Volume 3, Issue 6, June 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Efficient Algorithm for Extracting Frequent Item Sets from a Data Set

Mohammad Mudassar Khan
M. tech Student
Shri Vaishnav Institute Of Technology & Science, Indore, India

Prof. Anand Rajavat
AssociateProfessor
Shri Vaishnav Institute Of Technology& Science, Indore, India

Abstract: Frequent item set mining is a heart favorite topic of research for many researchers over the years. It is the basis for association rule mining. Association rule mining is used in many applications like: market basket analysis, intrusion detection, privacy preserving, etc. In this paper, we have developed a method to discover large item sets from the transaction database. The proposed method is fast in comparison to older algorithms. Also it takes les main memory space for computation purpose.

Key words: Data Mining, Frequent Item sets, Apriori Algorithm, FP-Growth, proposed Algorithm.

I. Introduction

Data mining generally involves four classes of task; classification, clustering, regression, and association rule learning. Data mining refers to discover knowledge in huge amounts of data. It is a scientific discipline that is concerned with analyzing 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. Frequent patterns, such as frequent itemsets, substructures, sequences term-sets, phrasesets, 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 item set mining plays an important role in several data mining fields as association rules [1,2,4] warehousing [10], correlations, clustering of high-dimensional biological data, and classification [9]. Given a data set d that contains k items, the number of itemsets that could be generated is 2k - 1, excluding the empty set [1]. In order to searching the frequent itemsets, the support of each item set 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 item sets (Apriori techniques). The second strategy is to use a compressed data representation to facilitate in-core processing of the Item sets (FP-tree techniques). In addition another new algorithm has been developed [5] which uses top down graph based approach. In addition, many researches have been developed algorithms using tree structure, such as H-mine [3], FP-growth [6], and AFP-Tree [7]. 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 frequent item set mining is motivated by problems such as market basket analysis [3]. 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.

II. Background & Problem Definition

[2] 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:

Association rule is an implication of the form X -> Y where X,Y subset of I are the sets of items called Item sets and $X \cap Y = \Phi$. Association rules show attributes value conditions that occur frequently together in a given dataset. A commonly used example of association rule mining is Market Basket Analysis [2]. We use a small example from the supermarket domain. The set of items is-

I = {Milk, Bread, Butter, Beer}

A rule for the shopping market could be **{Butter, Bread} => {Milk}** meaning that if butter and bread are bought, customers also buy milk. For example, data are collected using bar-code scanners in supermarkets. Such shopping market

databases consist of a large number of transaction records. Each record lists all items bought by a customer on a single purchase transaction. Managers would be interested to know if certain groups of items are consistently purchased together. They could use this data for adjusting store layouts (placing items optimally with respect to each other), for cross-selling, for promotions, for catalog design and to identify customer segments based on buying patterns.

Association rules provide information 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. If 90% of transactions that purchase bread and butter, then also purchase milk.

Antecedent: bread and butter

Consequent: milk

Confidence factor: 90%

In addition to the antecedent (the "if" part) and the consequent (the "then" part), an association rule has two numbers that express the degree of uncertainty about the rule. In association analysis the antecedent and consequent are sets of items (called item sets) that are disjoint (do not have any items in common).

Support for an association rule X->Y is the percentage of transaction in database that contains X U Y .The other number is known as the **Confidence** of the rule. Confidence or Strength for an association rule X U Y is the ratio of number of transactions that contains X U Y to number of transaction that contains X is an item set (or a pattern) is frequent if its support is equal to or more than a user specified minimum support (a statement of generality of the discovered association rules). Association rule mining is to identify all rules meeting user-specified constraints such as minimum support and minimum confidence (a statement of predictive ability of the discovered rules). One key step of association mining is frequent item set (pattern) mining, which is to mine all itemsets satisfying user specified minimum support. [8]

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 above a predetermined minimum threshold known as frequent Itemsets.
2. Use the large itemsets to generate the association rules for the database that have confidence 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. Our main consideration of the thesis is First step i.e. to find the extraction of frequent itemsets.

