704)
681 (704)
11 (711)
10 (712)
535 (718)
488 (720)
806 (741)
12929 (742)
664 (752)
772 (752)
1344 (753)
23 (758)
107 (765)
1435 (765)
334 (774)
371 (782)
570 (792)
766 (797)
1600 (801)
703 (806)
103 (812)
976 (817)
229 (820)
812 (821)
156 (830)
389 (831)
549 (843)
544 (851)
649 (857)
18 (860)
76 (869)
1578 (874)
405 (879)
10515 (882)
264 (895)
2958 (904)
45 (911)
242 (911)
956 (911)
31 (920)
479 (926)
3270 (950)
783 (965)

175 (970)
522 (974)
258 (987)
179 (998)
19 (1005)
161 (1010)
117 (1026)
13041 (1051)
78 (1060)
37 (1074)
1004 (1102)
677 (1110)
589 (1119)
49 (1120)
201 (1133)
548 (1137)
15832 (1143)
249 (1160)
1393 (1161)
16217 (1166)
740 (1181)
286 (1183)
301 (1204)
604 (1209)
824 (1210)
592 (1227)
338 (1274)
14098 (1291)
123 (1302)
16010 (1316)
9 (1372)
185 (1376)
1146 (1426)
12925 (1467)
255 (1474)
533 (1487)
60 (1489)
79 (1600)
2238 (1715)
270 (1734)
147 (1779)
1327 (1786)
438 (1863)
413 (1880)
271 (2094)
475 (2167)
101 (2237)
310 (2594)
110 (2794)
36 (2936)
237 (3032)
170 (3099)
225 (3257)
89 (3837)
65 (4472)
41 (14945)
32 (15167)
38 (15596)
48 (42135)
39 (50675)

APPENDIX 2: Whole set of frequent 2-itemsets

Itemsets (frequency)	Itemsets (frequency)	Itemsets (frequency)
3005 38 (195)	414 39 (188)	2573 39 (178)
4975 39 (164)	673 39 (172)	4945 48 (177)
2805 38 (212)	3012 39 (165)	10420 48 (178)
2113 39 (171)	347 48 (177)	10420 39 (183)
1996 39 (174)	1629 (264)	128 48 (170)
10074 39 (165)	1629 48 (165)	169 39 (177)
887 48 (164)	1786 (264)	250 48 (178)
1476 39 (168)	1786 48 (168)	1188 38 (256)
3311 1819 (180)	1786 39 (206)	1188 39 (176)
491 48 (182)	2763 39 (164)	684 48 (177)
491 39 (186)	3635 48 (166)	684 39 (168)
708 39 (178)	3635 39 (168)	1291 39 (170)
1104 39 (167)	1183 48 (179)	1294 48 (166)
1591 39 (169)	1938 39 (169)	1294 39 (167)
767 48 (164)	116 39 (166)	4070 48 (175)
2998 48 (169)	1010 48 (165)	4070 39 (177)
2998 39 (168)	1010 39 (193)	1269 39 (194)
256 39 (188)	2167 48 (181)	3279 39 (181)
597 39 (209)	2167 39 (181)	611 39 (165)
795 1819 (167)	285 48 (185)	764 39 (194)
1804 39 (164)	2856 48 (176)	2344 48 (180)
924 48 (164)	2856 39 (170)	2344 39 (183)
925 39 (165)	12473 48 (166)	1976 48 (177)
1143 39 (178)	12473 39 (172)	1976 39 (184)
5728 39 (191)	1602 48 (165)	2625 48 (166)
576 39 (170)	1602 39 (176)	1280 48 (165)
167 48 (164)	1616 39 (169)	1280 39 (169)
1026 39 (168)	4685 48 (178)	1379 309 (166)
1163 48 (177)	4685 39 (166)	1379 39 (196)
1966 39 (184)	10605 48 (179)	1564 48 (182)
1899 39 (174)	10605 39 (191)	234 39 (185)
4307 48 (168)	95 39 (189)	412 48 (181)
489 48 (165)	2053 48 (176)	412 39 (188)
492 39 (164)	1232 48 (179)	2635 39 (178)
552 48 (174)	5124 48 (170)	2965 48 (179)
757 39 (184)	5248 39 (175)	2965 39 (179)
1831 38 (252)	2997 39 (164)	10551 48 (168)
2259 39 (175)	165 48 (195)	10551 39 (178)
2915 39 (164)	615 48 (170)	662 39 (169)
4344 39 (184)	805 48 (171)	3668 48 (180)
536 48 (164)	805 39 (193)	3668 39 (174)
932 39 (166)	3315 39 (175)	82 48 (177)
308 39 (170)	1741 39 (194)	82 39 (165)

718 48 (170)
718 39 (180)
907 39 (194)
1001 48 (183)
4899 39 (203)
423 39 (164)
2388 48 (178)
2388 39 (223)
5405 48 (170)
5405 39 (173)
1282 48 (173)
1282 39 (195)
3294 39 (198)
5729 48 (192)
5729 39 (216)
938 48 (168)
938 39 (165)
9579 48 (190)
9579 39 (192)
1763 48 (164)
1763 39 (165)
168 48 (189)
168 39 (208)
1590 38 (281)
1590 39 (199)
2164 48 (186)
2164 39 (188)
2464 48 (184)
2464 39 (186)
8068 48 (166)
8068 39 (202)
5074 48 (182)
5074 39 (184)
543 48 (172)
543 39 (175)
1080 48 (167)
1080 39 (183)
1172 48 (184)
1172 39 (176)
682 48 (191)
682 39 (188)
1842 48 (176)
1842 39 (210)
2440 48 (185)
2440 39 (199)
4994 48 (173)
4994 39 (188)
2054 48 (176)
2054 39 (188)
5152 39 (189)
504 38 (245)
504 39 (196)
1281 39 (178)
1489 39 (189)
4779 48 (172)
166 48 (183)
166 39 (180)
697 48 (165)
697 39 (200)
1939 48 (178)
1939 39 (181)
2375 48 (187)
2375 39 (184)
2425 39 (164)
3316 48 (169)
3316 39 (168)
927 48 (180)
927 39 (183)
2364 48 (179)
2364 39 (168)
2749 48 (166)
2749 39 (192)
14386 39 (169)
989 48 (189)
989 39 (198)
1956 48 (243)
1956 39 (194)
10451 48 (175)
10451 39 (168)
4207 48 (205)
4207 39 (171)
12981 39 (174)
213 39 (216)
320 48 (195)
320 39 (199)
278 48 (174)
278 39 (171)
1991 39 (225)
397 48 (210)
397 39 (198)
235 48 (184)
235 39 (197)
309 39 (211)
2792 48 (226)
2792 39 (225)
1034 769 (172)
2524 48 (213)
2524 39 (167)
1333 48 (171)
1333 39 (187)
1655 48 (197)
1655 39 (190)
10943 48 (204)
3759 48 (195)
3759 39 (219)
1808 48 (181)
1808 39 (183)
204 48 (190)
204 39 (191)
534 48 (203)
534 39 (215)
2191 48 (172)
2191 39 (182)
3317 48 (172)
3317 39 (179)
262 48 (186)
262 39 (192)
622 48 (191)
622 39 (177)
2629 48 (197)
2629 39 (198)
66 48 (168)
66 39 (180)
342 48 (193)
342 39 (179)
2075 48 (181)
2075 39 (210)
215 48 (206)
215 39 (215)
433 48 (173)
433 39 (197)
921 48 (183)
921 39 (188)
1529 38 (295)
1529 39 (223)
75 48 (209)
75 39 (220)
1209 48 (201)
1209 39 (220)
2495 48 (202)
2495 39 (196)

