Tribhuvan University

Institute of Science and Technology

Comparison of Association rule mining algorithms - Apriori and FP Growth

Dissertation

Submitted to

Central Department of Computer Science and Information Technology

Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements

for the Master's Degree in Computer Science and Information Technology

By

Krishan Dev Bhatt

November, 2011

Tribhuvan University

Institute of Science and Technology

Comparison of Association rule mining algorithms - Apriori and FP Growth

Dissertation

Submitted to

Central Department of Computer Science and Information Technology

Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements

for the Master's Degree in Computer Science and Information Technology

By

Krishan Dev Bhatt

November, 2011

Supervisor

Prof. Dr. Shashidhar Ram Joshi

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk, Nepal

(Head)

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

… … … … … … …

Krishan Dev Bhatt

Date: November, 2011

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Supervisor's Recommendation

I hereby recommend that the dissertation prepared under my supervision by **Mr. Krishan Dev Bhatt** entitled **"Comparison of Association rule mining algorithms - Apriori and FP Growth"** be accepted as fulfilling in partial requirements for the degree of M. Sc. in Computer Science and Information Technology.

Prof. Dr. Shashidhar Ram Joshi

Department of Electronics and Computer Engineering,

Institute Of Engineering, Pulchowk, Nepal

(Head)

Date:_______________________________

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

LETTER OF APPROVAL

We certify that we have read this dissertation work and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of Master of Science in Computer Science and Information Technology.

Evaluation Committee

Dr. Tank Nath Dhamala

Head, Central Department of Computer Science and Information Technology
Tribhuvan University

Prof. Dr. Shashidhar Ram Joshi

Head,Department of Electronics and Computer Engineering, Institute of Engineering
Pulchowk, Nepal (Supervisor)

(External Examiner)

(Internal Examiner)

Date: _______________

ACKNOWLEDGEMENTS

I would like to express heartfelt regards to my supervisor Prof. Dr. Sashidhar Ram Joshi, Head of the Department, Institute of Engineering (IOE), Pulchowk for his continuous guideline, support and inspiration throughout the thesis work.

Dr. Tank Nath Dhamala, head of the Central Department of Computer Science and Information Technology, Kritipur deserves my special thanks as he has played vital role in bringing my supervisor and me together and his continuous support throughout the work.

I would like to express my deep sense of appreciation to Mr. Yoga Raj Joshi, lecturer of Central Department of Computer Science and Information Technology, Kritipur and all faculties of the department for their suggestion and comments for the improvement of the thesis.

ABSTRACT

Data mining is a part of a process called KDD-knowledge discovery in databases. This process consists basically of steps that are performed before carrying out data mining, such as data selection, data cleaning, pre-processing, and data transformation. Association rule techniques are used for data mining if the goal is to detect relationships or associations between specific values of categorical variables in large data sets. There may be thousands or millions of records that have to be read and to extract the rules. Frequent pattern mining is a very important task in data mining. The approaches applied to generate frequent set generally adopt candidate generation and pruning techniques for the satisfaction of the desired objectives. This dissertation shows how the different approaches achieve the objective of frequent mining along with the complexities required to perform the job. This dissertation looks into a comparison among Apriori and FP Growth algorithm. The process of the mining is helpful in generation of support systems for many computer related applications. It has been observed that with higher support and confidence on both algorithms, FP-Growth extracts the better association rules than Apriori algorithm. While decreasing the support and confidence value Apriori seems better than FP-Growth algorithm.

To my parents

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

FP	Frequent Pattern
KDD	knowledge discovery in databases
DB	Database

CHAPTER I

INTRODUCTION

1. Data, Information, and Knowledge

1.1. Data

Data are any facts, numbers, or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data in different formats and different databases. This includes:

- operational or transactional data such as, sales, cost, inventory, payroll, and accounting
- nonoperational data, such as industry sales, forecast data, and macro economic data
- meta data - data about the data itself, such as logical database design or data dictionary definitions

1.2. Information

The patterns, associations, or relationships among all this data can provide information. For example, analysis of retail point of sale transaction data can yield information on which products are selling and when.

1.3. Knowledge

Information can be converted into knowledge about historical patterns and future trends. For example, summary information on retail supermarket sales can be analyzed in light of promotional efforts to provide knowledge of consumer buying behavior. Thus, a manufacturer or retailer could determine which items are most susceptible to promotional efforts.

2. Data Mining

Data mining can be abstractly defined as the extraction of hidden predictive information from large databases. Data mining, as one of the IT services most needed by organizations, has been realized as an important way for discovering knowledge from the data and converting data rich to knowledge rich so as to assist strategic decision making. The benefits of using data mining for business and administrative problems have been demonstrated in various industries and governmental sectors, e.g., banking, insurance, direct-mail marketing, telecommunications, retails, and health care.

The ultimate goal of data mining is to create a model, a model that can improve the way to read and interpret the existing data and future data. Since there are so many techniques with data mining, the major step to creating a good model is to determine what type of technique to use. That will come with practice and experience, and some guidance. From there, the model needs to be refined to make it even more useful. Data mining commonly involves four classes of tasks:

2.1. Clustering

Clustering is the task of discovering groups and structures in the data that are in some way, without using known structures in the data. It is a division of data into groups of similar objects. Each group called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. It represents many data objects by few clusters, and hence, it models data by its clusters.

Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. Therefore, clustering is unsupervised learning of a hidden data concept. Data mining deals with large databases that impose on clustering analysis additional severe computational requirements.

Clustering techniques fall into a group of undirected data mining tools. The goal of undirected data mining is to discover structure in the data as a whole. There is no target variable to be variable to be predicted, thus no distinction is being made between independent and dependent variables.

Clustering techniques are used for combining observed examples into clusters (groups) which satisfy two main criteria:

a. Each group or cluster is homogenous; examples that belong to the same group are similar to each other.

b. Each group or cluster should be different from other clusters that are examples that belong to one cluster should be different from the examples of other clusters.

Depending on the clustering technique, clusters can be expressed in different ways:

a. Identified clusters may be exclusive, so that any example belongs to only one cluster.

b. They may be overlapping: an example may belong to several clusters.

c. They may be probabilistic, whereby an example belongs to each cluster with a certain probability.

d. Clusters might have hierarchical structure, having crude division of examples at highest level of hierarchy, which is then refined to sub-clusters at lower levels

2.2. Classification

Classification is the task of generalizing known structure to apply to new data. For example, an email program might attempt to classify an email as legitimate or spam. Common algorithms include decision tree learning, nearest neighbor, naive Bayesian classification, neural networks and support vector machines.

A classification task begins with a data set in which the class assignments are known. For example, a classification model that predicts credit risk could be developed based on observed data for many loan applicants over a period of time. In addition to the historical credit rating, the data might track employment history, home ownership or rental, years of residence, number and type of investments, and so on. Credit rating would be the target, the other attributes would be the predictors, and the data for each customer would constitute a case.

Classifications are discrete and do not imply order. Continuous, floating-point values would indicate a numerical, rather than a categorical target. A predictive model with a numerical target uses a regression algorithm, not a classification algorithm.

The simplest type of classification problem is binary classification. In binary classification, the target attribute has only two possible values: for example, high credit rating or low credit rating. Multiclass targets have more than two values: for example, low, medium, high, or unknown credit rating.

In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown.

Classification models are tested by comparing the predicted values to known target values in a set of test data. The historical data for a classification project is typically divided into two data sets: one for building the model; the other for testing the model. For example, a model that classifies customers as low, medium, or high value would also predict the probability of each classification for each customer.

