

Efficient Techniques for Mining Association Rules: - A Comparative Study

S . Ramaiah
Asst.Professor
Dept. of CSE,KMMITS
TIRUPATHI

N . Jaya Krishna
Asst.Professor,
Dept. of CSE ,SREC
TIRUPATHI

Y . Suresh
Asst.Professor,
Dept. of CSE,KMMITS
TIRUPATHI

Abstract: The amount of data is being stored for analyzation is gradually increasing with the advancement of IT technologies. It has resulted in large amount of data stored in databases. Thus the data mining plays major role in analyzing the databases to extract the required data, predicting unknown patterns and forming Association rules. In data mining, Association rule mining becomes one of the important tasks of descriptive technique which can be defined as discovering meaningful patterns form large collection of data. Mining Frequent Itemset is very fundamental part of association rule mining. Many algorithms have been proposed from last many decades including horizontal layout based techniques, vertical layout based techniques, and projected layout based techniques. But most of the techniques suffer from repeated database scan, and more number of Candidate generation in Apriori algorithm. In Fp-Tree algorithm, the memory consumption is more and it forms a tree. By using Mining Frequent Itemset Algorithm, it can generate Associations. As in medical retailer industry many transactional databases contain same set of transactions many times, to apply this thought, the present work compares the three popular algorithms namely Apriori algorithm, Fp-Tree algorithm and Mining Frequent Itemset algorithm. The present work observes the Database Scan, Candidate set generation, Conditional FP-Trees and Conditional Pattern Trees gives better performance in MFI over Apriori algorithm and FP-Tree algorithms.

Keywords: data mining, Association rules, candidate generation, Apriori, Conditional Fp-tree, MFI.

1. INTRODUCTION

Data mining has attracted a great deal of attention in the information industry and in society as a whole in recent years , due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration.

Data mining can be viewed as a result of the natural evolution of information technology. The

abundance of data, coupled with the need for powerful data analysis tools, has been described as a data rich but information poor situation. The fast-growing amount of data, collected and stored in large and numerous data repositories, has far exceeded our human ability for comprehension without power tools. Consequently, important decisions are often made based not on the information – rich data stored in data repositories, but rather on a decision maker’s intuition, simply because the decision maker does not have the tools to extract the valuable knowledge embedded in the vast amounts of data. In addition, considered expert system technologies, which typically rely on users or domain experts to manually input knowledge into knowledge bases. But, this procedure is prone to biases and errors, and is extremely time-consuming and costly. Data mining tools perform data analysis and may uncover important data patterns, contributing greatly to business strategies, knowledge bases, and scientific and medical research. The widening gap between data and information calls for a systematic development of data mining tools that will turn to knowledge.

1.1 Data Mining

Data mining refers to extracting or mining knowledge from large amounts of data. Thus data mining should have been more appropriately named knowledge mining from data. Knowledge mining may not reflect the emphasis on mining from large amounts of data. Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - that can be used to increase revenue, cuts costs, or both. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Many other terms carry a similar or slightly different meaning to data mining, such as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging.

1.2 Data mining functionalities

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general, data mining tasks can be classified into two categories: descriptive and predictive. Descriptive mining

tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make predictions.

Data mining system should allow users to specify hints to guide or focus the search for interesting patterns. Because some patterns may not hold for all of the data in the database, a measure of certainty is usually associated with each discovered pattern.

1.2.1 Concept/Class Characterization and Description

Data can be associated with classes or concepts. For example, in the Medical retailer store, classes of items for sale include tablets and injections, and concepts of customers include patients. It can be useful to describe individual classes and concepts in summarized, concise, and yet precise terms. Such description of a class or concept is called class/concept description. These descriptions can be derived from

- Data characterization, by summarizing the data of the class under study (often called the target class)
- Data discrimination, by comparison of the target class with one or a set of comparative classes (often called the contrasting classes).Or
- Both characterization and discrimination.