3964 48 (214)
3964 39 (225)
4524 48 (187)
4524 39 (193)
13032 48 (195)
13032 39 (186)
652 48 (187)
652 39 (205)
1196 48 (185)
1196 39 (192)
12982 48 (177)
12982 39 (190)
13174 48 (184)
13174 39 (197)
203 48 (204)
203 39 (199)
1698 48 (197)
1698 39 (192)
2215 48 (195)
2215 39 (164)
751 48 (202)
751 39 (193)
15578 48 (182)
15578 39 (233)
4393 48 (185)
4393 39 (191)
164 48 (224)
164 39 (179)
3827 48 (188)
3827 39 (239)
13334 48 (189)
13334 39 (172)
1185 48 (195)
1185 39 (205)
2091 48 (187)
2091 39 (205)
96 48 (191)
96 39 (224)
984 48 (205)
984 39 (226)
2894 48 (180)
2894 39 (182)
346 48 (216)
346 39 (205)
643 48 (212)
643 39 (208)
195 48 (178)
195 39 (217)
1081 48 (200)
1081 39 (210)
1360 48 (194)
1360 39 (204)
4650 48 (188)
4650 39 (184)
10525 48 (208)
10525 39 (206)
653 48 (195)
653 39 (192)
1444 48 (208)
1444 39 (206)
1638 48 (226)
1638 39 (223)
2325 48 (212)
2325 39 (208)
2633 48 (181)
2633 39 (190)
7975 48 (200)
7975 39 (186)
10442 48 (229)
10442 39 (202)
1425 48 (172)
1425 39 (202)
3904 38 (333)
3904 48 (200)
3904 39 (275)
1308 48 (191)
1308 39 (184)
2468 39 (213)
993 48 (201)
993 39 (197)
15 48 (210)
15 39 (208)
2065 48 (209)
2065 39 (235)
497 48 (176)
497 39 (200)
610 48 (196)
610 39 (182)
1027 48 (228)
1027 39 (221)
2254 48 (198)
2254 39 (216)
16431 16430 (348)
12933 48 (199)
12933 39 (227)
16431 48 (175)
16431 39 (189)
424 48 (218)
424 39 (201)
829 48 (200)
829 39 (214)
3510 48 (213)
3510 39 (225)
2626 48 (214)
2626 39 (199)
768 39 (188)
640 48 (232)
640 39 (227)
5051 48 (217)
5051 39 (216)
1046 32 (339)
1046 48 (190)
1046 39 (170)
232 48 (207)
232 39 (200)
734 48 (329)
734 39 (255)
7205 48 (209)
7205 39 (209)
730 48 (215)
730 39 (241)
785 48 (189)
785 39 (222)
817 48 (210)
817 39 (182)
2991 48 (189)
2991 39 (230)
8691 48 (288)
8691 39 (259)
978 48 (203)
978 39 (273)
1973 48 (236)
1973 39 (191)
2552 48 (221)
2552 39 (205)
1619 48 (236)
1619 39 (252)
1562 48 (206)
1562 39 (193)
2187 48 (233)
2187 39 (221)

370 38 (362)
370 48 (196)
370 39 (248)
2115 48 (237)
2115 39 (213)
390 38 (354)
390 48 (210)
390 39 (258)
903 48 (192)
903 39 (264)
2015 48 (259)
2015 39 (230)
1082 48 (212)
1082 39 (196)
1930 48 (215)
1930 39 (226)
3502 48 (214)
3502 39 (244)
715 48 (254)
715 39 (217)
10490 48 (240)
10490 39 (237)
727 48 (253)
727 39 (241)
1257 48 (248)
1257 39 (256)
2353 48 (215)
2353 39 (253)
1003 48 (250)
1003 39 (252)
2515 48 (218)
2515 39 (197)
12935 48 (302)
12935 39 (267)
840 38 (379)
840 48 (206)
840 39 (251)
1214 48 (195)
1214 39 (230)
1859 48 (210)
1859 39 (218)
1313 48 (250)
1313 39 (250)
2492 48 (238)
2492 39 (217)
341 48 (235)
341 39 (228)
13443 48 (217)
13443 39 (229)
2208 48 (222)
2208 39 (235)
1361 48 (215)
1361 39 (261)
1052 48 (266)
1052 39 (206)
12951 48 (219)
12951 39 (283)
176 48 (226)
176 39 (204)
246 48 (265)
246 39 (259)
4698 48 (251)
4698 39 (263)
1103 48 (250)
1103 39 (252)
12 48 (238)
12 39 (230)
1481 48 (236)
1481 39 (272)
3361 48 (229)
3361 39 (231)
14099 48 (233)
14099 39 (242)
2775 48 (239)
2775 39 (249)
1066 48 (250)
1066 39 (236)
2673 48 (280)
2673 39 (246)
248 48 (285)
248 39 (275)
415 48 (255)
415 39 (243)
694 48 (215)
694 39 (267)
842 48 (201)
842 39(263)
8985 48 (220)
8985 39 (276)
1557 48 (250)
1557 39 (252)
10656 48 (245)
10656 39 (227)
12943 48 (207)
12943 39 (253)
10491 48 (234)
10491 39 (257)
10653 48 (252)
10653 39 (246)
1543 48 (240)
1543 39 (290)
2056 48 (258)
2056 39 (257)
351 48 (221)
351 39 (220)
1277 48 (248)
1277 39 (241)
2241 48 (284)
2241 39 (285)
3799 48 (264)
3799 39 (284)
9555 48 (224)
9555 39 (240)
1513 48 (248)
1513 39 (284)
1000 48 (221)
1000 39 (245)
1486 48 (270)
1486 39 (256)
1121 48 (250)
1121 39 (257)
1126 48 (246)
1126 39 (246)
2343 48 (239)
2343 39 (233)
2523 48 (269)
2523 39 (247)
16430 48 (219)
16430 39 (221)
514 48 (293)
514 39 (271)
547 48 (252)
547 39 (254)
830 48 (294)
830 39 (256)
1158 48 (232)
1158 39 (242)
9501 48 (232)
9501 39 (275)
2987 48 (284)
2987 39 (238)

591 48 (280)
591 39 (289)
1062 48 (232)
1062 39 (268)
2505 2284 (194)
2505 48 (256)
2505 39 (297)
319 48 (261)
319 39 (270)
384 48 (247)
384 39 (279)
352 41 (170)
352 48 (246)
352 39 (224)
13189 48 (236)
13189 39 (266)
979 48 (266)
979 39 (332)
365 48 (259)
365 39 (288)
47 38 (432)
47 48 (274)
47 39 (320)
490 48 (279)
490 39 (279)
1404 48 (277)
1404 39 (273)
1113 48 (248)
1113 39 (248)
1585 48 (292)
1585 39 (262)
1704 48 (246)
1704 39 (287)
281 38 (432)
281 48 (269)
281 39 (288)
831 48 (311)
831 39 (314)
1659 48 (258)
1659 39 (307)
2051 48 (300)
2051 39 (308)
2135 48 (293)
2135 39 (317)
2184 48 (288)
2184 39 (294)
80 48 (271)
80 39 (316)
1020 48 (274)
1020 39 (284)
432 48 (251)
432 39 (286)
136 48 (286)
136 39 (306)
1867 48 (275)
1867 39 (274)
1677 48 (306)
1677 39 (291)
1479 48 (273)
1479 39 (302)
10444 48 (302)
10444 39 (293)
12959 48 (228)
12959 39 (262)
178 48 (246)
178 39 (313)
408 48 (279)
408 39 (278)
14933 48 (274)
14933 39 (320)
418 48 (267)
418 39 (293)
2080 48 (285)
2080 39 (281)
2399 48 (276)
2399 39 (291)
2879 48 (276)
2879 39 (292)
808 48 (267)
808 39 (283)
1987 48 (298)
1987 39 (309)
2329 48 (324)
2329 39 (270)
345 48 (316)
345 39 (300)
793 48 (304)
793 39 (312)
856 48 (271)
856 39 (291)
2168 48 (265)
2168 39 (301)
52 48 (326)
52 39 (320)
798 48 (329)
52 39 (320)
798 48 (329)
1714 39 (317)
426 48 (327)
426 39 (334)
2284 48 (292)
2284 39 (349)
53 48 (322)
53 39 (351)
1355 175 (177)
1355 48 (260)
1355 39 (313)
571 48 (318)
571 39 (322)
398 48 (320)
398 39 (320)
94 48 (293)
94 39 (300)
855 48 (306)
855 39 (325)
910 48 (313)
910 39 (322)
4883 48 (305)
4883 39 (313)
261 48 (330)
261 39 (349)
790 41 (204)
790 38 (508)
790 48 (315)
790 39(364)
1693 48 (316)
1693 39 (347)
297 48 (328)
297 39 (356)
623 48 (310)
623 39 (315)
2118 48 (341)
2118 39 (311)
704 48 (307)
704 39 (279)
208 41 (210)
208 48 (330)
208 39 (326)
56 38 (514)
56 48 (309)
56 39 (364)

12946 48 (286)
12946 39 (339)
1417 48 (315)
1417 39 (321)
30 48 (298)
30 39 (344)
961 48 (327)
961 39 (331)
62 48 (314)
62 39 (281)
913 48 (320)
913 39 (336)
916 48 (311)
916 39 (350)
155 48 (314)
155 39 (347)
2 48 (268)
2 39 (324)
1043 48 (306)
1043 39 (342)
1239 48 (346)
1239 39 (350)
2383 48 (356)
2383 39 (377)
878 48 (310)
878 39 (344)
2437 48 (346)
2437 39 (308)
1872 48 (333)
1872 39 (331)
581 48 (353)
581 39 (368)
885 48 (307)
885 39 (354)
267 48 (368)
267 39 (407)
374 48 (340)
374 39 (376)
1067 107 (171)
1067 48 (369)
1067 39 (332)
4336 48 (330)
4336 39 (345)
1715 41 (186)
1715 48 (350)
1715 39 (356)
947 48 (324)