2.3. Regression

Regression attempts to find a function which models the data with the least error. Regression is a data mining (machine learning) technique used to fit an equation to a dataset. The simplest form of regression, linear regression, uses the formula of a straight line ($y = mx + b$) and determines the appropriate values for m and b to predict the value of y based upon a given value of x. Advanced techniques, such as multiple regression, allow the use of more than one input variable and allow for the fitting of more complex models, such as a quadratic equation.

Basically, regression takes a numerical dataset and develops a mathematical formula that fits the data. When it is ready to use the results to predict future behavior, simply take the new data, plug it into the developed formula and get a prediction. The major limitation of this technique is that it only works well with continuous quantitative data (like weight, speed or age). For the categorical data where order is not significant (like color, name or gender) it is better off choosing another technique.

2.4. Association rule learning

Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk."Association rule searches for relationships between variables. For example a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. An association rule has two parts, an antecedent (if) and a consequent (then). An antecedent is an item found in the data. A consequent is an item that is found in combination with the antecedent.

3. Association Rule Mining

In data mining, association rule learning is a method for discovering interesting relations between variables in large databases. For example, the rule found in the sales data of a supermarket would indicate that if a customer buys onions and vegetables together, he or she is likely to also buy beef. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements.

Association rules provide information of this type in the form of "if-then" statements. These rules are computed from the data and, unlike the if-then rules of logic, association rules are probabilistic in nature.

Among all the available data mining methods, the discovery of associations between business events or transactions is one of the most commonly used data mining techniques. Association rule mining has been an important application in decision support and marketing strategy.

The problem of association rule mining is defined as: Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n binary attributes called items. Let $D = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the database. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of the form $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. The sets of items (for short itemsets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule respectively. To illustrate the concepts, we use a small example from the supermarket domain. The set of items is I = {milk, bread, butter, beer} and a small database containing the items (1 codes presence and 0 absence of an item in a transaction) is shown in the table to the right. An example rule for the supermarket could be $\{\mathrm{butter}, \mathrm{bread}\} \Rightarrow \{\mathrm{milk}\}$ meaning that if butter and bread is bought, customers also buy milk.

4. Support and Confidence

Association rules are created by analyzing data for frequent if/then patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the database. Confidence indicates the number of times the if/then statements have been found to be true.

Association rule is of the form: X → Y where X, Y are subsets of I, and X intersect Y = empty. There are rule two measures of value, support, and confidence Percentage of transactions which contain that item set is called support of the item set. Ssupport indicates the frequencies of the occurring patterns, and confidence denotes the strength of implication in the rule in dataset. The support of the rule X → Y is support (X UNION Y) and confidence of rule X → Y if percentage of transactions that contain X also contain Y, which can be written as the ration: Support(X UNION Y)/support(X)

Let D be a set of transactions where each transaction T is a set of items such that T   I. An itemset is a non-empty set of items.

Ex. {A,B} An itemset that contains k distinct items is K-itemset Frequency of a k-itemset : # of Ts that contain the k-itemset Ex. {A,B} is 2-itemset and support =1. A frequent k-itemset must have a frequency >min_sup

Tid	Items bought
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F

Support: Support, s, probability that a transaction contains X   Y. Support (X → Y) = P(X   Y)

Ex. support(A → D)= P(A   D) =3/5=60%

Confidence: Confidence, c, conditional probability that a transaction having X also contains Y. Confidence(X → Y) = P (Y|X) = support(X   Y) /support(X)

Ex.

Confidence (A → D) = P (A| D) =3/3=100%

4.1. Frequent pattern

A perceptual structure that occurs frequently in dataset is called frequent pattern.

4.2. Frequent pattern mining

In frequent pattern mining various kinds of association rules has been extracted from dataset.

4.3. Algorithms for association rule

Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all frequent itemsets in a database.
2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

While the second step is straight forward, the first step needs more attention. Finding all frequent itemsets in a database is difficult since it involves searching all possible itemsets (item combinations). The set of possible itemsets is the power set over I and has size 2n − 1 (excluding the empty set which is not a valid itemset). Although the size of the powerset grows exponentially in the number of items n in I, efficient search is possible using the downward-closure property of support (also called anti - monotonicity) which guarantees that for a frequent itemset, all its subsets are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent. Exploiting this property, efficient algorithms (e.g., Apriori and Eclat) can find all frequent itemsets.

Here is a list of the algorithms that can be used for association rule learning.

4.3.1. Apriori algorithm

Apriori is the one of the best-known algorithm to mine association rules. It uses a breadth-first search strategy to counting the support of itemsets and uses a candidate generation function which exploits the downward closure property of support. In computing and data mining, Apriori algorithm is a classical search algorithm of the associations. It is used for frequent itemset generation, successive approximation, starting with one item from itemset. In summary, the theoretical basis upon which the algorithm starts from the consideration that if a set of items (itemset) is frequent, then all its subsets are frequent, but if an itemset is not frequent, even then the sets containing it are frequent (monotonicity principle).

One area where this algorithm is applicable is the big market / basketball problem. To estimate the associations of use of a bottom up approach, where frequent subsets are constructed by adding an item at a time (the generation of candidate), groups of candidates are then tested on the data and the algorithm ends when there are no further extensions possible. In this process, the number of iterations is kmax + 1, kmax indicates the maximum cardinality of a frequent itemset.

There are other algorithms with similar purposes (and Winepi Minepi), but which are most common in areas where data are lacking timestamp (eg DNA sequences).

Apriori, while historically significant, suffers from some inefficiencies.In particular, the generation of candidates creates many subsets. In the process identifies the significant subsets only after finding all 2 | S | – 1 proper subsets, where S is the set of specific elements (support) where a particular subset of objects appears. The pseudo code for apriori algorithm is:

Following are some drawbacks of Apriori algorithm.

- Multiple scans on the transaction database
- Huge number of candidates generated and tested
- Tedious workload of support counting for candidates

Ideas for improvement

- Reduce the number transaction database scans
- Shrink number of candidates
- Facilitate support counting of candidates

4.3.2. FP-growth algorithm

FP-growth (frequent pattern growth) uses an extended prefix-tree (FP-tree) structure to store the database in a compressed form. FP-growth adopts a divide-and-conquer approach to decompose both the mining tasks and the databases. It uses a pattern fragment growth method to avoid the costly process of candidate generation and testing used by Apriori.

The idea behind the Frequent pattern growth is:

- Recursively grow frequent patterns by pattern and database partition

Method

- For each frequent item, construct its conditional pattern-base, and then its conditional FP-tree
- Repeat the process on each newly created conditional FP-tree
- Until the resulting FP-tree is empty, or it contains only one path—single path will generate all the combinations of its sub-paths, each of which is a frequent pattern. Note the “m”-conditional tree had a single path (ie we didn’t have to do recursion there, shown only for illustration purposes).

4.3.3. Eclat algorithm

Eclat is a depth-first search algorithm using set intersection. The Eclat algorithm is used to perform itemset mining. Itemset mining let us find frequent patterns in data like if a consumer buys milk, he also buys bread. This type of pattern is called association rules and is used in many application domains.

The basic idea for the eclat algorithm is use tidset intersections to compute the support of a candidate itemset avoiding the generation of subsets that does not exist in the prefix tree.

4.3.4. GUHA procedure ASSOC

GUHA is a general method for exploratory data analysis that has theoretical foundations in observational calculi. The ASSOC procedure is a GUHA method which mines for generalized association rules using fast bitstrings operations. The association rules mined by this method are more general than those output by apriori, for example "items" can be connected both with conjunction and disjunctions and the relation between antecedent and consequent of the rule is not restricted to setting minimum support and confidence as in apriori: an arbitrary combination of supported interest measures can be used.