1.2.2 Mining frequent patterns, Associations

Frequent patterns, as the name suggests, are patterns that occur frequently in data. There are many kinds of frequent patterns, including Itemset, subsequences and substructures. A frequent Itemset typically refers to a set of items that frequently appear together in a transactional dataset, and database as medical retailer sales in that one transaction is Axeruff and Paracetemal A frequently occurring sub sequence such as the pattern that customers tend to purchase first Axeruff followed by a Paracetemal and the a Acetuff is a (frequent) sequential pattern, a substructure can refer to different structural forms, such as graphs, trees, or lattices, which may be combined with item sets or subsequences. If a substructure occurs frequently, it is called a (frequent) structured pattern. Mining frequent patterns leads to the discovery of interesting associations and correlations within the data.

1.2.3 Classification and prediction

Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of the set of training data. Basically classification is a Two-step process, the first step is supervised learning for the sake of the predefined class label for training data set. Second step is classification accuracy evaluation. Likewise data prediction is also Two-step process. Before the classification and prediction, something should be done beforehand like data cleaning, relevance analysis, data transformation and reduction.

Numeric prediction is the task of predicting continuous (or ordered) values for given input. Some classification techniques (such as back propagation, support vector machines,

and k-nearest neighbor classifiers) can be adapted for prediction.

1.2.4 Cluster Analysis

Statistical classification technique in which cases, data, or objects (events, people, things, etc.) are subdivided into groups (clusters) such that the items in a cluster are very similar (but not identical) to one another and very different from the items in other clusters. It is a discovery tool that reveals associations, patterns, relationships, and structures in masses of data. Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

1.2.5 Outlier Analysis

Outliers can occur by chance in any distribution, but they are often indicative *either* of measurement error or that the population has a heavy-tailed distribution. A database may contain data objects that do not comply with the general behavior or model of the data. These data objects are outliers. Most data mining methods discard outliers as noise and exceptions. In the former case one wishes to discard them or use statistics that are robust to outliers, while in the latter case they indicate that the distribution has high kurtosis and that one should be very cautious in using tools or intuitions that assume a normal distribution. A frequent cause of outliers is a mixture of two distributions, which may be two distinct sub-populations, or may indicate 'correct trial' versus 'measurement error' this is modeled by a mixture model.

1.2.6 Evolution Analysis

Data evolution analysis describes and models regularities or trends for objects whose behavior changes over time. Although this may include characterization, discrimination, association and correlation analysis, classification, prediction, or clustering of time related, distinct features of such an analysis include time series data analysis, sequence or periodicity pattern matching, and similarity based pattern analysis. Example, Time-series data. If the stock market data (time-series) of the last several years available from the New York Stock exchange and one would like to invest in shares of high tech industrial companies. A data mining study of stock exchange data may identify stock evolution regularities for overall stocks and for the stocks of particular companies. Such regularities may help predict future trends in stock market prices, contributing to one's decision making regarding stock investments.

1.3 Frequent Itemset Mining

The space of items in a transactional database gives rise to a subset lattice. The itemset lattice is a conceptualization of the search space when mining frequent itemsets. There are then basically two types of algorithms to mine frequent itemsets, breath-first

algorithms and depth-first algorithms. The breath-first algorithms, such as Apriori [1, 2] and It uses a breadth-first search strategy to count the support of itemsets and uses a candidate generation function which exploits the downward closure property of support [3], apply a bottom-up level-wise search in the Itemset lattice. Candidate itemsets with $K+1$ items are only generated from frequent itemsets with K items. For each level, all candidate itemsets are tested for frequency by scanning the database. On the other hand, depth-first algorithms search the lattice bottom-up in “depth-first “way to generate strong rules frequent items are found.

1.4 Association rules

Association rules are one of the major techniques of data mining. 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 [4]. 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. First introduced the association rules mining in [1, 2, 3]. Many studies have been conducted to address various conceptual, implementation, and application issues relating to the association rules mining task.