947 39 (324)
846 41 (180)
846 48 (297)
846 39 (328)
150 48 (353)
150 39 (364)
2199 48 (383)
2199 39 (349)
1144 249 (189)
1144 48 (336)
1144 39 (344)
420 48 (311)
420 39 (340)
1783 48 (335)
1783 39 (372)
675 48 (374)
675 39 (414)
425 48 (408)
425 39 (396)
8978 16010 (195)
8978 41 (290)
8978 48(344)
8978 39 (433)
441 48 (341)
441 39 (393)
209 48 (312)
209 39 (323)
1986 48 (346)
1986 39 (413)
15618 41 (179)
15618 48 (323)
15618 39 (352)
10446 48 (345)
10446 39 (368)
251 48 (338)
251 39 (349)
2046 48 (305)
2046 39 (333)
809 48 (373)
809 39 (404)
865 48 (390)
865 39 (382)
279 48 (383)
279 39 (401)
340 41 (165)
340 48 (407)
340 39 (396)

789 41 (192)
789 48 (421)
789 39 (447)
272 48 (380)
272 39 (446)
3616 1146 (293)
3616 41 (204)
3616 48 (376)
3616 39 (419)
105 38 (643)
105 48 (372)
105 39 (455)
186 48 (417)
186 39 (413)
1198 41 (214)
1198 48 (376)
1198 39 (407)
1135 32 (193)
1135 48 (541)
1135 39 (458)
16011 16010 (651)
16011 41 (244)
16011 48 (374)
16011 39 (427)
396 41 (180)
396 48 (378)
396 39 (436)
379 48 (348)
379 39 (429)
259 48 (380)
259 39 (374)
1814 48 (451)
1814 39 (471)
449 48 (382)
449 39 (372)
260 48 (379)
260 39 (428)
407 48 (422)
464 39 (412)
269 48 (439)
269 39 (441)
55 41 (175)
55 38 (657)
55 48 (325)
55 39 (452)
681 48 (378)
681 39 (422)

11 41 (197)
11 48 (463)
11 39 (472)
10 48 (427)
10 39 (452)
535 41 (173)
535 48 (409)
535 39 (400)
488 48 (432)
488 39 (389)
806 175(195)
806 41 (176)
806 38 (216)
806 48 (396)
806 39 (493)
12929 32 (171)
12929 48 (430)
12929 39 (470)
664 38 (172)
664 48 (387)
664 39 (481)
772 32 (166)
772 48 (442)
772 39 (406)
1344 41 (258)
1344 32 (172)
1344 38 (188)
1344 48 (427)
1344 39 (485)
23 41 (182)
23 48 (416)
23 39 (512)
107 549 (181)
107 41 (176)
107 48 (467)
107 39 (426)
1435 48 (433)
1435 39 (459)
334 48 (443)
334 39 (506)
371 41 (172)
371 38 (767)
371 48 (369)
371 39 (532)
570 41 (204)
570 48 (514)
570 39 (558)

766 41 (173)
766 48 (489)
766 39 (470)
1600 41 (197)
1600 32 (166)
1600 48 (444)
1600 39 (474)
703 48 (467)
703 39 (453)
103 41 (172)
103 32 (191)
103 38 (207)
103 48 (431)
103 39 (512)
976 117 (312)
976 41 (184)
976 48 (456)
976 39 (471)
229 41 (188)
229 32 (190)
229 38 (196)
229 48 (545)
229 39 (503)
812 32 (182)
812 48 (463)
812 39 (471)
156 38 (178)
156 48 (473)
156 39 (582)
389 41 (191)
389 32 (197)
389 38 (211)
389 48 (422)
389 39 (544)
549 41 (196)
549 32(198)
549 48 (513)
549 39 (492)
544 41 (197)
544 48 (504)
544 39 (524)
649 41 (197)
649 48 (490)
649 39 (531)
18 41 (198)
18 32 (184)
18 48 (465)

18 39 (540)
76 41 (217)
76 48 (485)
76 39 (529)
1578 41 (170)
1578 32 (174)
1578 38 (169)
1578 48 (469)
1578 39 (576)
405 10515 (294)
405 32 (165)
405 48 (525)
405 39 (500)
10515 41 (172)
10515 48 (524)
10515 39 (528)
264 38 (251)
264 48 (534)
264 39 (599)
2958 41 (217)
264 41 (204)
2958 32 (189)
2958 38 (185)
2958 48 (779)
2958 39 (655)
45 41 (172)
45 38 (165)
45 48 (492)
45 39 (559)
242 38 (200)
242 48 (399)
242 39 (564)
956 41 (225)
956 32 (209)
956 48 (540)
956 39 (600)
31 41 (207)
31 32 (185)
31 38 (165)
31 48 (493)
31 39 (538)
479 41 (203)
479 32 (167)
479 48 (513)
479 39 (580)
3270 41 (214)
3270 32 (175)

3270 38 (174)
3270 48 (592)
3270 39 (576)
783 41 (201)
783 32 (194)
783 48 (583)
783 39 (640)
175 41 (293)
175 32 (198)
175 38 (272)
175 48 (493)
175 39 (631)
522 41 (214)
522 32 (180)
522 48 (582)
522 39 (648)
258 41 (254)
258 32 (194)
258 48 (643)
258 39 (628)
179 41 (224)
179 32 (228)
179 48 (624)
179 39 (653)
19 41 (202)
19 32 (180)
19 48 (591)
19 39 (592)
161 41 (210)
161 32 (188)
161 38 (218)
161 48 (517)
161 39 (622)
117 32 (193)
117 48 (601)
117 41 (264)
117 39 (554)
13041 41 (390)
13041 32 (225)
13041 38 (183)
13041 48 (602)
13041 39 (688)
78 41 (201)
78 32 (232)
78 38 (219)
78 48 (824)
78 39 (774)
37 41 (247)
37 32 (186)
37 38 (1046)
37 48 (565)
37 39 (707)
1004 41 (198)
1004 32 (285)
1004 38 (177)
1004 48 (614)
1004 39 (583)
677 41 (229)
677 32 (266)
677 38 (173)
677 48 (666)
677 39 (635)
589 41 (217)
589 32 (228)
589 38 (170)
589 48 (628)
589 39 (708)
49 41 (220)
49 32 (205)
49 38 (224)
49 48 (843)
49 39 (768)
201 41 (253)
201 32 (184)
201 48 (674)
201 39 (669)
548 41 (242)
548 32 (280)
548 38 (206)
548 48 (658)
548 39 (663)
15832 41 (421)
15832 32 (231)
15832 48 (585)
15832 39 (718)
249 237 (175)
249 41 (258)
249 32 (216)
249 38 (269)
249 48 (675)
249 39 (752)
1393 41 (215)
1393 32 (231)
1393 38 (203)
1393 48 (669)
1393 39 (789)
16217 41 (433)
16217 32 (204)
16217 38 (193)
16217 48 (647)
16217 39 (728)
740 41 (206)
740 32 (167)
740 38 (221)
740 48 (520)
740 39 (759)
286 41 (268)
286 32 (229)
286 38 (1116)
286 48 (591)
286 39 (750)
301 41 (256)
301 32 (279)
301 48 (658)
301 39 (715)
604 41 (286)
604 32 (224)
604 38 (195)
604 48 (687)
604 39 (775)
824 41 (320)
824 32 (262)
824 38 (172)
824 48 (701)
824 39 (703)
592 41 (303)
592 32 (257)
592 38 (185)
592 48 (743)
592 39 (723)
338 41 (256)
338 32 (229)
338 38 (184)
338 48 (779)
338 39 (742)
14098 41 (170)
14098 32 (224)
14098 38 (181)
14098 48 (698)
123 32 (252)
123 38 (201)

123 48 (797)
123 39 (726)
16010 41 (467)
16010 32 (281)
16010 38 (221)
16010 48 (730)
16010 39 (829)
9 41 (307)
9 32 (250)
9 38(198)
9 48 (790)
9 32 (250)
9 38(198)
9 48 (790)
185 32 (289)
185 38 (217)
185 48 (816)
185 39 (828)
1146 41 (501)
1146 32 (340)
1146 38 (306)
1146 48 (811)
1146 39 (983)
12925 32 (232)
12925 38 (294)
12925 48 (817)
12925 39 (938)
255 41 (332)
255 32 (321)
255 38 (267)
255 48(1057)
255 39 (1057)
533 41 (336)
533 32 (300)
533 38 (235)
533 48 (861)
533 39 (922)
60 65 (176)
60 41 (264)
60 32 (262)
60 38 (193)
60 48 (815)
60 39 (983)
79 41 (499)
79 32 (384)
79 38 (313)
79 48 (893)
79 39 (1111)
2238 270 (264)
2238 271 (282)
2238 225 (405)
2238 41 (570)
2238 32 (358)
2238 38 (407)
2238 48 (955)
2238 39 (1287)
270 271 (680)
270 225 (295)
270 41 (508)
270 32 (437)
270 38 (390)
270 48(957)
270 39 (1194)
147 41 (438)
147 32 (359)
147 38 (328)
147 48 (1036)
147 39 (1137)
1327 41 (459)
1327 32 (323)
1327 38 (347)
1327 48 (968)
1327 39 (1156)
438 41 (428)
438 32 (349)
438 38 (389)
438 48 (1025)
438 39 (1260)
413 101 (167)
413 65 (176)
413 41 (414)
413 32 (326)
413 38 (297)
413 48 (1135)
413 39 (1130)
271 225 (395)
271 65 (168)
271 41 (584)
271 32 (470)
271 38 (392)
271 48 (1090)
271 39 (1434)
475 237 (179)
475 65 (168)
475 41 (570)
475 32 (447)
475 38 (393)
475 48 (1428)
475 39 (1500)
101 89 (191)
101 65 (230)
101 41 (476)
101 32 (412)
101 38 (345)
101 48 (1311)
101 39 (1400)
310 237 (201)
310 225 (165)
310 89 (228)
310 65 (226)
310 41 (719)
310 32 (541)
310 38 (433)
310 48 (1692)
310 39 (1852)
110 36 (186)
110 41 (677)
110 32 (449)
110 38 (2725)
110 48 (1380)
110 39 (1759)
36 170 (214)
36 41 (700)
36 32 (494)
36 38 (2790)
36 48 (1416)
36 39 (2037)
237 89 (232)
237 65 (215)
237 41 (597)
237 32 (568)
237 38 (683)
237 48 (1682)
237 39 (1929)
170 89 (171)
170 41 (805)
170 32 (540)
170 38 (3031)
170 48 (1557)
170 39 (2059)
225 65 (213)