4.3.5. OPUS search

OPUS is an efficient algorithm for rule discovery that, in contrast to most alternatives, does not require either monotone or anti-monotone constraints such as minimum support. Initially used to find rules for a fixed consequent it has subsequently been extended to find rules with any item as a consequent. OPUS search is the core technology in the popular Magnum Opus association discovery system.

4.3.6. Lore

A famous story about association rule mining is the "beer and diaper" story. A purported survey of behavior of supermarket shoppers discovered that customers (presumably young men) who buy diapers tend also to buy beer. This anecdote became popular as an example of how unexpected association rules might be found from everyday data. There are varying opinions as to how much of the story is true.

5. Motivation

Use of technology for data collection has seen an unprecedented growth in the last couple of decades. Individuals and organizations generate huge amount of data through everyday activities. Decreasing storage and computation costs have enabled us to collect data on different aspects of people’s lives such as their transaction records, phone call and email lists, personal health information and web browsing habits. Security issues, government regulations, and corporate policies require most of this data to be scanned for important information such as terrorist activities, credit card fraud detection, cheaper communications, and even personalized shopping recommendations. Such analysis of private information often raises concerns regarding the privacy rights of individuals and organizations. The data mining community has responded to this challenge by developing a new breed of algorithms that analyze the data while paying attention to privacy issues.

Mining for associations among items in a large database of sales transaction is an important database mining function.

6. Problem Statement

- Find rules that have support and confidence greater than user-specified minimum support and mínimum confidence

- Apriori algorithm uses frequent item sets, join & prune methods and Apriori property to derive strong association rules. Where as in Frequent-Pattern Growth method avoids repeated database scanning of Apriori algorithm. So, compare these two algorithms and find out which is better.
- Find out the frequent item set from large database accurately and efficiently.

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

1. Data mining

The concept of association rules was popularized particularly due to the 1993 article of Agrawal, which has acquired more than 6000 citations according to Google Scholar, as of March 2008, and is thus one of the most cited papers in the Data Mining field. However, it is possible that what is now called "association rules" is similar to what appears in the 1966 pape on GUHA, a general data mining method developed by Petr Hajek.

Association rule mining has been an active research area since its introduction [1]. Various algorithms have been proposed to improve the performance of mining association rules and frequent itemsets. An interesting direction is the development of techniques that incorporate privacy concerns. One type of these techniques is perturbation based, which perturbs the data to a certain degree before data mining so that the real values of sensitive data are obscured while statistics properties of the data are preserved. An early work of Agrawal and Srikant proposed a perturbation based approach for decision tree learning [2].

The problem of discovering association rules between items in a large database of sales transactions. There are two new algorithms for solving this problem that are fundamentally different from the known algorithms. Empirical evaluation shows that these algorithms outperform the known algorithms by factors ranging from three for small problems to more than an order of magnitude for large problems. Also show how the best features of the two proposed algorithms can be combined into a hybrid algorithm, called AprioriHybrid. Scale-up experiments show that AprioriHybrid scales linearly with the number of transactions. AprioriHybrid also has excellent scale-up properties with respect to the transaction size and the number of items in the database [3]. The problem of discovering association rules has received considerable research attention and several fast algorithms for mining association rules have been developed. In practice, users are often interested in a subset of association rules [3].

Some recent work [5,6,7,8] investigates the tradeoff between the extent of private information leakage and the degree of data mining accuracy. One problem of perturbation based approach is that it may introduce some false association rules. Another drawback of this approach is that it cannot always fully preserve privacy of data while achieving precision of mining results, the effect of the amount of perturbation of the data on the accuracy of mining results is unpredictable.

2. Association rule mining

It is one of the first to introduce hierarchy/ taxonomy into association rule learning. It focused on databases of transactions and is-a hierarchies, formally defined the concept of generalized association rule and designed and compared couple of rule mining algorithms based on frequent itemsets and Apriori candidate generations [9]. The other one of the first in the transactional database domain to incorporate the concept of hierarchy/taxonomy into association rule mining. It detailed a progressive deepening rule mining algorithm and variants based on frequent itemset and apriori strategies, which was aimed at finding multiple-level rules at same levels. The paper also had discussions on considering mixed hierarchies and learning rules involving different level concepts in a hierarchy [10]. Introduced the idea of quantitative association rules, an extension of generalized association rules considering both quantitative and categorical attibutes, in the domain of large relational tables. The paper emphasized on how to partition a quantitative attribute according to a data-dependant information loss measure -- partial completeness, and gave a rule mining algorithm which would combine adjacent partitions as necessary [11].

Introduces an approach to find all maximal frequent itemsets, which were frequent itemsets of maximal size, i.e., extending them into any supersets would result in non-frequent itemsets. Given a complete set of maximal frequent itemsets, data miners could generate all other frequent itemsets by taking non-empty subsets of them. As compared with traditional approaches to find all frequent item, finding maximal frequent itemsets were proved to be computationally much cheaper [12].

S. Brin, R. Motwani, and C. Silverstein pointed out association rules were estimating conditional probabilities of positive events, while the underlying correlation might be either positive or negative. By generalizing association rules to correlations, both positive and negative relations would be accounted for. Chi-squared test on contingency tables was proposed as a measure of correlation [13].

R. Srikant, Q. Vu, and R. Agrawal provided mechanisms for users to specify constraints when mining association rules. Boolean expressions over the presence or absence of items, as well as their descendants and/or antecedents if hierarchies are considered, could be inputted into the rule mining algorithms presented in this paper, thus execution time might be largely reduced as compared to mining complete set of rules and applying post-processing filtering [14].

This paper was written from the database management perspective and presented a notation "query flock", which consisted of a parameterized query and a filter that selects values for the parameters by applying a condition to the query results. Apriori technique developed from association rule mining was generalized to apply to such query flocks [15].

J. Hipp, A. Myka, R. Wirth, and U. Guntzer presented an algorithm called Prutax, which combined several previous frequent itemset mining optimizations, as a way to discover generalized frequent itemsets faster. It was still locked in the traditional framework of "finding frequent itemset first". However, it did not take into consideration rules that could learn in depth in hierarchies, and further redundancy issues related to such rules [16].

Introduces an approach to find all maximal frequent itemsets, which were frequent itemsets of maximal size, i.e., extending them into any supersets would result in non-frequent itemsets. Given a complete set of maximal frequent itemsets, data miners could generate all other frequent itemsets by taking non-empty subsets of them. As compared with traditional approaches to find all frequent item, finding maximal frequent itemsets were proved to be computationally much cheaper [17].

Bayardo introduces an approach to find all maximal frequent itemsets, which were frequent itemsets of maximal size, i.e., extending them into any supersets would result in non-frequent itemsets. Given a complete set of maximal frequent itemsets, data miners could generate all other frequent itemsets by taking non-empty subsets of them. As compared with traditional approaches to find all frequent items, finding maximal frequent itemsets were proved to be computationally much cheaper [18].

Zaki and Hsiao discuss the redundancy of traditional frequent itemsets based association rule generation framework, and introduced the concept of closed frequent itemset and corresponding mining algorithms. A framework based on closed frequent itemsets could cut redundancy caused by traditional means dramatically, and without loss of information (whereas maximal frequent itemsets would). This presents the case where attributes are treated as categorical only, i.e., with 2-level hierarchies [19].

Perng and Wang et al. presents the idea of learning frequent itemsets off transactional data rendered in various ways from original relational data: partitioning attributes into grouping and itemizing categories disjointly, and mining frequent itemsets based on these different settings. A lattice captures the partial ordering of these partitioning settings and thus improves efficiency. User-defined inner/inter-attribute constraints can be described by a language provided [20].

