

Association Rule Mining: An Optimization Approach for the Generation of Frequent Itemsets

Authors Name:A.SOUMYA
Computer Science & Engineering,
Christhu Jyoti Institute of Technology &Science
Jangaon, India
Soumya1251@gmail.com

Authors Name:P.AVANIKETH
Computer Science & Engineering,
Christhu Jyoti Institute of Technology &Science
Jangaon, India
avaniketh@gmail.com

Abstract- Mining frequent itemsets is one of the most investigated fields in data mining. It is a fundamental and crucial task. Association rule mining is the most important technique in the field of data mining. Association rule mining delivers frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories. Frequent pattern mining is one of the active areas in data mining. It plays an important role in all data mining tasks such as clustering, classification, prediction, and association analysis. Identifying all frequent patterns is the most time consuming process due to a massive number of patterns generated. In this paper, we present the study of finding the frequent items in the fastest way. We also present an approach for mining of association rule.

Keywords: Association rules, Apriori algorithm, Data mining, frequent itemsets.

I. INTRODUCTION

Data mining is a powerful new technology with great potential to help companies focus on the most important information in the data they have collected about the behavior of their customers and potential customers. Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems.

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information. Data mining software is one of a number of analytical tools for

analyzing data. Data mining sometimes called data or knowledge discovery. Data mining, also known as knowledge discovery in databases, has been recognized as a new area for database research. The area can be defined as efficiently discovering interesting rules from large collections of data. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Correlation typically is association rules. Association rules are one of the major techniques of data mining. 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 help in much business can decision making processes, such as cross marketing, Basket data analysis, and promotion assortment. It helps to find the association relationship among the large number of database items and its most typical application is to find the new useful rules in the sales transaction database, which reflects the customer purchasing behavior patterns, such as the impact on the other goods after buying a certain kind of goods. These rules can be used in many fields, such as customer shopping analysis, additional sales, goods shelves design, storage planning and classifying the users according to the buying patterns, etc. 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.

In data mining, Apriori [1] is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. In association rule mining, given a set of itemsets, the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length $k - 1$.

The major reason behind data mining's great deal of attraction and attention in information industry in recent years, is due to the wide availability of huge amounts of data, and

the eminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration [7]. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates. The quest to mine frequent patterns appears in many domains. The prototypical application is market basket analysis, i.e., to mine the sets of items that are frequent bought together, at a supermarket by analyzing the customer shopping carts (the so-called “market baskets”). Once we mine the frequent sets, they allow us to extract association rules among the item sets, where we make some statement about how likely are two sets of items to co-occur or to conditionally occur.

II. ASSOCIATION RULE

An example of an association rule would be "If a customer buys a bread, he is 80% likely to also purchase milk." Given a minimum confidence threshold *minconf* and a minimum support threshold *minsup*, the problem is to generate all association rules [4] that have support and confidence greater than the user-specified minimum support and minimum confidence. In the first pass, the support of each individual item is counted, and the large ones are determined. In each subsequent pass, the large itemsets determined in the previous pass is used to generate new itemsets called candidate itemsets. The support of each candidate itemset is counted, and the large ones are determined. This process continues until no new large itemsets are found. In Association Rule mining find rules that will predict the occurrence of an item based on the occurrence of the other items in the transaction. Table shows Market-Basket Transactions

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
2	Milk , Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Table1. Transaction items

Example of Association Rules:

{Diaper} @ {Beer},
 {Bread, Milk} @ {Egg, Coke},
 {Bread, Beer} @ {Milk},

Implication means co-occurrence, not causality. Association rule [6] is an implication expression of the form $X @ Y$, where X and Y are itemsets.

Example: {Milk, Diaper} @ {Beer}

A) Rule Evaluation

- **Support (S):** Fraction of transactions that contain both X and Y .
- **Confidence (C):** Measures how often items in Y appear in transactions that contain X .

Example:

{Milk, Diaper} → {Beer}

$S = \sigma(\{\text{Milk, Diaper, Beer}\}) / [T]$

$S = 2/5$ $S = 0.4$

$C = \sigma(\{\text{Milk, Diaper, Beer}\}) / \sigma(\{\text{Milk, Diaper}\})$

$S = 2/3$ $S = 0.67$

- **Itemset:** A collection of one or more items. Example {Milk, Diaper, Beer}. K -itemset that contains k -items.
- **Frequent Itemset:** An itemset whose support is greater than or equal to a min_sup threshold. In association rule mining task from a set of transactions T , the goal of association rule mining is to find all rules having $\text{Support} \geq \text{min_sup}$ threshold and $\text{Confidence} \geq \text{min_conf}$ threshold.

There are two phases in the problem of data mining association rules.

1. Find all frequent itemsets: i.e. all itemsets that have support s above a predetermined minimum threshold.
2. Generate strong association rules from the frequent itemsets: these association rules must have confidence c above a predetermined minimum threshold.

After the large item sets are identified, the corresponding association rules can be derived in a relatively straightforward manner. Thus the overall Performance of mining association rules is determined primarily by the first step. Efficient counting of large itemsets is thus the focus of most association rules mining algorithms [3].

III. APRIORI ALGORITHM: MINING FREQUENT ITEMSETS

Apriori algorithm is the most classical and important algorithm for mining frequent itemsets, proposed by R. Agrawal and R. Srikant in 1994 [2]. 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". 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. 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 . 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^2 , the set of frequent 2-itemsets, which is used to find L^3 , and so on, until no more frequent k -itemsets can be found. The basic steps to mine the frequent elements are as follows:

- **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.
- **Join step:** To attain the next level elements C_k join the previous frequent elements by self-join i.e. $L_{k-1} * L_{k-1}$ 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.
- **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 are included in C_k thus prunes the C_k to find L_k 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 a large 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 Ck. Step2 and 3 is repeated until no new candidate set is generated. It is no doubt that Apriori algorithm successfully finds the frequent elements from the database. But as the dimensionality of the database [5] 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.

IV. CONCLUSION

Mining significant association rules between items in a large database of transactions, an association rules are basic data mining tools for initial data exploration usually applied to large data sets, seeking to identify the most common groups of items occurring together. There are various association rule mining algorithms. In this paper we have studied and presented association rule mining algorithm, Apriori.

V. REFERENCES

- [1] WEI Yong-qing, YANG Ren-hua, LIU Pei-yu, "An Improved Apriori Algorithm for Association Rules of Mining" IEEE(2009)
- [2] Rakesh Agrawal and Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," Proc 20th International Conference Very Large Data Bases (VLDB), pp. 487-499, Year 1994.
- [3] Rakesh Agrawal and Ramakrishnan Srikant, "Privacy-Preserving Data Mining," Proc. ACM SIGMOD
- [4] Rakesh Agrawal and Ramakrishnan Srikant, "Mining Generalized Association Rules," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 407-419, 1995
- [5] Secure Mining of Association Rules in Horizontally Distributed Databases, Tamir Tassa, IEEE Transactions On Knowledge And Data Engineering, VOL. 26, NO. 4, April 2014.
- [6] Gagandeep Kaur, Shruti Aggarwal, Performance Analysis of Association Rule Mining Algorithms, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, August 2013, ISSN: 2277 128.
- [7] H. Grosskreutz, B. Lemmen, and S. Reuping, "Secure Distributed Subgroup Discovery in Horizontally Partitioned Data," Trans. Data Privacy, vol. 4, no. 3, pp. 147-165, 2011.