225 41 (877)		
225 32 (655)		
225 38 (681)		
225 48 (1736)		
225 39 (2351)		
89 65 (314)		
89 41 (732)		
89 32 (713)		
89 38 (764)		
89 48 (2798)		
89 39 (2749)		
65 41 (995)		
65 32 (774)		
65 38 (643)		
65 48 (2529)		
65 39 (2787)		
41 32 (3196)		
41 38 (3897)		
41 48 (9018)		
41 39 (11414)		
32 38 (2833)		
32 48 (8034)		
32 39 (8455)		
38 48 (7944)		
38 39 (10345)		
48 39 (29142)		
14098 39 (801)		
123 41 (236)		

APPENDIX 3: Whole set of frequent 3-itemsets

Itemsets (frequency)	Itemsets (frequency)	Itemsets (frequency)
1590 38 39 (193)	1066 48 39 (169)	281 48 39 (193)
504 38 39 (170)	2673 48 39 (190)	831 48 39 (237)
2792 48 39 (175)	248 48 39 (200)	1659 48 39 (194)
1529 38 39 (202)	415 48 39 (179)	2051 48 39 (222)
1638 48 39 (164)	8985 48 39 (168)	2135 48 39 (228)
3904 38 48 (193)	1557 48 39 (166)	2184 48 39 (210)
3904 38 39 (268)	10656 48 39 (167)	80 48 39 (208)
3904 48 39 (179)	10491 48 39 (176)	1020 48 39 (190)
1027 48 39 (165)	10653 48 39 (169)	432 48 39 (190)
16431 16430 48 (174)	1543 48 39 (189)	136 48 39 (219)
16431 16430 39 (186)	2056 48 39 (180)	1867 48 39 (187)
640 48 39 (169)	1277 48 39 (166)	1677 48 39 (211)
1046 32 48 (171)	2241 48 39 (214)	1479 48 39 (197)
734 48 39 (229)	3799 48 39 (206)	10444 48 39 (221)
730 48 39 (167)	1513 48 39 (196)	178 48 39(194)
8691 48 39 (213)	1000 48 39 (164)	408 48 39 (198)
978 48 39 (164)	1486 48 39 (181)	14933 48 39 (205)
1619 48 39 (174)	1121 48 39 (168)	418 48 39 (200)
370 38 48 (191)	1126 48 39 (165)	2080 48 39 (187)
370 38 39 (241)	2523 48 39 (187)	2399 48 39 (194)
2115 48 39 (168)	514 48 39 (201)	2879 48 39 (199)
390 38 48 (197)	547 48 39 (171)	808 48 39 (181)
390 38 39 (241)	830 48 39 (196)	1987 48 39 (217)
390 48 39 (165)	9501 48 39 (170)	2329 48 39 (214)
2015 48 39 (178)	2987 48 39 (186)	345 48 39 (212)
3502 48 39 (164)	591 48 39 (210)	793 48 39 (218)
715 48 39 (167)	1062 48 39 (168)	856 48 39 (192)
10490 48 39 (176)	2505 48 39 (201)	2168 48 39 (205)
727 48 39 (184)	319 48 39 (194)	52 48 39 (239)
1257 48 39 (193)	384 48 39 (175)	798 48 39 (230)
2353 48 39 (166)	13189 48 39 (173)	1714 48 39 (193)
1003 48 39 (182)	979 48 39 (222)	426 48 39 (250)
12935 48 39 (214)	365 48 39 (197)	2284 48 39 (226)
840 38 48 (202)	47 38 48 (269)	53 48 39 (253)
840 38 39 (245)	47 38 39 (313)	1355 48 39 (194)
1313 48 39 (179)	47 48 39 (216)	571 48 39 (240)
12951 48 39 (183)	490 48 39 (193)	398 48 39 (224)
246 48 39 (182)	1404 48 39 (202)	94 48 39 (209)
4698 48 39 (187)	1113 48 39 (171)	855 48 39 (228)
1103 48 39 (189)	1585 48 39 (191)	910 48 39 (222)
1481 48 39 (188)	1704 48 39 (180)	4883 48 39 (200)
14099 48 39 (172)	281 38 48 (259)	261 48 39 (244)
2775 48 39 (167)	281 38 39 (274)	790 41 38 (202)

790 38 48 (307)
790 38 39 (356)
790 48 39 (239)
1693 48 39 (232)
297 48 39 (242)
623 48 39 (213)
2118 48 39 (232)
704 48 39 (191)
208 41 39 (165)
208 48 39 (222)
56 38 48 (295)
56 38 39 (352)
56 48 39 (226)
12946 48 39 (212)
1417 48 39 (217)
30 48 39 (218)
961 48 39 (240)
62 48 39 (198)
913 48 39 (232)
916 48 39 (241)
155 48 39 (214)
2 48 39 (186)
1043 48 39 (229)
1239 48 39 (239)
2383 48 39 (270)
878 48 39 (231)
2437 48 39 (215)
1872 48 39 (213)
581 48 39 (264)
885 48 39 (222)
267 48 39 (292)
374 48 39 (250)
1067 48 39 (251)
4336 48 39 (230)
1715 48 39 (235)
947 48 39 (207)
846 48 39 (197)
150 48 39 (255)
2199 48 39 (266)
1144 48 39 (234)
420 48 39 (214)
1783 48 39 (240)
675 48 39 (291)
425 48 39 (299)
8978 41 48 (196)
8978 41 39 (243)
8978 48 39 (276)

441 48 39 (260)
209 48 39 (207)
1986 48 39 (264)
15618 48 39 (222)
10446 48 39 (246)
251 48 39 (220)
2046 48 39 (217)
809 48 39 (290)
865 48 39 (272)
279 48 39 (279)
340 48 39 (281)
789 48 39 (327)
272 48 39 (283)
3616 1146 48 (167)
3616 1146 39 (189)
3616 48 39 (270)
105 38 48 (367)
105 38 39 (449)
105 48 39 (291)
186 48 39 (305)
1198 41 39 (170)
1198 48 39 (277)
1135 48 39 (386)
16011 16010 41 (236)
16011 16010 48 (362)
16011 16010 39 (419)
16011 41 48 (165)
16011 41 39 (194)
16011 48 39 (275)
396 48 39 (278)
379 48 39 (253)
259 48 39 (255)
1814 48 39 (344)
449 48 39 (239)
260 48 39 (271)
407 48 39 (304)
464 48 39 (289)
269 48 39 (310)
55 41 38 (171)
55 38 48 (309)
55 38 39 (422)
55 48 39 (251)
681 48 39 (275)
11 41 39 (166)
11 48 39 (346)
10 48 39 (322)
535 48 39 (275)

488 48 39 (287)
806 48 39 (302)
12929 48 39 (320)
664 48 39 (296)
772 48 39 (275)
1344 41 48 (166)
1344 41 39 (213)
1344 48 39 (314)
23 48 39 (322)
107 48 39 (322)
1435 48 39 (307)
334 48 39 (333)
371 41 38 (171)
371 38 48 (365)
371 38 39 (526)
371 48 39 (295)
570 41 39 (173)
570 48 39 (404)
766 48 39 (335)
1600 48 39 (307)
703 48 39 (302)
103 48 39 (321)
976 117 48 (169)
976 48 39 (325)
229 48 39(382)
812 48 39 (310)
156 48 39 (377)
389 48 39 (323)
549 48 39 (355)
544 48 39 (348)
649 48 39 (350)
18 48 39 (330)
76 41 39 (172)
76 48 39 (345)
1578 48 39(350)
405 10515 39 (178)
405 48 39 (340)
10515 48 39 (373)
264 41 39 (167)
264 38 39 (177)
264 48 39 (403)
2958 41 48 (201)
2958 41 39 (169)
2958 32 48 (173)
2958 48 39 (570)
45 48 39 (361)
242 48 39 (300)