The concept of frequent item set was first introduced for mining transaction databases (Agrawal et al. 1993). Let I = {I1, I2, In} be a set of all items. A k-item set α, which consists of k items from I, is frequent if α occurs in a transaction database D no lower than θ |D| times, where θ is a user-specified minimum support threshold (called min_sup), and |D| is the total number of transactions in D.

Proposed Algorithm:

Input:

- A Transaction Database D
- MST – Minimum support Threshold

Step1: Scan the transaction database to find the frequency of all size - 1itemsets
Step 2: Eliminate all those size-1 itemsets of step 1 whose support is less than the MST
Step 3: Eliminate the infrequent item from each transaction.
Step 4: Now arrange the itemsets in descending order of their itemcount (frequency)
Step 5: Call Recursivemining(new TDB)
Step 6: Stop

Recursivemining(new TDB)
Step 1: Create a matrix and put the transaction and the respective itemcount into the matrix.
Step 2: Find the size-k itemsets from the matrix whose support count is greater than the MST. If the support count is less than the MST then look for size-k itemsets and size k-1 itemsets together to find a new size k-1 item set and so on until no itemsets found greater then MST.
St**ep 3:** All maximum frequent itemsets are found in step 2, than according to downward closure property all the subsets are also frequent.
Step 4: There may be itemsets left over which are not included in maximal frequent item set but they are frequent. Consequently find all frequent 1-itemset and reduce the database just consider only those transactions which contain frequent 1-itemset element but not contain the maximal frequent transaction.
Step 5: If no such transaction found then return otherwise go to step 6.
Step 6: Call recursivemining(Reduced Transaction Database) Procedure.
Output : All frequent item set

III. Conclusion

In this paper, we presented a novel algorithm for mining frequent item sets. Frequent item set mining is crucial for association rule mining. We have evaluated the performance of our proposed algorithm. It is fast. Also it is taking less main memory for computation in comparison to previous algorithm.

References

[1] Ashok Savasere, E. Omiecinski and S. Navathe, "An efficient algorithm for mining association rules in large databases", Proceedings of the 21st International Conference on Very large database, 1995, pp. 420-431.

[2] Jia Ling, Koh and Vi-Lang Tu, " A Tree-based Approach for Efficiently Mining Approximate Frequent Itemsets", IEEE International Conference on Research Challenges in Information Science, 2010, pp. 25-36.

[3] Jian Pei ,J. Han, J. Lu, H. Nishio.S.and Tang, "H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases", ICDM International Conference on Data Mining, ICDM, 2001, pp. 441-448.

[4] Jiawei Han, Jian Pei, and Yiwen Yin, "Mining Frequent Patterns without Candidate Generation", Proceedings of ACM SIGMOD Conference, Dallas, TX, 2000, pp.53-87.

[5] Ramesh Agrawal and Ramakrishnan Srikant, "Fast algorithms for mining association rules", proceedings of the 20th VLDB Conference Santiago,Chille, 1994, pp. 487-499.

[6] Ramesh Agrawal, Tomasz Imielinski, and A. Swami, "Mining association rules between sets of items in large databases", ACM-SIGMOD Int. Conf. Management of Data, Washington, D.C., May 1993, pp 207–216.

[7] Yudho Giri Sucahyo and Gopalan.R, "Efficient Frequent Item Set Mining using a Compressed Prefix Tree with Pattern Growth", Proceedings of 14th Australian Database Conference, Adelaide, Australia, 2003, pp.95-104

[8] R. Cheng, D. Kalashnikov, and S. Prabhakar, "Evaluating Probabilistic Queries over Imprecise Data," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2003.

[9] W. Cheung and O.R. Zaı¨ane, "Incremental Mining of Frequent Patterns without Candidate Generation or Support Constraint," Proc. Seventh Int'l Database Eng. and Applications Symp. (IDEAS), 2003.

[10] C.K.-S. Leung, Q.I. Khan, and T. Hoque, "Cantree: A Tree Structure for Efficient Incremental Mining of Frequent Patterns," Proc. IEEE Fifth Int'l Conf. Data Mining (ICDM), 2005