Researcher in application issues focuses on applying association rules to a variety of application domains. For example: Relational Databases, Data Warehouses, Transactional Databases, and Advanced Database Systems (Object-Relational, Spatial and Temporal, Time-Series, Multimedia, Text, Heterogeneous, Legacy, Distributed, and WWW) [5].

The task of association rules mining is to find all strong association rules that satisfy a minimum support (min_sup) threshold and a minimum confidence (min_conf) threshold. The process for mining association rules consists of two phases. In the first phase, all frequent itemsets (FI) that satisfy the min_sup are found. In the second phase, strong association rules are generated from the frequent itemsets found in the first phase. Most research considers only the first phase because once frequent itemsets are found, mining association rules is trivial.

1.5 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 of dataming into their processes are banking, insurance, retail and telecom More lately it has been implemented in pharmaceuticals, health, government and all sorts of e-businesses.

One describes a scheme to generate a whole set of trading strategies that take into account application constraints, for example timing, current position and pricing [6]. The authors highlight the importance of developing a suitable back testing 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.

Apriori algorithm is used as a 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 [4].

A government application is proposed by [5]. 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 action ability 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.

2. PROBLEM STATEMENT

The problem of mining frequent itemsets arises in the large transactional databases when there is need to find the association rules among the transactional data for the growth of business. Mining frequent Itemset is the very crucial task to find the association rules between the various items. In the market industry, every one wants to enhance the business thus it is very important to find out the items which are more frequently sale or purchase. Once the selling or purchasing trend of the customer is known then one can easily provide the good services to customer with result in enhancing the business. As it is very common in retail selling database that two or more items sell or purchase together many times therefore database contains same set of items many times, by using this concept aim to overcome the limitation of above existing approaches and a novel approach to mine frequent itemsets from a large transactional database without the candidate with the help of existing techniques.

Studies of frequent item set (or pattern) mining is acknowledged in the data mining field because of its broad applications in mining association rules, correlations and graph pattern constraint based on frequent patterns, sequential patterns and many other data mining tasks. For predicting the frequent items we using

basic popular algorithms. Efficient algorithms for mining frequent itemsets are crucial for mining association rules as well as for many other data mining tasks. We predict the frequent patterns based on the threshold values, those are minimum support of user defined and support of database (count) as well as minimum confidence.

In paper work, considered database as medical retailer of one month transaction for predicting the frequents of items and their associations. Market basket database is the most commonly database use to finding the frequent items. Medical transactional database and minimum support threshold is given, therefore the problem is to find the complete set of frequent itemsets from transactional type of databases to increase the business, so that relation between customers behavior can be found between various items (drugs). The medical retailer transactions of the one month transaction period are taken for predicting.

In Apriori algorithm, we use the input database as medical retailer transactional database is used to predict the frequent items, not only frequent items but also focused on predicting number of scans of data is required, and numbers of conditional trees as well as number of conditional pattern bases are required. While using medical retailer transaction database, how the Apriori algorithm works (run).

Fp-tree algorithm is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth. In FP-Tree algorithm using an extended prefix- tree structure for storing compressed and crucial information about frequent patterns named frequent pattern tree (FP-Tree), so we use the input as medical retailer database transactions, predict the trees and frequent patterns.

Mining Frequent Itemset algorithm is used for predict the association rules and also frequent patterns. We use the medical retailer transaction database as input of Mining Frequent Itemset algorithm

We compare the basic algorithms (Apriori, FP-Tree and MFI algorithm) for medical retailer transactional database predicting the frequent items and Associations in MFI algorithm over the Apriori and Frequent Pattern (FP-Tree) algorithms.