956 41 39 (170)
956 48 39(405)
31 48 39 (336)
479 48 39 (358)
3270 48 39 (416)
783 48 39 (439)
175 41 48 (188)
175 41 39 (239)
175 38 39 (166)
175 48 39 (383)
522 41 39 (168)
522 48 39 (425)
258 41 48 (193)
258 41 39 (205)
258 48 39 (452)
179 41 39 (184)
179 48 39 (452)
19 48 39 (402)
161 41 39 (168)
161 48 39 (356)
117 41 48 (168)
117 41 39 (189)
117 48 39 (378)
13041 41 48 (264)
13041 41 39 (311)
13041 48 39 (444)
78 41 48 (167)
78 41 39 (164)
78 32 48 (183)
78 38 48 (171)
78 38 39 (181)
78 48 39 (610)
37 41 38 (246)
37 41 48 (172)
37 41 39 (203)
37 32 38 (185)
37 38 48 (557)
37 38 39 (684)
37 48 39 (440)
1004 32 48 (167)
1004 32 39 (167)
1004 48 39 (381)
677 41 48 (166)
677 41 39 (175)
677 32 48 (179)
677 48 39 (446)

589 48 39 (443)
49 41 48 (177)
49 41 39 (172)
49 32 48 (168)
49 38 48 (176)
49 38 39 (172)
49 48 39 (617)
201 41 48 (183)
201 41 39 (185)
201 48 39 (451)
548 41 48 (176)
548 41 39 (182)
548 32 48 (170)
548 32 39 (171)
548 48 39 (467)
15832 41 48 (260)
15832 41 39 (322)
15832 48 39 (436)
249 41 48 (183)
249 41 39 (209)
249 38 48 (167)
249 38 39 (207)
249 48 39 (510)
1393 41 39 (180)
1393 48 39 (507)
16217 41 48 (287)
16217 41 39 (335)
16217 48 39 (463)
740 48 39 (392)
286 41 38 (259)
286 41 39 (208)
286 38 39 (728)
286 48 39 (464)
301 41 48 (172)
301 41 39 (194)
301 32 48 (188)
301 32 39 (184)
301 48 39 (473)
604 41 48 (199)
286 32 38 (222)
286 38 48 (581)
604 41 39 (236)
604 48 39 (520)
824 41 48 (218)
824 41 39 (238)
824 32 48 (166)

824 48 39 (471)
592 41 48 (206)
592 41 39 (235)
592 32 39 (173)
592 48 39 (515)
338 41 48 (173)
338 41 39 (187)
338 48 39 (516)
14098 48 39 (540)
123 41 48 (168)
123 41 39 (168)
123 48 39 (509)
16010 41 48 (315)
16010 41 39 (368)
16010 32 48 (190)
16010 32 39 (180)
16010 48 39 (529)
9 41 48 (231)
9 41 39 (239)
9 48 39 (546)
185 41 48 (210)
185 41 39 (223)
185 32 48 (184)
185 32 39 (183)
185 38 39 (165)
185 48 39 (555)
1146 41 48 (313)
1146 41 39 (391)
1146 32 48 (211)
1146 32 39 (216)
1146 38 48 (181)
1146 38 39 (239)
1146 48 39 (623)
12925 38 48 (175)
12925 38 39 (230)
12925 48 39 (588)
255 41 48 (271)
255 41 39 (274)
255 32 48 (243)
255 32 39 (230)
255 38 48 (198)
255 38 39 (216)
255 48 39 (813)
533 41 48 (248)
533 41 39 (275)
533 32 48 (179)

533 32 39 (182)
533 38 48 (170)
533 38 39 (178)
533 48 39 (632)
60 41 39 (219)
60 32 39 (181)
60 48 39 (609)
79 41 48 (332)
79 41 39 (378)
79 32 48 (231)
79 32 39 (244)
79 38 48 (193)
79 38 39 (231)
79 48 39 (699)
2238 270 39 (194)
2238 271 39 (221)
2238 225 48 (219)
2238 225 39 (317)
2238 41 48 (371)
2238 41 39 (472)
2238 32 48 (203)
2238 32 39 (257)
2238 38 48 (247)
2238 38 39 (308)
2238 48 39 (788)
270 271 41 (219)
270 271 48 (369)
270 271 39 (484)
270 225 48 (175)
270 225 39 (236)
270 41 48 (315)
270 41 39 (399)
270 32 48 (265)
270 32 39 (276)
270 38 48 (239)
270 38 39 (291)
270 48 39 (733)
147 41 48 (280)
147 41 39 (336)
147 32 48 (223)
147 32 39 (249)
147 38 48 (215)
147 38 39 (253)
147 48 39 (742)
1327 41 48 (307)
1327 41 39 (380)
1327 32 48 (187)

1327 32 39 (209)
1327 38 48 (198)
1327 38 39 (274)
1327 48 39 (720)
438 41 48 (296)
438 41 39 (365)
438 32 48 (227)
438 32 39 (244)
438 38 48 (211)
438 38 39 (272)
438 48 39 (780)
413 41 48 (287)
413 41 39 (318)
413 32 48 (216)
413 32 39 (217)
413 38 48 (213)
413 38 39 (221)
413 48 39 (781)
271 225 48 (220)
271 225 39 (309)
271 41 48 (346)
271 41 39 (451)
271 32 48 (256)
271 32 39 (310)
271 38 48 (232)
271 38 39 (290)
271 48 39 (827)
475 41 48 (428)
475 41 39 (476)
475 32 48 (308)
475 32 39 (313)
475 38 48 (290)
475 38 39 (308)
475 48 39(1092)
101 41 48 (330)
101 41 39 (368)
101 32 48 (256)
101 32 39 (270)
101 38 48 (229)
101 38 39 (254)
101 48 39 (946)
310 89 48 (190)
310 89 39 (194)
310 41 32 (186)
310 41 38 (166)
310 41 48 (547)
310 41 39 (625)

310 32 48 (391)
310 32 39 (382)
310 38 48 (298)
310 38 39 (336)
310 48 39 (1347)
110 36 38 (182)
110 41 38 (666)
110 41 48 (423)
110 41 39 (515)
110 32 38 (443)
110 32 48 (267)
110 32 39 (286)
110 38 48 (1361)
110 38 39 (1740)
110 48 39 (1037)
36 170 38 (210)
36 41 38 (671)
36 41 38 39 (553)
36 41 48 (388)
36 41 39 (572)
36 32 38 (472)
36 32 48 (262)
36 32 39 (324)
36 38 48 (1360)
36 38 39 (1945)
36 48 39 (1116)
237 89 48 (174)
237 89 39 (181)
237 41 38 (205)
237 41 48 (375)
237 41 39 (477)
237 32 48 (363)
237 32 39 (349)
237 38 48 (414)
237 38 39 (512)
237 48 39 (1244)
170 89 38 (169)
170 41 38 (794)
170 41 48 (492)
170 41 39 (624)
170 32 38 (532)
170 32 48 (308)
170 32 39 (332)
170 38 48 (1538)
170 38 39 (2019)
170 48 39 (1206)
225 41 32 (207)

225 41 38 (224)		
225 41 48 (537)		
225 41 39 (726)		
225 32 48 (366)		
225 32 39 (466)		
225 38 48 (398)		
225 38 39 (535)		
225 48 39 (1400)		
89 65 48 (243)		
89 65 39 (225)		
89 41 32 (187)		
89 41 38 (210)		
89 41 38 48 (173)		
89 41 38 39 (174)		
89 41 48 (606)		
89 41 39 (619)		
89 32 48 (573)		
89 32 39 (532)		
89 38 48 (578)		
89 38 39 (589)		
89 48 39 (2125)		
65 41 32 (221)		
65 41 38 (227)		
65 41 48 (663)		
65 41 39 (792)		
65 32 48 (492)		
65 32 39 (512)		
65 38 48 (408)		
65 38 39 (488)		
65 48 39 (1797)		
41 32 38 (805)		
41 32 48 (2063)		
41 38 48 (2374)		
41 38 39 (3051)		
41 48 39 (7366)		
32 38 48 (1646)		
32 38 39 (1840)		
32 48 39 (5402)		
38 48 39 (6102)		