3. Apriori Algorithm

The most representative method of associations was proposed as Apriori algorithm [21] by Agrawal et al. in 1994. In Apriori algorithm process, two steps are included. Fist, it finds out the satisfied frequent itemsets of minimum support; second, it finds out the satisfied rules of minimum confidence. In other words, we are used to find out information of frequent itemset and mine all association rules. Apriori algorithm is continuously repeated to scan database, find out all frequent itemsets, until it does not produce new candidate itemsets. Apriroi algorithm does not filter prior candidate itemsets, so that reduces the amount of candidate itemsets to scan. Therefore, it needs many times to complete scanning a database. In implementing efficiency, Apriori algorithm is not completely efficient.

Among the many mining algorithms of association rules, Apriori Algorithm is a classical algorithm that has caused the most discussion; it can effectively carry out the mining association rules. However, based on Apriori Algorithm, most of the traditional algorithms existed “item sets generation bottleneck” problem, and are very time-consuming. An enhance algorithm associating which is based on the user interest and the importance of itemsets is put forward by the paper, incorporate item that user is interested in into the itemsets as a seed item, then scan the database, incorporate all other items which are in the same transaction into itemsets, Construct user interest itemsets, reduce unnecessary itemsets; through the design of the support functions algorithm not only considered the frequency of itemsets, but also consider different importance between different itemsets. And also reduces the storage space, improves the efficiency and accuracy of the algorithm [22].

In the large database of customer transactions, each transaction consists of items purchased by a customer in a visit. An efficient algorithm that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel estimation and pruning techniques. Also present results of applying this algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm. [23]

Based on association analysis, an improved algorithm of Apriori is the main ideas of the algorithm are:

- Count the probability of each attribute item(A1 , A2,…Am) of a DB by scanning the DB first time
- The probability of any two items Ak and Am appeared synchronously in one record is Pkm.

$min(Pk , Pm) \leq Pkm \leq Pk *Pm$,

if Ak and Am is total correlation, then the Pkm is the minimum of the Pk and Pm,;

if Ak and Am is total independent, then the Pkm is Pk *Pm;

So we can estimate:

$Pkm =(a*min(Pk, Pm)+b*Pk*Pm)/(a+b); a+b=1$

Parameter “a” is the probability while Ak and Am are total correlation,

Parameter “b” is the probability while Ak and Am are total independent,

Parameter “a” and “b” can use other method such as association analysis to count. In this paper a method for calculate the parameter “a” and”b” with association analysis is provided.

If Pkm is more than the threshold value which the user set, then Ak, Am are the frequent itemsets.

We can use the method which described above to find out all the frequent itemsets without scanning DB so many times.

- Count the support of the frequent itemsets by scanning the DB another time;
- Output the association rules from the frequent itemsets.

The detailed algorithm and it's sample are described in the paper [23]. At last we compared it with algorithm apriori. The best quality is that the algorithm in our paper reduce the times of scanning DB[23].

4. FP Growth Algorithm

One of the currently fastest and most popular algorithms for frequent item set mining is the FP-growth algorithm. It is based on a prefix tree representation of the given database of transactions (called an FP-tree), which can save considerable amounts of memory for storing the transactions. The basic idea of the FP-growth algorithm can be described as a recursive elimination scheme: in a preprocessing step delete all items from the transactions that are not frequent individually, i.e., do not appear in a user-specified minimum number of transactions. Then select all transactions that contain the least frequent item (least frequent among those that are frequent) and delete this item from them. Recurs to process the obtained reduced (also known as projected) database, remembering that the item sets found in the recursion share the deleted item as a prefix. On return, remove the processed item also from the database of all transactions and start over, i.e., process the second frequent item etc. In these processing steps the prefix tree, which is enhanced by links between the branches, is exploited to quickly find the transactions containing a given item and also to remove this item from the transactions after it has been processed.

The new concept of FP-growth algorithm was proposed by Han et al.[24], it can be one of the representations of the itemsets which do not require candidate generations. It does not need association length to proceed phases which generate candidate itemsets in Apriori algorithm. However, mining with Apriori algorithm does not archive the goal efficiently because it may need many times to scan database and generate a lot of candidate itemsets. Therefore, FP-growth proceeds the first scan in transaction database, it then later filters the frequent itemsets and gradually increases support. Next, in the second scan, establish a FP-tree structure by the transaction database. Then, use a Header table to allocate each item node in FPtree, each item of tree will link each other. Last, a Header table mines conditional pattern tree which finds out all frequent itemsets in recursive method. It is a very efficient and memory saving algorithm [25].

Chapter III

Implementation

1. Ariori Algorithm

The algorithm is based on following 2 points.

1. Find all frequent itemsets:
   - Get frequent items:
     - Items whose occurrence in database is greater than or equal to the min.support threshold.
   - Get frequent itemsets:
     - Generate candidates from frequent items.
     - Prune the results to find the frequent itemsets.
2. Generate strong association rules from frequent itemsets
   - Rules which satisfy the minimum support and minimum confidence threshold.

The pseudo code for frequent itemset generation is:

```
$C_k$: Candidate itemset of size k
$L_k$ : frequent itemset of size k
$L_1$ = {frequent items};
for ($k$ = 1; $L_k$ !=   ; $k$++) do begin
     $C_{k+1}$ = candidates generated from $L_k$;
    for each transaction $t$ in database do
       increment the count of all candidates in $C_{k+1}$ that t contains
    $L_{k+1}$  = candidates in $C_{k+1}$ with min_support
    end
return the set of frequent itemsets;
```