3. APRIORI, FP-TREE AND MFI ALGORITHMS

3.1 APRIORI ALGORITHM

The first algorithm for mining all frequent itemsets and strong association rules was the AIS algorithm by [10]. Then, the algorithm was improved and renamed as 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. The working of Apriori algorithm is fairly depends upon the Apriori property which states that "All nonempty subsets of a frequent itemsets must be frequent"

[12]. It also described the anti monotonic property which says if the system cannot pass the minimum support test, all its supersets will fail to pass the test [10 12]. Therefore if the one set is infrequent then all its supersets are also frequent and vice versa. This property is used to prune the infrequent candidate elements. 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 L_1 . 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, L is used to find L_1 , the set of frequent 2-itemsets, which is used to find L_1 , and so on, until no more frequent k-itemsets can be found. The feature first invented by [12] in Apriori algorithm is used by the many algorithms for frequent pattern generation. The basic steps to mine the frequent elements are as follows [10]:

1. Generate and test: In this first find the 1-itemset frequent elements L by scanning the database and removing all those elements from C which cannot satisfy the minimum support criteria.
2. Join step: To attain the next level elements C_K join the previous frequent elements by self join i.e. L_{K-1} known as Cartesian product of L_{K-1} . i.e. 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.
3. Prune step: C_K is the superset of L_K so members of C_K may or may not be frequent but all K-1 frequent itemsets with the help of Apriori property. i.e. This step eliminates some of the candidate K-itemsets using the Apriori property 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 . Step 2 and 3 is repeated until no new candidate set is generated.

To illustrate this, suppose n frequent 1-itemsets and minimum support is 1 then according to Apriori will generate $n^2 + (n-2)$ candidate 2- itemset (n-3) candidate 3-itemset and so on. The total number of candidates generated is greater than $\sum_{K=1}^n (n-K)$ therefore suppose there are 1000 elements then 1499500 candidates are produced in 2 itemset frequent and 166167000 are produced in 3 –itemset frequent [13].

It is no doubt that Apriori algorithm successfully finds the frequent elements from the database. But as the dimensionality of the database increase with the number of items then:

- More search space is needed and I/O cost will increase.
- Number of database scan is increased thus candidate generation will increase results in increase in computational cost.

Therefore many variations have been takes place in the Apriori algorithm to minimize the above limitations arises due to increase in size of database.

The algorithms improve the Apriori algorithms by:

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

3.2 FP-TREE ALGORITHM

FP-Tree algorithm [4 14] is based upon the recursively divide and conquers strategy; first the set of frequent 1-itemset and their counts is discovered. With start from each frequent pattern, construct the conditional pattern base, then its conditional FP-tree is constructed (which is a prefix tree.). Until the resulting FP-tree is empty, or contains only one single path. (Single path will generate all the combinations of its sub-paths, each of which is a frequent pattern). The items in each transaction are processed in L order. (i.e. items in the set were sorted based on their frequencies in the descending order to form a list).

Conditional pattern base

A “sub database” which consists of the set of prefix paths in the FP-tree co-occurring with suffix pattern.eg for an itemset X, the set of prefix paths of X forms the conditional pattern base of X which co-occurs with X [14].

Steps of FP-tree:

1. Create root of the tree as a “null”.
2. After scanning the database D for finding the 1-itemset then process the each transaction in decreasing order of their frequency.
3. A new branch is created for each transaction with the corresponding support.
4. If same node is encountered in another transaction, just increment the support count by 1 of the common node.
5. Each item points to the occurrence in the tree using the chain of node-link by maintaining the header table.

After above process mining of the FP-tree will be done by Creating Conditional (sub)pattern bases:

1. Start from node constructs its conditional pattern base.
2. Then, Construct its conditional FP-tree & perform mining on such a tree.
3. Join the suffix patterns with a frequent pattern generated from a conditional FP-tree for achieving FP-growth.
4. The union of all frequent patterns found by above step gives the required frequent itemset.