APPENDIX 4: Whole set of frequent 4-itemsets

Itemsets (frequency)	Itemsets (frequency)	Itemsets (frequency)
3904 38 48 39 (173)	147 38 48 39 (176)	170 32 38 48 (305)
47 38 48 39 (212)	1327 41 48 39 (260)	170 32 38 39 (326)
281 38 48 39 (184)	1327 38 48 39 (170)	170 32 48 39 (215)
790 38 48 39 (235)	438 41 48 39 (265)	170 38 48 39 (1193)
56 38 48 39 (219)	438 32 48 39 (177)	225 41 32 39 (178)
8978 41 48 39 (172)	438 38 48 39 (164)	225 41 38 39 (198)
105 38 48 39 (290)	413 41 48 39 (228)	225 41 48 39 (469)
16011 16010 41 39 (190)	413 38 48 39 (170)	225 32 48 39 (298)
16011 16010 48 39 (269)	271 225 48 39 (183)	225 38 48 39 (337)
55 38 48 39 (243)	271 41 48 39 (272)	89 65 48 39 (186)
371 38 48 39 (294)	271 32 48 39 (188)	89 41 48 39 (522)
175 41 48 39 (166)	271 38 48 39 (188)	89 32 48 39 (451)
13041 41 48 39 (219)	475 41 48 39 (372)	89 38 48 39 (467)
37 41 38 48 (172)	475 32 48 39 (234)	65 41 32 39 (172)
37 41 38 39 (202)	475 38 48 39 (243)	65 41 38 39 (191)
37 38 48 39 (433)	101 41 48 39 (264)	65 41 48 39 (547)
15832 41 48 39 (220)	101 32 48 39 (182)	65 32 48 39 (357)
16217 41 48 39 (234)	101 38 48 39 (181)	65 38 48 39 (325)
286 41 38 39 (203)	310 89 48 39 (167)	41 32 38 48 (540)
286 38 48 39 (458)	310 41 48 39 (486)	41 32 38 39 (622)
604 41 48 39 (171)	310 32 48 39 (307)	41 32 48 39 (1646)
824 41 48 39 (172)	310 38 48 39 (251)	41 38 48 39 (1991)
592 41 48 39 (170)	110 41 38 48 (419)	32 38 48 39 (1236)
16010 41 48 39 (261)	110 41 38 39 (511)	
9 41 48 39 (191)	110 41 48 39 (347)	
185 41 48 39 (168)	110 32 38 48 (264)	
1146 41 48 39 (260)	110 32 38 39 (284)	
255 41 48 39 (230)	110 32 48 39 (202)	
255 32 48 39 (189)	110 38 48 39 (1031)	
255 38 48 39 (170)	36 41 38 48 (377)	
533 41 48 39 (215)	36 41 48 39 (343)	
79 41 48 39 (270)	36 32 38 48 (254)	
79 32 48 39 (174)	36 32 38 39 (308)	
2238 225 48 39 (196)	36 32 48 39 (196)	
2238 41 48 39 (322)	36 38 48 39 (1080)	
2238 38 48 39 (203)	237 41 38 39 (171)	
270 271 41 39 (179)	237 41 48 39 (312)	
270 271 48 39 (284)	237 32 48 39 (256)	
270 41 48 39 (258)	237 38 48 39 (352)	
270 32 48 39 (193)	170 41 38 48 (484)	
270 38 48 39 (196)	170 41 38 39 (615)	
147 41 48 39 (234)	170 41 48 39 (419)	

APPENDIX 5: The whole Set of association rules

Rule	Support	Confidence	Correlation	Cosine	Interest
9 41 48 ====>> 39	191	0. 82684	0. 0261311	0. 0558252	1. 4385
11 41 ====>> 39	166	0. 84264	0. 0256393	0. 0525386	1. 46599
36 32 ====>> 38	472	0. 955466	0. 15316	0. 170048	5. 40111
36 32 39 ====>> 38	308	0. 950617	0. 123146	0. 137016	5. 37371
36 32 48 ====>> 38	254	0. 969466	0. 113396	0. 125654	5. 48025
36 32 48 39 ====>> 38	191	0. 97449	0. 0986637	0. 109244	5. 50865
36 ====>> 38	2790	0. 950273	0. 376173	0. 412306	5. 37176
36 39 ====>> 38	1945	0. 954836	0. 313532	0. 345078	5. 39755
36 48 ====>> 38	1360	0. 960452	0. 262351	0. 289401	5. 4293
36 48 39 ====>> 38	1080	0. 967742	0. 234668	0. 258872	5. 47051
36 41 ====>> 38	671	0. 958571	0. 183261	0. 20308	5. 41867
36 41 38 ====>> 39	553	0. 824143	0. 0441704	0. 0948346	1. 43381
36 41 39 ====>> 38	553	0. 966783	0. 167279	0. 185149	5. 46509
36 41 48 ====>> 38	377	0. 971649	0. 138475	0. 153256	5. 4926
36 41 38 48 ====>> 39	334	0. 885942	0. 041245	0. 0764151	1. 54132
36 41 48 39 ====>> 38	334	0. 973761	0. 13051	0. 144409	5. 50453
36 41 48 ====>> 38 39	334	0. 860825	0. 153597	0. 166711	7. 33611
36 41 ====>> 39	572	0. 817143	0. 0438555	0. 0960396	1. 42163
36 41 48 ====>> 39	343	0. 884021	0. 0415866	0. 0773538	1. 53798
36 170 ====>> 38	210	0. 981308	0. 103987	0. 114949	5. 5472
37 32 ====>> 38	185	0. 994624	0. 0985345	0. 10862	5. 62247
37 ====>> 38	1046	0. 973929	0. 231955	0. 255578	5. 50549
37 39 ====>> 38	684	0. 967468	0. 186279	0. 205987	5. 46896
37 48 ====>> 38	557	0. 985841	0. 170256	0. 18764	5. 57282
37 48 39 ====>> 38	433	0. 984091	0. 149815	0. 165293	5. 56293
37 41 ====>> 38	246	0. 995951	0. 113772	0. 125337	5. 62997
37 41 38 ====>> 39	202	0. 821138	0. 0263585	0. 057212	1. 42858
37 41 39 ====>> 38	202	0. 995074	0. 103006	0. 113526	5. 62501
37 41 ====>> 38 39	202	0. 817814	0. 115369	0. 126368	6. 96956
37 41 48 ====>> 38	172	1	0. 0953691	0. 105017	5. 65286
37 41 ====>> 39	203	0. 821862	0. 0264898	0. 0573787	1. 42984
41 32 38 48 ====>> 39	448	0. 82963	0. 0404664	0. 0856415	1. 44335
41 38 48 ====>> 39	1991	0. 838669	0. 088791	0. 181524	1. 45908
41 48 ====>> 39	7366	0. 816811	0. 165248	0. 344572	1. 42105
47 ====>> 38	432	0. 972973	0. 148425	0. 164167	5. 50008
47 39 ====>> 38	313	0. 978125	0. 126732	0. 140108	5. 5292
47 48 ====>> 38	269	0. 981752	0. 11777	0. 130128	5. 54971
47 48 39 ====>> 38	212	0. 981481	0. 104495	0. 115505	5. 54818
49 32 ====>> 48	168	0. 819512	0. 0330137	0. 0571625	1. 71472
49 41 ====>> 48	177	0. 804545	0. 0327045	0. 0581354	1. 68341
49 39 ====>> 48	617	0. 803385	0. 0610786	0. 108463	1. 68098
55 ====>> 38	657	0. 933239	0. 177832	0. 198277	5. 27547
55 39 ====>> 38	422	0. 933628	0. 142361	0. 158941	5. 27767
55 48 ====>> 38	309	0. 950769	0. 123361	0. 137249	5. 37457