High Level Design

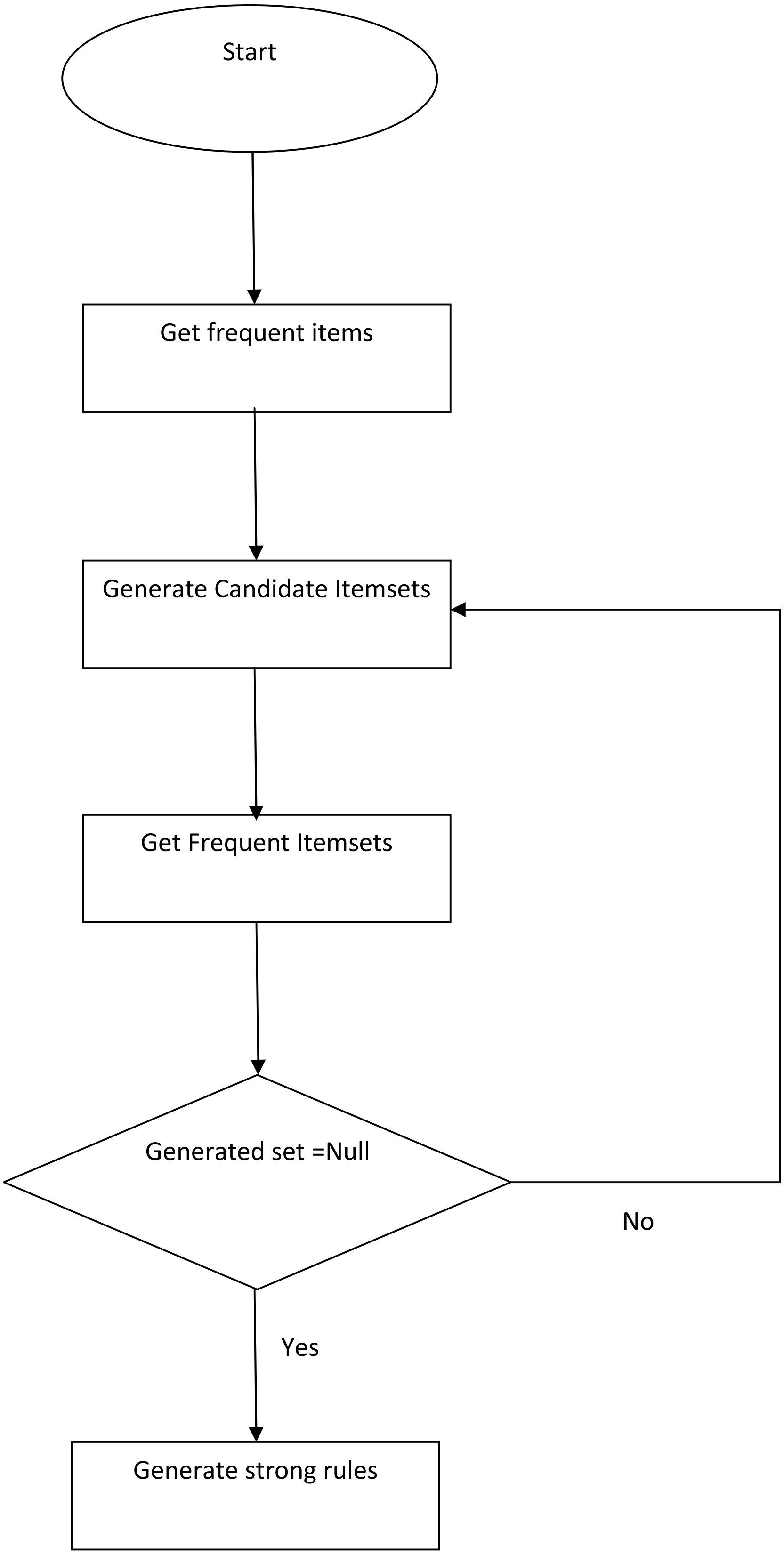
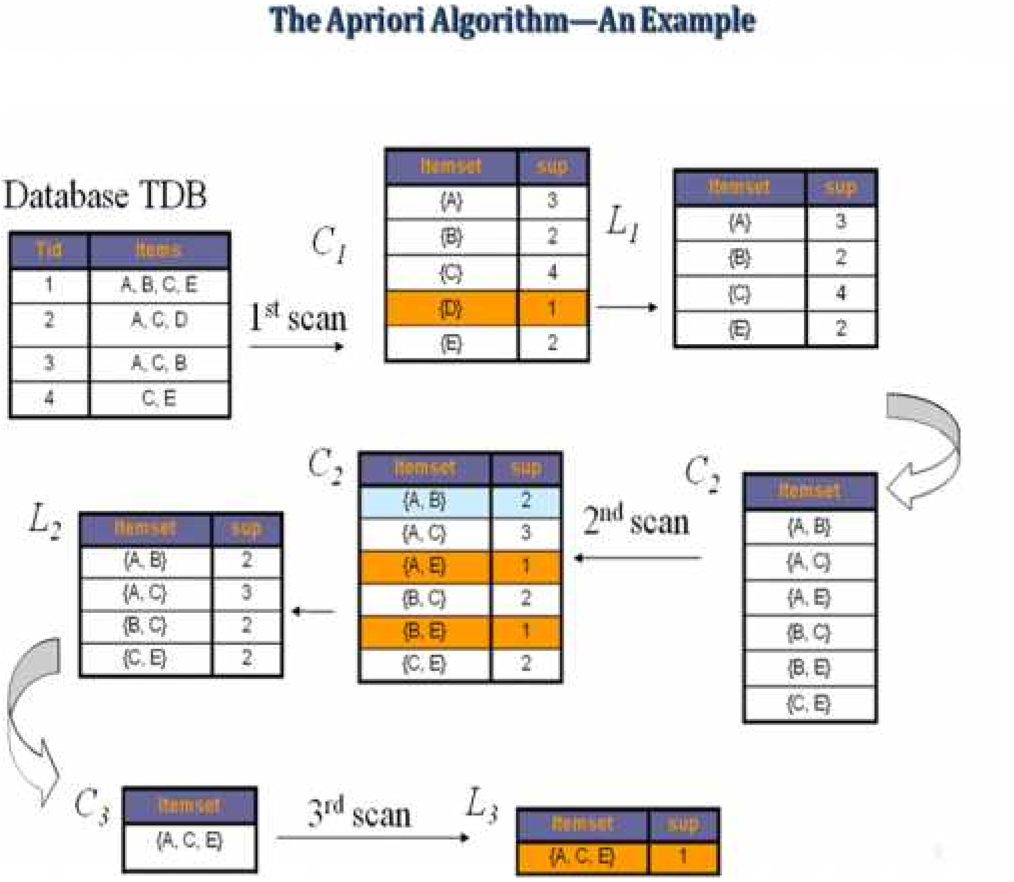


Figure 1: High Level Design

1.1 Apriori Algorithm an example



The Apriori Algorithm—An Example

Tid	Items
1	A, B, C, E
2	A, C, D
3	A, C, B
4	C, E

Itemset	sup
{A}	3
{B}	2
{C}	4
{D}	1
{E}	2

Itemset	sup
{A}	3
{B}	2
{C}	4
{E}	2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

Itemset	sup
{A, B}	2
{A, C}	3
{A, E}	1
{B, C}	2
{B, E}	1
{C, E}	2

Itemset	sup
{A, B}	2
{A, C}	3
{B, C}	2
{C, E}	2

Itemset
{A, C, E}

Itemset	sup
{A, C, E}	1

Figure 2: Apriori Algorithm an example

2. FP-Growth

FP-Growth allows frequent itemset discovery without candidate itemset generation. Two step approach:

1. Build a compact data structure called the FP-tree
   - Scan data and find support for each item.
   - Discard infrequent items.
   - Sort frequent items in decreasing order based on their support.

2. Extracts frequent itemsets directly from the FP-tree
   - FP-Growth extracts frequent itemsets from the FP-tree.
   - Bottom-up algorithm - from the leaves towards the root

- Each prefix path sub-tree is processed recursively to extract the frequent itemsets. Solutions are then merge

2.1 FP Growth Algorithm an example

Let us take a dataset:

TID	Items bought
100	{f, a, c, d, g, i, m, p}
200	{a, b, c, f, l, m, o}
300	{b, f, h, j, o}
400	{b, c, k, s, p}
500	{a, f, c, e, l, p, m, n}