3.3 MINING FREQUENT ITEMSET (MFI) ALGORITHM

MFI algorithm is used avoid candidate generation, the conventional FP growth algorithm needs to generate a large number of conditional pattern bases then calculate the conditional FP-tree. In case of large database it is inefficient and need a huge memory requirement that can occupies more memory space in main memory [13]. The MFI (mining frequent item set) algorithm is used to mine the frequent itemset. This algorithm uses the frequency of the improved FP-tree and stable count. The association rule mining have two steps- frequent item set generation and based on user defined support, find the valid association rules.

This algorithm has a simple approach to generate desired frequent item set that used to generate strong association rules. In this algorithm use the inputs parameters are user defined support S, frequency in Stable count C, frequency of the FP-tree F, root R. this algorithm take all possible combinations of items like existing FP-growth. This is take user defined support, based on support (database) and frequency of the FP-tree.

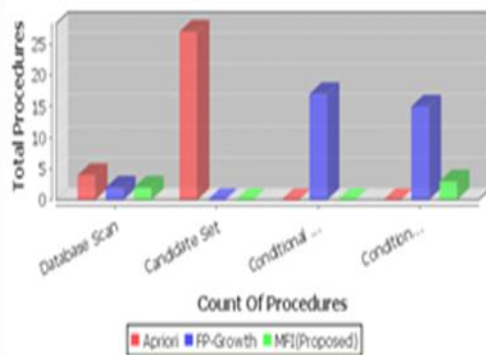
Steps of Mining Frequent Itemset:

1. Taking inputs parameters are FP-tree frequency, support of user defined, Stable count frequency, root node
2. Compares the each node in fp tree items to root node.
3. Suppose support of the item and frequency of the item in fp tree is same then it assigned to frequency of the frequent Itemset.
4. Generate item set with item and all possible combinations of the item and nodes with higher frequency in fp tree.
5. If suppose the support of the item is morethan the frequency of the item then adds the stable count to the frequency of frequent Itemset.
6. Generate Itemset with item and all possible combinations of item and all intermediate nodes up to most frequent item node in fp tree.
7. Suppose the support of the item is lesser than frequency of the item then it generate Itemset with item and all possible combinations of item and its parent node in fp tree.

If the frequency of the item in FP-tree is equal to the user defined support, then the frequency of frequent item sets for that item is FP-tree frequency. And the frequent items should be only those items, which have greater or equal FP-tree frequency at FP-tree. Because association rules are only concerns in those items that satisfy the minimum support. In the case where support is greater than the FP- tree frequency, the frequency for the frequent item sets is the total of the FP- tree frequency and the Stable count. The main reason for adding is, as the FP tree represent the most frequent items, then some element (spare items) will store in Stable though they have high frequency of occurrence in main transaction dataset. The frequent items should be all intermediate nodes of the path up to the most frequent item node, as all the nodes at same path represent relation between nodes and for strong

association rule, all possible relations need to be counted. In the last case where support is less than the FP- tree frequency, the frequency value for frequent item set is the frequency of the FP-tree. As the support is less than the frequency, no need to add the frequency of the Stable count. The frequent item sets are the only immediate parent nodes of the FP-tree. Because the entire parent node in improved FP-tree have greater frequency than the child node.

3.4 Experimental Results



4. CONCLUSIONS

In this paper, Database Scan, Candidate Set Generation, Conditional FP-Trees and Conditional Pattern Base consumption are considered for generating new scheme. The above factors also offered for finding the frequent item sets. The performances of the algorithms are strongly depends on the support levels and the features of the data sets. It is found that for a transactional database where many transaction items are repeated many times as a super set, In this case frequent Itemset algorithm is best suited for mining frequent itemsets. In the Apriori algorithm, number of database scans is more for generating Candidate sets. In the FP-Tree algorithm, items are in form of tree for finding frequent itemsets. Complete set of frequent items are generated in FP-Tree algorithm. The MFI (mining frequent Itemset) algorithm generates the association rules without candidate generation.