55 48 39 ====>> 38	243	0. 968127	0. 110796	0. 122818	5. 47269
55 41 ====>> 38	171	0. 977143	0. 0935274	0. 103507	5. 52365
56 ====>> 38	514	0. 960748	0. 160508	0. 177942	5. 43097
56 39 ====>> 38	352	0. 967033	0. 133326	0. 147736	5. 4665
56 48 ====>> 38	295	0. 954693	0. 120885	0. 13438	5. 39674
56 48 39 ====>> 38	219	0. 969027	0. 105238	0. 11665	5. 47777
60 41 ====>> 39	219	0. 829545	0. 0282405	0. 059875	1. 4432
65 41 38 ====>> 39	191	0. 84141	0. 0274007	0. 0563149	1. 46385
65 41 48 ====>> 39	547	0. 825038	0. 0440618	0. 0943699	1. 43536
78 38 ====>> 39	181	0. 826484	0. 0254057	0. 0543325	1. 43788
78 41 ====>> 39	164	0. 81592	0. 0233153	0. 0513865	1. 4195
78 41 ====>> 48	167	0. 830846	0. 0337739	0. 0573848	1. 73844
79 41 48 ====>> 39	270	0. 813253	0. 0296555	0. 0658261	1. 41486
89 32 ====>> 48	573	0. 803647	0. 0588796	0. 104541	1. 68153
89 32 39 ====>> 48	451	0. 847744	0. 057686	0. 0952575	1. 77379
89 38 48 ====>> 39	467	0. 807958	0. 038314	0. 0862891	1. 40565
89 41 38 ====>> 39	174	0. 828571	0. 0250832	0. 0533387	1. 44151
89 41 38 ====>> 48	173	0. 82381	0. 0338352	0. 0581587	1. 72371
89 41 ====>> 39	619	0. 845628	0. 0501272	0. 101634	1. 47118
89 41 ====>> 48	606	0. 827869	0. 0641025	0. 109118	1. 73221
89 41 48 ====>> 39	522	0. 861386	0. 0482283	0. 094197	1. 4986
89 41 39 ====>> 48	522	0. 843296	0. 0615063	0. 102212	1. 76449
89 65 39 ====>> 48	186	0. 826667	0. 0353151	0. 0604088	1. 72969
105 ====>> 38	643	0. 978691	0. 182068	0. 200873	5. 5324
105 39 ====>> 38	449	0. 986813	0. 152874	0. 168552	5. 57832
105 48 ====>> 38	367	0. 986559	0. 138121	0. 152366	5. 57688
105 48 39 ====>> 38	290	0. 996564	0. 123614	0. 136127	5. 63343
110 32 ====>> 38	443	0. 986637	0. 151825	0. 167407	5. 57732
110 32 39 ====>> 38	284	0. 993007	0. 122012	0. 134471	5. 61333
110 32 48 ====>> 38	264	0. 988764	0. 117264	0. 129372	5. 58934
110 32 48 39 ====>> 38	201	0. 995049	0. 102748	0. 113244	5. 62488
110 36 ====>> 38	182	0. 978495	0. 096591	0. 106858	5. 53129
110 ====>> 38	2725	0. 975304	0. 378526	0. 412807	5. 51326
110 39 ====>> 38	1740	0. 989198	0. 303733	0. 332208	5. 5918
110 48 ====>> 38	1361	0. 986232	0. 26746	0. 293367	5. 57503
110 48 39 ====>> 38	1031	0. 994214	0. 233676	0. 256367	5. 62015
110 41 ====>> 38	666	0. 983752	0. 186007	0. 204962	5. 56101
110 41 39 ====>> 38	511	0. 992233	0. 163786	0. 180306	5. 60895
110 41 48 ====>> 38	419	0. 990544	0. 148052	0. 163131	5. 59941
110 41 38 48 ====>> 39	346	0. 825776	0. 0350822	0. 0750883	1. 43665
110 41 48 39 ====>> 38	346	0. 997118	0. 135119	0. 148732	5. 63657
110 41 48 ====>> 38 39	346	0. 817967	0. 151161	0. 165402	6. 97086
110 41 48 ====>> 39	347	0. 820331	0. 0344853	0. 0749484	1. 42717
147 38 48 ====>> 39	176	0. 818605	0. 024384	0. 0533208	1. 42417
147 41 48 ====>> 39	234	0. 835714	0. 0297907	0. 0621212	1. 45394
170 32 ====>> 38	532	0. 985185	0. 166289	0. 183319	5. 56911
170 32 39 ====>> 38	326	0. 981928	0. 129708	0. 143266	5. 5507

170 32 48 ====>> 38	305	0. 99026	0. 126207	0. 139161	5. 5978
170 32 48 39 ====>> 38	213	0. 990698	0. 105447	0. 11632	5. 60028
170 ====>> 38	3031	0. 978057	0. 400743	0. 435982	5. 52882
170 39 ====>> 38	2019	0. 980573	0. 325691	0. 356288	5. 54304
170 48 ====>> 38	1538	0. 987797	0. 284935	0. 312108	5. 58388
170 48 39 ====>> 38	1193	0. 989221	0. 250703	0. 275081	5. 59193
170 41 ====>> 38	794	0. 986335	0. 203629	0. 224087	5. 57562
170 41 39 ====>> 38	615	0. 985577	0. 178927	0. 197141	5. 57133
170 41 48 ====>> 38	484	0. 98374	0. 158399	0. 174725	5. 56094
170 41 38 48 ====>> 39	413	0. 853306	0. 0418567	0. 0833932	1. 48454
170 41 48 39 ====>> 38	413	0. 98568	0. 146467	0. 161561	5. 57191
170 41 48 ====>> 38 39	413	0. 839431	0. 168084	0. 183064	7. 15378
170 41 48 ====>> 39	419	0. 851626	0. 0419486	0. 083914	1. 48162
170 89 ====>> 38	169	0. 988304	0. 0937397	0. 103486	5. 58674
175 41 ====>> 39	239	0. 8157	0. 0281389	0. 0620251	1. 41912
175 41 48 ====>> 39	166	0. 882979	0. 0288175	0. 0537814	1. 53617
179 41 ====>> 39	184	0. 821429	0. 0251787	0. 0546131	1. 42908
225 32 48 ====>> 39	298	0. 814208	0. 0312677	0. 0691957	1. 41652
225 38 48 ====>> 39	337	0. 846734	0. 0370425	0. 0750398	1. 47311
225 41 32 ====>> 39	178	0. 859903	0. 0279776	0. 0549589	1. 49602
225 41 38 ====>> 39	198	0. 883929	0. 0315593	0. 0587685	1. 53782
225 41 ====>> 39	726	0. 827822	0. 051303	0. 108903	1. 44021
225 41 48 ====>> 39	469	0. 873371	0. 0472795	0. 089906	1. 51945
225 48 ====>> 39	1400	0. 806452	0. 0664115	0. 149264	1. 40303
237 38 48 ====>> 39	352	0. 850242	0. 0382705	0. 0768503	1. 47921
237 41 38 ====>> 39	171	0. 834146	0. 0253265	0. 0530545	1. 45121
237 41 48 ====>> 39	312	0. 832	0. 0340037	0. 0715718	1. 44747
249 41 ====>> 39	209	0. 810078	0. 0257835	0. 0578016	1. 40934
255 32 39 ====>> 48	189	0. 821739	0. 0352018	0. 0607123	1. 71938
255 38 ====>> 39	216	0. 808989	0. 0261093	0. 0587221	1. 40744
255 38 48 ====>> 39	170	0. 858586	0. 0272348	0. 0536685	1. 49373
255 41 ====>> 39	274	0. 825301	0. 0311539	0. 0668013	1. 43582
255 41 ====>> 48	271	0. 816265	0. 041644	0. 0724567	1. 70793
255 41 48 ====>> 39	230	0. 848709	0. 030766	0. 0620649	1. 47654
255 41 39 ====>> 48	230	0. 839416	0. 0404072	0. 067691	1. 75637
258 41 ====>> 39	205	0. 807087	0. 025257	0. 05714	1. 40413
264 41 ====>> 39	167	0. 818627	0. 0237528	0. 0519403	1. 42421
270 38 48 ====>> 39	196	0. 820084	0. 0258685	0. 0563197	1. 42674
270 41 48 ====>> 39	258	0. 819048	0. 0295854	0. 0645755	1. 42494
270 271 41 ====>> 39	179	0. 817352	0. 0244839	0. 0537321	1. 42199
271 38 48 ====>> 39	188	0. 810345	0. 024474	0. 0548298	1. 4098
271 225 48 ====>> 39	183	0. 831818	0. 0260035	0. 0548079	1. 44716
281 ====>> 38	432	0. 953642	0. 146289	0. 162528	5. 39081
281 39 ====>> 38	274	0. 951389	0. 116195	0. 129285	5. 37807
281 48 ====>> 38	259	0. 962825	0. 113943	0. 126449	5. 44272
281 48 39 ====>> 38	184	0. 953368	0. 0953113	0. 106055	5. 38926
286 32 ====>> 38	222	0. 969432	0. 10599	0. 11747	5. 48006