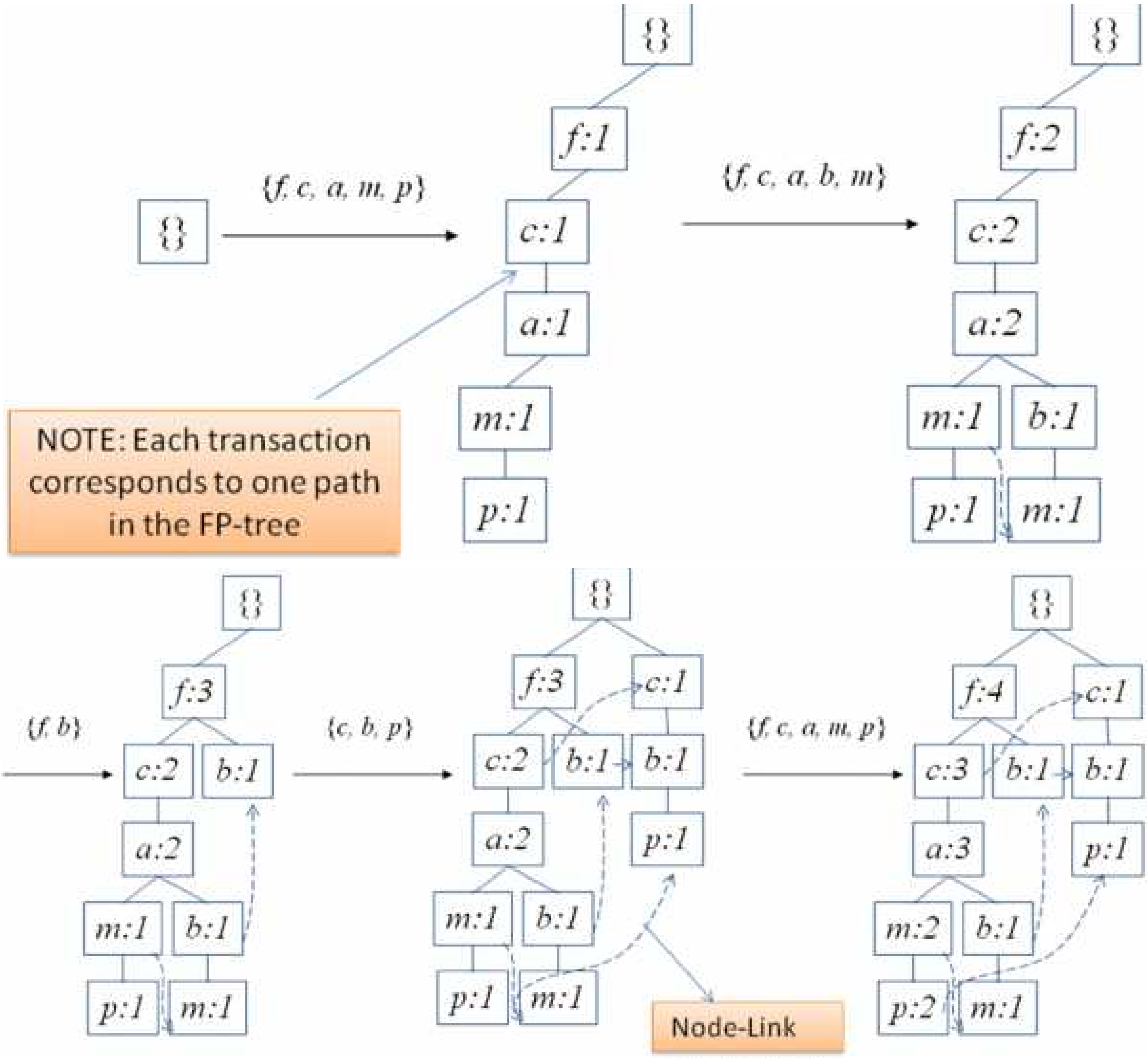
a. Scan DB for the first time to generate L

Item	frequency
f	4
c	4
a	3
b	3
m	3
p	3

b. Scan the DB for the second time, order frequent items in each transaction

Items bought	(ordered) frequent items
{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
{a, b, c, f, l, m, o}	{f, c, a, b, m}
{b, f, h, j, o, w}	{f, b}
{b, c, k, s, p}	{c, b, p}
{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

c. Construct FP-tree



d. Final Tree

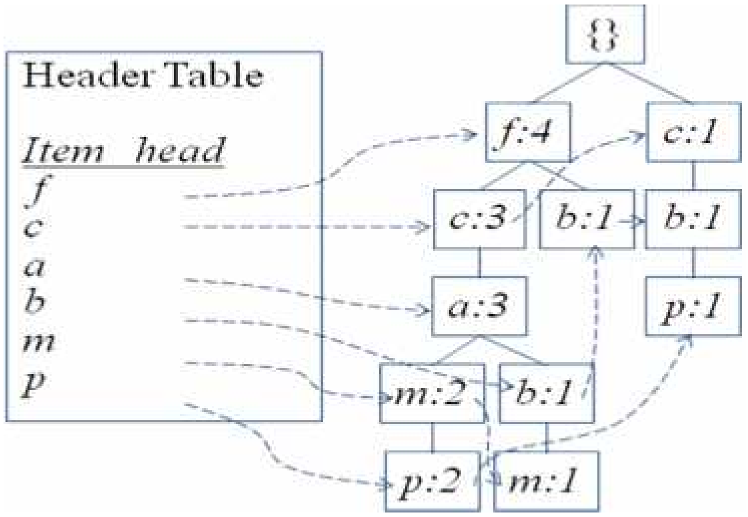


Figure 3: FP - Growth Algorithm an example

3. Apriori VS FP-Growth

Following are some drawbacks of Apriori algorithm.

- Multiple scans on the transaction database
- Huge number of candidates generated and tested
- Tedious workload of support counting for candidates

Ideas for improvement

- Reduce the number transaction database scans
- Shrink number of candidates
- Facilitate support counting of candidates

CHAPTER V

TESTING AND ANALYSIS

1. Algorithm implementation specification

Algorithms have been implemented into Visual Studio 2008. Programming language is C#. And hardware specification is as same as the specification needed by Visual Studio 2008.

2. Input

2.1. Following are two different datasets that having 10 different transactions.

Dataset A:

Transaction ID	**List of Items**
TID1	a,b
TID2	a,c
TID3	a,d
TID4	b,c
TID5	c,d
TID6	a,b,c
TID7	a,b,e
TID8	a,b,c,d
TID9	a,b,d
TID10	a,d,e

Dataset B:

Transaction ID	**List of Items**
TID1	a,b
TID2	a,b,c
TID3	a,c,d
TID4	b,c
TID5	c,d,e
TID6	a,b,c,e
TID7	a,b,e,f
TID8	a,b,c,d,f
TID9	a,b,f
TID10	a,d,e,f

Table 1: Transactional dataset

2.2. Minimum Confidence and Minimum support, using two different cases for support and confidence.

	Case 1	Case 2
Min. Support	25%	40%
Min. Confidence	25%	50%

Table 2: Minimum Support and Confidence

3. Output

3.1. Number of association rules extracted by the Apriori algorithm with corresponding support:

Dataset A			
Case 1		Case 2	
Item	Support	Item	Support
a	8	a	8
b	6	b	6
c	5	c	5
d	5	d	5
bc	3	ad	4
ad	4	ab	5
ac	3		
ab	5		

Dataset B			
Case 1		Case 2	
Item	Support	Item	Support
a	8	a	8
b	7	b	7
c	6	c	6
d	4	d	4
e	4	e	4
f	4	f	4
cd	3	bc	4
bf	3	af	4
bc	4	ac	4
af	4	ab	6
ac	4		
ab	6		
abc	3		
abf	3		

Table 3: Number of association rules extracted by the Apriori with corresponding support

3.2. Number of association rules extracted by the FP growth algorithm with corresponding support:

Dataset A			
Case 1		Case 2	
Item	Support	Item	Support
a	8	a	8
b	6	b	6
c	5	c	5
d	5	d	5
bc	3	ab	4
ad	4		
ac	3		
ab	5		
bd	3		

Dataset B			
Case 1		Case 2	
Item	Support	Item	Support
a	8	a	8
b	6	b	6
c	5	c	5
d	5	d	5
bc	3	ad	4
ad	4	ab	5
ac	3		
ab	5		

Table 4: Number of association rules extracted by the FP-growth with corresponding support

3.3. Number of association rules and corresponding confidence by the Apriori algorithm:

Dataset A			
Case 1		Case 2	
Rule	Confidence	Rule	Confidence
a->d	50.00%	a->d	50.00%
a->c	37.50%	a->b	62.50%
a->b	62.50%	b->a	83.33%
b->a	83.33%	d->a	80.00%
b->c	50.00%		
c->a	60.00%		
c->b	60.00%		
d->a	80.00%		