5. REFERENCES

- [1] A. Savasere, E. Omiecinski, and S. Navathe. "An efficient algorithm for mining association rules in large databases". In Proc. Int'l Conf. Very Large Data Bases (VLDB), Sept. 1995, pages 432-443.
- [2] Aggrawal.R, Imielinski.t, Swami.A. "Mining Association Rules between Sets of Items in Large Databases". In Proc. Int'l Conf. of the 1993 ACM SIGMOD Conference Washington DC, USA.
- [3] Agrawal.R and Srikant.R. "Fast algorithms for mining association rules". In Proc. Int'l Conf. Very Large Data Bases (VLDB), Sept. 1994, pages 487-499.
- [4] Brin.S, Motwani. R, Ullman. J.D, and S. Tsur. "Dynamic itemset counting and implication rules for market basket analysis". In Proc. ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD), May 1997, pages 255-264.
- [5] C. Borgelt. "An Implementation of the FP- growth Algorithm". Proc. Workshop Open Software for Data Mining, 1-5.ACMPress, New York, NY, USA 2005.
- [6] Han.J, Pei.J, and Yin. Y. "Mining frequent patterns without candidate generation". In Proc. ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD), 2000
- [7] Park. J. S, M.S. Chen, P.S. Yu. "An effective hash-based algorithm for mining association rules". In Proc. ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD), San Jose, CA, May 1995, pages 175-186.
- [8] Pei.J, Han.J, Lu.H, Nishio.S. Tang. S. and Yang. D. "H-mine: Hyper-structure mining of frequent patterns in large databases". In Proc. Int'l Conf. Data Mining (ICDM), November 2001.
- [9] C.Borgelt. "Efficient Implementations of Apriori and Eclat". In Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations, CEUR Workshop Proceedings 90, Aachen, Germany 2003.
- [10] Toivonen.H. "Sampling large databases for association rules". In Proc. Int'l Conf. Very Large Data Bases (VLDB), Sept. 1996, Bombay, India, pages 134-145.
- [11] Yiwu Xie, Yutong Li, Chunli Wang, Mingyu Lu. "The Optimization and Improvement of the Apriori Algorithm". In Proc. Int'l Workshop on Education Technology and Training & International Workshop on Geoscience and Remote Sensing 2008.
- [12] "Data mining Concepts and Techniques" by Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers, 2006.
- [13] S.P Latha, DR. N.Ramaraj. "Algorithm for Efficient Data Mining". In Proc. Int'l Conf. on IEEE International Computational Intelligence and Multimedia Applications, 2007, pp. 66-70.
- [14] Q.Lan, D.Zhang, B.Wu. "A New Algorithm For Frequent Itemsets Mining Based On Apriori And FP-Tree". In Proc. Int'l Conf. on Global Congress on Intelligent System, 2009, pp.360-364.
- [15] W.LIU, J.CHEN, S.Qu, W.Wan. "An Improved Apriori Algorithm. In Proc. IEEE International Conference, 2008, pp.221-224".
- [16] S.P Latha, DR. N.Ramaraj. "Algorithm for Efficient Data Mining". In Proc. Int'l Conf. IEEE International Computational Intelligence and Multimedia Applications, 2007, pp. 66-70.
- [17] M. El-Hajj and O. R. Zaiane. "Inverted matrix: Efficient discovery of frequent items in large datasets in the context of interactive mining". In Proc. Int'l Conf. on Data Mining and Knowledge Discovery (ACM SIGKDD), August 2003.
- [18] M. El-Hajj and O. R. Zaiane. "COFI-tree Mining:A New Approach to Pattern Growth with Reduced Candidacy Generation". Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, Melbourne, Florida, USA, CEUR Workshop Proceedings, vol. 90, pp. 112-119, 2003.
- [19] Y. G. Suchayo and R. P. Gopalan. "CT-ITL: Efficient Frequent Item Set Mining Using a Compressed Prefix Tree with Pattern Growth". Proceedings of the 14th Australasian Database Conference, Adelaide, Australia, 2003.