286 ====>> 38	1116	0. 943364	0. 234253	0. 259816	5. 33271
286 39 ====>> 38	728	0. 970667	0. 192684	0. 21286	5. 48704
286 48 ====>> 38	581	0. 98308	0. 173561	0. 191371	5. 55721
286 48 39 ====>> 38	458	0. 987069	0. 154436	0. 170255	5. 57976
286 41 ====>> 38	259	0. 966418	0. 11425	0. 126685	5. 46303
286 41 39 ====>> 38	203	0. 975962	0. 101834	0. 112709	5. 51697
310 32 39 ====>> 48	307	0. 803665	0. 0430186	0. 0765218	1. 68156
310 38 48 ====>> 39	251	0. 842282	0. 0315102	0. 0645905	1. 46536
310 41 ====>> 39	625	0. 869263	0. 0540114	0. 103543	1. 5123
310 41 48 ====>> 39	486	0. 888483	0. 0501357	0. 0923094	1. 54574
310 89 ====>> 39	194	0. 850877	0. 0284363	0. 0570739	1. 48032
310 89 ====>> 48	190	0. 833333	0. 03623	0. 0613006	1. 74364
310 89 48 ====>> 39	167	0. 878947	0. 0285918	0. 0538199	1. 52915
310 89 39 ====>> 48	167	0. 860825	0. 0359977	0. 0584109	1. 80116
370 ====>> 38	362	0. 965333	0. 135043	0. 149688	5. 45689
370 39 ====>> 38	241	0. 971774	0. 110638	0. 122542	5. 4933
370 48 ====>> 38	191	0. 97449	0. 0986637	0. 109244	5. 50865
371 ====>> 38	767	0. 980818	0. 199304	0. 219627	5. 54443
371 39 ====>> 38	526	0. 988722	0. 165767	0. 182609	5. 58911
371 48 ====>> 38	365	0. 98916	0. 138002	0. 15215	5. 59158
371 38 48 ====>> 39	294	0. 805479	0. 0300864	0. 0683603	1. 40134
371 48 39 ====>> 38	294	0. 99661	0. 12447	0. 137066	5. 6337
371 41 ====>> 38	171	0. 994186	0. 0946954	0. 104406	5. 61999
390 ====>> 38	354	0. 941489	0. 131134	0. 146185	5. 32211
390 39 ====>> 38	241	0. 934109	0. 107505	0. 120144	5. 28038
390 48 ====>> 38	197	0. 938095	0. 0974743	0. 108855	5. 30292
438 41 ====>> 39	365	0. 852804	0. 0392773	0. 0783744	1. 48367
438 41 48 ====>> 39	265	0. 89527	0. 0376249	0. 0684232	1. 55755
475 38 48 ====>> 39	243	0. 837931	0. 0305773	0. 0633885	1. 45779
475 41 ====>> 39	476	0. 835088	0. 042473	0. 0885671	1. 45285
475 41 48 ====>> 39	372	0. 869159	0. 041588	0. 0798774	1. 51212
504 ====>> 38	245	0. 822148	0. 0984773	0. 113645	4. 64749
504 39 ====>> 38	170	0. 867347	0. 0854099	0. 0972331	4. 90299
533 41 ====>> 39	275	0. 818452	0. 0304848	0. 0666448	1. 42391
533 41 48 ====>> 39	215	0. 866935	0. 0313859	0. 0606479	1. 50825
570 41 ====>> 39	173	0. 848039	0. 0266179	0. 0538064	1. 47538
592 41 48 ====>> 39	170	0. 825243	0. 0245168	0. 0526161	1. 43572
597 ====>> 39	209	0. 832669	0. 0278721	0. 058602	1. 44864
604 41 ====>> 39	236	0. 825175	0. 028893	0. 0619915	1. 4356
604 41 48 ====>> 39	171	0. 859297	0. 027372	0. 0538484	1. 49496
734 ====>> 48	329	0. 898907	0. 054415	0. 0837787	1. 88085
734 39 ====>> 48	229	0. 898039	0. 0452978	0. 0698625	1. 87903
790 ====>> 38	508	0. 971319	0. 160827	0. 177871	5. 49073
790 39 ====>> 38	356	0. 978022	0. 135181	0. 149414	5. 52862
790 48 ====>> 38	307	0. 974603	0. 125182	0. 138509	5. 50929
790 48 39 ====>> 38	235	0. 983264	0. 110176	0. 12172	5. 55825
790 41 ====>> 38	202	0. 990196	0. 102644	0. 113248	5. 59744

840 ====>> 38	379	0. 969309	0. 138602	0. 153477	5. 47937
840 39 ====>> 38	245	0. 976096	0. 111912	0. 123829	5. 51773
840 48 ====>> 38	202	0. 980583	0. 101928	0. 112697	5. 5431
978 48 ====>> 39	164	0. 807882	0. 0226502	0. 0511327	1. 40551
979 48 ====>> 39	222	0. 834586	0. 0289086	0. 0604666	1. 45197
1046 ====>> 32	339	0. 928767	0. 129281	0. 14408	5. 39869
1046 48 ====>> 32	171	0. 9	0. 0896396	0. 100732	5. 23148
1135 ====>> 48	541	0. 811094	0. 0582355	0. 10205	1. 69711
1135 39 ====>> 48	386	0. 842795	0. 0527852	0. 0878685	1. 76344
1146 41 48 ====>> 39	260	0. 830671	0. 0308943	0. 0652836	1. 44516
1188 ====>> 38	256	0. 927536	0. 110238	0. 12339	5. 24323
1327 38 48 ====>> 39	170	0. 858586	0. 0272348	0. 0536685	1. 49373
1327 41 ====>> 39	380	0. 827887	0. 0370359	0. 0787917	1. 44032
1327 41 48 ====>> 39	260	0. 846906	0. 0325369	0. 0659185	1. 47341
1344 41 ====>> 39	213	0. 825581	0. 0274824	0. 0589078	1. 43631
1393 41 ====>> 39	180	0. 837209	0. 0262447	0. 0545326	1. 45654
1529 ====>> 38	295	0. 916149	0. 117295	0. 13164	5. 17886
1529 39 ====>> 38	202	0. 90583	0. 0961954	0. 108316	5. 12053
1590 ====>> 38	281	0. 959044	0. 118361	0. 131452	5. 42134
1590 39 ====>> 38	193	0. 969849	0. 0988392	0. 109553	5. 48242
1831 ====>> 38	252	0. 969231	0. 112928	0. 125143	5. 47893
1956 ====>> 48	243	0. 804636	0. 0383462	0. 068121	1. 6836
2238 38 48 ====>> 39	203	0. 821862	0. 0264898	0. 0573787	1. 42984
2238 41 ====>> 39	472	0. 82807	0. 0413279	0. 0878229	1. 44064
2238 41 48 ====>> 39	322	0. 867925	0. 0385449	0. 0742629	1. 50997
2238 48 ====>> 39	788	0. 825131	0. 0529901	0. 113273	1. 43552
2238 225 48 ====>> 39	196	0. 894977	0. 0323195	0. 0588352	1. 55704
2805 ====>> 38	212	0. 954955	0. 102447	0. 113934	5. 39823
2958 32 ====>> 48	173	0. 915344	0. 0405887	0. 0613047	1. 91524
2958 41 ====>> 48	201	0. 926267	0. 0445847	0. 0664729	1. 93809
2958 ====>> 48	779	0. 861726	0. 0782057	0. 126221	1. 80305
2958 39 ====>> 48	570	0. 870229	0. 0679475	0. 108501	1. 82084
3005 ====>> 38	195	0. 951219	0. 0979646	0. 109056	5. 37711
3904 ====>> 38	333	0. 97654	0. 130581	0. 144398	5. 52024
3904 38 ====>> 39	268	0. 804805	0. 0286481	0. 0652403	1. 40016
3904 39 ====>> 38	268	0. 974545	0. 116929	0. 129408	5. 50897
3904 48 ====>> 38	193	0. 965	0. 0984818	0. 109279	5. 45501
3904 38 48 ====>> 39	173	0. 896373	0. 0304681	0. 0553185	1. 55947
3904 48 39 ====>> 38	173	0. 96648	0. 0933321	0. 103541	5. 46338
3904 48 ====>> 38 39	173	0. 865	0. 110777	0. 120272	7. 37169
3904 ====>> 39	275	0. 806452	0. 029199	0. 0661544	1. 40303
3904 48 ====>> 39	179	0. 895	0. 0308845	0. 0562265	1. 55708
8691 39 ====>> 48	213	0. 822394	0. 0374325	0. 0644775	1. 72075
8978 41 ====>> 39	243	0. 837931	0. 0305773	0. 0633885	1. 45779
8978 41 48 ====>> 39	172	0. 877551	0. 0289074	0. 0545762	1. 52672
8978 48 ====>> 39	276	0. 802326	0. 0288053	0. 0661048	1. 39585
12935 39 ====>> 48	214	0. 801498	0. 0357024	0. 0638023	1. 67703

12951 48 ====>> 39	183	0.835616	0.0263275	0.0549329	1.45377
13041 41 48 ====>> 39	219	0.829545	0.0282405	0.059875	1.4432
15832 41 48 ====>> 39	220	0.846154	0.0298522	0.0606093	1.4721
16010 41 48 ====>> 39	261	0.828571	0.0307389	0.0653263	1.44151
16011 ====>> 16010	651	0.973094	0.690949	0.693809	65.1899
16011 39 ====>> 16010	419	0.981265	0.555949	0.558949	65.7373
16011 41 ====>> 16010	236	0.967213	0.413716	0.416475	64.7959
16011 16010 41 ====>> 39	190	0.805085	0.0241334	0.0549415	1.40065
16011 41 39 ====>> 16010	190	0.979381	0.373507	0.376032	65.6111
16011 48 ====>> 16010	362	0.967914	0.512961	0.515994	64.8429
16011 48 39 ====>> 16010	269	0.978182	0.444349	0.447155	65.5307
16217 41 48 ====>> 39	234	0.815331	0.0278057	0.061359	1.41847
16431 ====>> 16430	348	0.991453	0.899533	0.899955	205.184
16430 ====>> 16431	348	0.816901	0.899533	0.899955	205.184
16431 39 ====>> 16430	186	0.984127	0.654572	0.655507	203.668
16430 39 ====>> 16431	186	0.841629	0.66683	0.667826	211.395
16431 48 ====>> 16430	174	0.994286	0.636346	0.637273	205.77