Dataset B			
Case 1		Case 2	
Rule	Confidence	Rule	Confidence
a->e	37.50%	a->f	50.00%
a->d	37.50%	a->c	50.00%
a->f	50.00%	a->b	75.00%
a->c	50.00%	b->a	85.71%
a->bf	37.50%	b->c	57.14%
a->bc	37.50%	c->a	66.67%
a->b	75.00%	c->b	66.67%

ab->c	50.00%	f->a	100.00%
ab->f	50.00%		
af->b	75.00%		
ac->b	75.00%		
b->f	42.86%		
b->ac	42.86%		
b->a	85.71%		
b->c	57.14%		
b->af	42.86%		
bc->c	75.00%		
bf->a	100.00%		
c->ab	50.00%		
c->d	50.00%		
c->b	66.67%		
c->a	66.67%		
d->c	75.00%		
d->a	75.00%		
e->a	75.00%		
f->a	100.00%		
f->ab	75.00%		
f->b	75.00%		

Table 5: Number of association rules and corresponding confidence by the Apriori

3.4. Number of association rules and corresponding confidence by the FP growth algorithm:

Dataset A			
Case 1		Case 2	
Rule	Confidence	Rule	Confidence
a->b	62.50%	a->b	62.50%
b->a	83.33%	b->a	83.33%
b->c	50.00%	d->a	80.00%
c->a	60.00%		
c->b	60.00%		
d->a	80.00%		

Dataset B			
Case 1		Case 2	
Rule	Confidence	Rule	Confidence
a->f	50.00%	a->f	50.00%
a->c	50.00%	a->c	50.00%
a->bf	37.50%	a->b	75.00%
a->bc	37.50%	b->a	85.71%
a->b	75.00%	b->c	57.14%
ab->f	50.00%	c->a	66.67%
af->b	75.00%	c->b	66.67%
ac->b	75.00%	f->a	100.00%
b->f	42.86%		

b->a	85.71%		
b->c	57.14%		
b->af	42.86%		
bc->c	75.00%		
bf->a	100.00%		
c->ab	50.00%		
c->b	66.67%		
c->a	66.67%		
f->a	100.00%		
f->ab	75.00%		
f->b	75.00%		

Table 6: Number of association rules and corresponding confidence by the FP growth

4. Discussion

4.1. Case 1: With Minimum support = 25% and minimum confidence =25%

Dataset Name	Number of item size	Number of Association rules – Apriori	Number of Association rules – FP
A	3	8	10
B	4	12	14
C	5	15	19
D	6	20	25
E	7	26	29
F	8	30	35
G	9	34	38
H	10	39	45
I	11	45	48
J	12	50	54

Table 7: Itemsize and number of association rules with 25% support and confidence

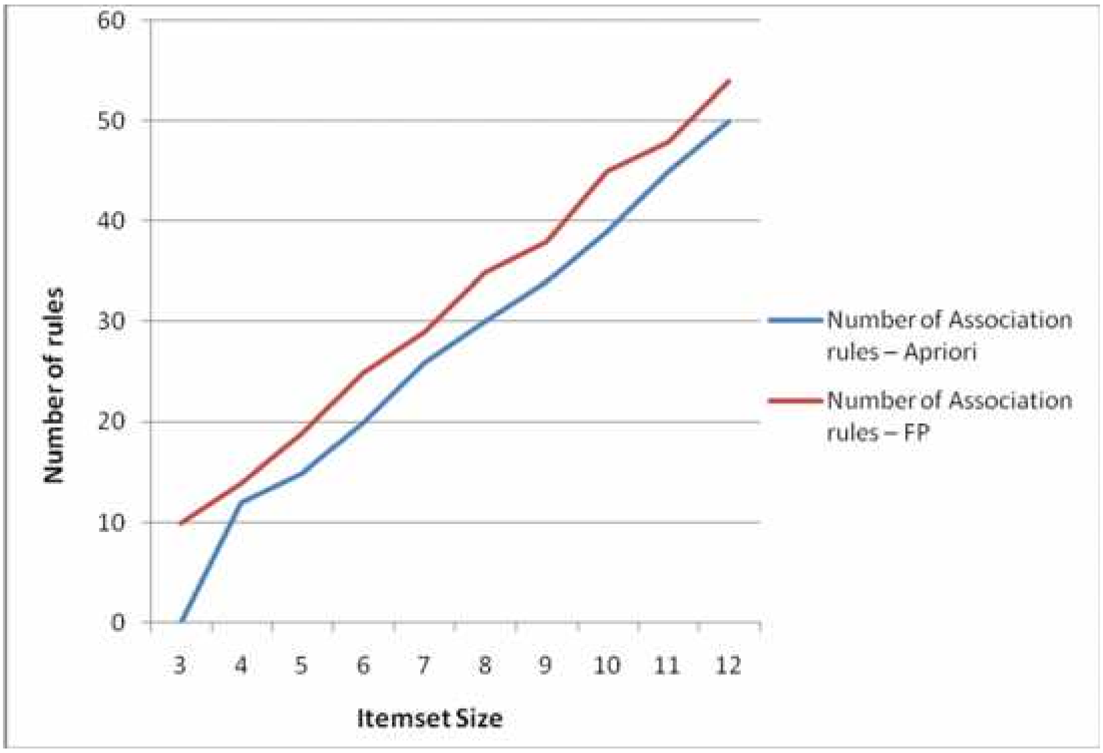


Figure 4: Graph between itemsize and number of association rules with 25% support and confidence

4.2. Case 2: With Minimum support = 40% and minimum confidence = 50%

Dataset Name	Number of items	Number of Association rules – Apriori	Number of Association rules - FP
A	3	6	4
B	4	10	6
C	5	13	7
D	6	17	9
E	7	23	11
F	8	28	14
G	9	33	17
H	10	37	19
I	11	40	23
J	12	44	26

Table 8: Itemsize and number of association rules with 40% support and 50% confidence

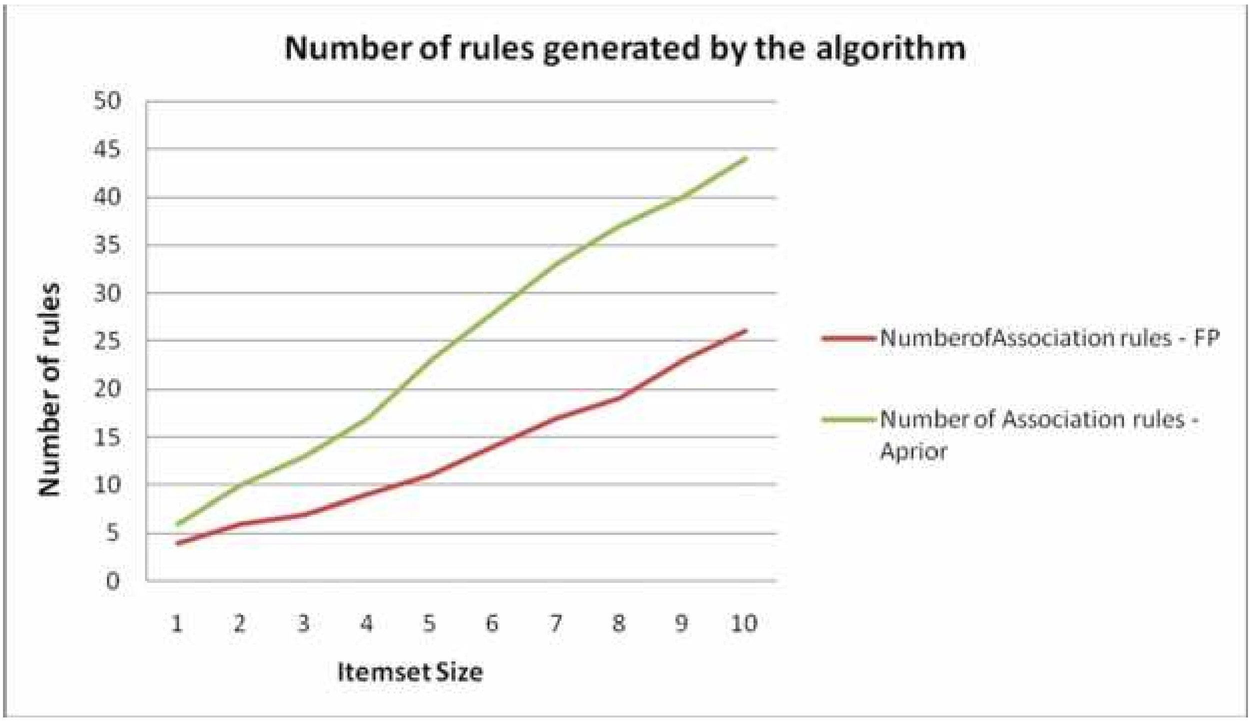


Figure 5: Graph between itemsize and number of association rules with 40% support and 50% confidence

5. Algorithm comparison

5.1. FP - Growth vs. Apriori algorithm

Apriori visits each transaction when generating a new candidate sets where as FP-Growth does not. FP- Growth can use data structures to reduce transaction list. FP-Growth traces the set of concurrent items but Apriori generates candidate sets. FP-Growth uses more complicated data structures & mining techniques.

5.2. Algorithm Analysis results

Above graph shows, with higher support and confidence on both algorithms, FP-Growth extracts the better association rules than Apriori algorithm. While decreasing the support and confidence value Apriori seems better than FP-Growth algorithm.

Besides this thing following are some important points.

2. FP-Growth is not inherently better than Apriori
   a. Intuitively, it appears to condense data
   b. Mining scheme requires some new work to replace candidate set generation
   c. Recursion obscures the additional effort
3. FP-Growth may run faster than Apriori in circumstances
   a. No guarantee through complexity which algorithm to use for efficiency

CHAPTER V

SUMMARY AND FURTHER WORK

Data mining process consists of some basic steps that are performed before carrying out data mining, such as data selection, data cleaning, pre-processing, and data transformation. Association rule techniques are used for data mining if the goal is to detect relationships or associations between specific values of categorical variables in large data sets. There may be thousands or millions of records that have to be read and to extract the rules. Frequent pattern mining is a very important task in data mining. The approaches applied to generate frequent set generally adopt candidate generation and pruning techniques for the satisfaction of the desired objectives. This dissertation shows how the different approaches achieve the objective of frequent mining along with the complexities required to perform the job. This dissertation looks into a comparison among Apriori and FP Growth algorithm. The process of the mining is helpful in generation of support systems for many computer related applications. It has been observed that with higher support and confidence on both algorithms, FP-Growth extracts the better association rules than Apriori algorithm. While decreasing the support and confidence value Apriori seems better than FP-Growth algorithm. Also we found that FP-Growth is not inherently better than Apriori, it depends upon the circumstances.

FP-growth is more better then the Apriori because of no candidate generation, no candidate test, and use compact data structure. Also eliminate repeated database scan. Mining frequent itemsets for the association rule mining from the large transactional database is a very crucial task. All of the previous studies were using Apriori approach and FP-Tree approach for extracting the frequent itemsets, which have scope for improvement. This chapter summarizes the work done in this thesis and then the future scope is given.

- Using constraints can further reduce the size of itemsets generated and improve mining efficiency.

Chapter VI
References

1. R. Agrawal; T. Imielinski; A. Swami: *Mining Association Rules Between Sets of Items in Large Databases,* SIGMOD Conference 1993
2. Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. *Algorithms for association rule mining - A general survey and comparison*. SIGKDD Explorations, 2(2):1-58, 2000.
3. R. Agrawal and R. Srikant, *"Fast Algorithms for Mining Association Rules", Proc. of the 20th Int'l Conf. on Very Large Databases,* Santiago, Chile, Sep. 1994.
4. R. Srikant, Q. Vu, R. Agrawal, *Mining Association Rules with Item Constraints*, Proc. of the 3rd Int'l Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, California, August, 1997
5. A. Evfimievski, R. Srikant, R. Agrawal and J. Gehrke, "Privacy Preserving Mining of Association Rules", *Proc. of the 8th ACM SIGKDD Int'l Conference on Knowledge Discovery in Databases and Data Mining*, Edmonton, Canada, July 2002
6. M., Bertino, Elmagarmid, Ibrahim & Verykios 1999
7. Oliveira, S. & Zaiane 2003,
8. Saygin, Y., Verykios & Clifton 2001
9. R. Srikant and R. Agrawal. Mining Generalized Association Rules. In *Proc. of the 21st Int'l Conference on Very Large Databases*, Zurich, Switzerland, September 1995.
10. J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *Proc. of the 21st Int'l Conference on Very Large Databases*, Zurich, Switzerland, September 1995.
11. R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In Proceedings of the *ACM SIGMOD Conference on Management of Data*, Montreal, Canada, June 1996.
12. D. Gunopulos, H. Mannila, and S. Saluja. Discovering all the most specific sentences by randomized algorithms. In *Intl. Conf. on Database Theory*, January 1997.

13. S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. *SIGMOD Record* (ACM Special Interest Group on Management of Data), 26(2):265, 1997.
14. R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. In Proc. 3rd *Int. Conf. Knowledge Discovery and Data Mining*, pages 67--73, 1997.
15. D. Tsur, J. D. Ullman, S. Abitboul, C. Clifton, R. Motwani, and S. Nestorov. Query flocks: A generalization of association-rule mining. In Proc. 1998 *ACM-SIGMOD,* pp 1--12.
16. J. Hipp, A. Myka, R. Wirth, and U. Guntzer. A new algorithm for faster mining of generalized association rules. In Proc. 2nd *PKKD*, 1998.
17. D. Lin and Z. Kedem. Pincer-search: A new algorithm for discovering the maximum frequent set. In 6th *Intl. Conf. Extending Database Technology*, March 1998.
18. R. Bayardo. Efficiently mining long patterns from databases. In *ACM SIGMOD Conf. Management of Data*, June 1998.
19. M. Zaki and C. Hsiao. CHARM: An Efficient Algorithm for Closed Itemset Mining, 2nd *SIAM International Conference on Data Mining*, Arlington, April 2002.
20. C. Perng, H. Wang, S. Ma and J. Hellerstein. Discovery in Multi-attribute Data with User-defined Constraints, *ACM SIGKDD Explorations Newsletter, Volume4, Issue 1*, Pages: 56 - 64, June 2002.
21. Agrawal, R. and Srikant, R., “Fast algorithms for mining association rules,” in Proceedings of 1994 International Conference on Very Large Data Bases, pp.487-499, 1994.
22. Lei Ji, Baowen Z, Jianhua Ji, "A New Improvement on Apriori Algorithm", IEEE Proceedings, Shanghai, 2006
23. Rakesh Agrawal , Tomasz Imieliński , Arun Swami, Mining association rules between sets of items in large databases, Proceedings of the 1993 ACM SIGMOD international conference on Management of data, p.207-216, May 25-28, 1993, Washington, D.C., United States
24. Han, J., Pei, J., and Yin, Y., “Mining *Frequent Patterns without Candidate Generation,*” in Proc. ACM SIGMOD Int. Conf. on Management of Data, pp, 1-12, 2000.
25. Tan. P. N., Michael Mteinbach, Vipin Kumar, *Introduction to Data Mining*, NY: Addison Wesley, 2005