Frequent Item Set Mining – A Review

Devendra Verma[1] Mr. Gajendra Singh[2]
[1,2]SSSSIT, Sehore

Abstract-- In this paper, we present a literature survey of existing frequent item set mining algorithms. The concept of frequent item set mining is also discussed in brief. The working procedure of some modern frequent item set mining techniques is given. Also the merits and demerits of each method are described. It is found that the frequent item set mining is still a burning research topic.

I. INTRODUCTION

With the increase in Information & Communication Technology, the size of the databases created by the organizations for information storage is also increasing. Some of such organizations include retail, telecommunications, petroleum, utilities, manufacturing, credit cards, transportation, insurance, banking etc. There are many more such organizations, involved in extracting the valuable data, it necessary to explore the databases completely and efficiently. Knowledge discovery in databases (KDD) helps to identifying precious information in such huge databases. This valuable information can help the decision maker to make accurate future decisions. KDD applications deliver measurable benefits, including profit maximization, increase in sales, reduced cost of doing business, enhanced profitability, and improved quality of service. So it is clear that the Knowledge Discovery in Databases has become one of the most active and exciting research areas in the database community.

In recent years, the size of database used for storing valuable information in an organization has increased exponentially. This has led to a increasing researchers interest in the development of tools capable in the automatic extraction of knowledge from data. Data mining or knowledge discovery in database is a field of research dealing with the automatic discovery of implicit information or knowledge within the databases. The implicit information within databases, mainly the interesting association relationships among sets of objects that lead to association rules may disclose useful patterns for rain forecast, decision support, , disease prediction, financial forecast, , attribute prediction, marketing policies, even medical diagnosis and many other applications.

II. LITERATURE SURVEY

A literature survey of some commonly used methods for frequent pattern mining as follows:

A. Apriori Algorithm

The first and foremost algorithm for mining all frequent itemsets and strong association rules was the AIS algorithm by [3]. After that, the algorithm was improved and renamed Apriori. Apriori algorithm is, the most popular, classical and important algorithm for mining frequent itemsets. Apriori algorithm is used to find all frequent itemsets in a given database DB. The basic idea of Apriori algorithm is to make multiple passes over the database. It mines pattern by using 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 Apriori algorithm is based on the concept of downward closure property, which states that all nonempty subsets of a frequent itemsets must be frequent [2]. It also uses the anti-monotonic property of a system which says if the system cannot pass the minimum support test then all its supersets will fail to pass the test [2, 3]. Therefore by above properties if the one set is infrequent then all its supersets are also frequent and vice versa. In this way, this property is used to prune the infrequent candidate elements in an efficient manner. In the first pass, the set of frequent 1-itemsets is found. The set of one item, which satisfy the support threshold, is denoted by L.

In each subsequent pass the algorithm begin with an initial set of itemsets found to be large in the previous pass. This initial set is used for generating new potentially large itemsets. These item sets are called candidate itemsets. In each pass, the support for these candidate itemsets is calculated. At the end of the pass, all the infrequent itemsets are eliminated and the frequent sets becomes the initial set for the next pass. This feature first invented by [2] in Apriori algorithm is used by the many algorithms for frequent pattern generation. The basic steps to find the frequent patterns are as follows [3]:

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

1 Extra search space is needed and also I/O cost will increase
2 Time increases as the number of database scan is increased thus candidate generation will increase results in increase in computational cost.

B. Direct Hashing and Pruning (DHP):

It is observed that reducing the candidate items from the database is one of the important task for increasing the efficiency. If we can reduce the number of candidate sets then it will result in lower time and space complexity. To do the same a DHP technique was proposed [5] to reduce the number of candidates in the early passes. In this method, the support of an item is counted by mapping the items from the candidate list into the buckets which is divided according to support known as Hash table structure. When a new itemset is encountered if item exist earlier then increase the bucket count else insert into new bucket. At the end the bucket

whose support count is less the minimum support is removed from the candidate set.
In this way the DHP algorithm reduces the generation of candidate sets in the earlier stages but as the level increase the size of bucket also increase thus difficult to manage hash table as well candidate set. It is a complex algorithm.

C. Partitioning Algorithm:

Partitioning algorithm [1] is based on the concept partitioning. To find the frequent elements on the basis partitioning of database in n parts, it uses memory efficiently. It overcomes the memory problem for large database which do not fit into main memory because small parts of database easily fit into main memory. This algorithm is divided into two main passes,
1 In the first pass of the algorithm the whole database is divided into n number of parts.
2 In second pass each partitioned database is loaded into main memory one by one and local frequent elements are found.
3 Then conquer all the locally frequent elements and make it globally candidate set.
4 Finally find the globally frequent elements from this candidate set.
If the minimum support for transactions in whole database is min_sup then the minimum support for partitioned transactions is min-sup number of transaction in that partition. It is also observed that local frequent itemset may or may not be frequent with respect to the entire database thus any itemset which is potentially frequent must include in any one of the frequent partition.

D. Sampling Algorithm:

The sampling algorithm [6] is used to overcome the limitation of I/O overhead by not considering the whole database for checking the frequency. This algorithm is based in the idea to pick a random sample of itemset R from the database instead of whole database D. The small sample is picked in such a way that whole sample is accommodated in the main memory. In this algorithm we try to find the frequent elements for the sample only and there is chance to miss the global frequent elements in that sample therefore lower threshold support is used instead of actual minimum support to find the frequent elements local to sample. In the best case , the sampling algorithm requires only one pass to find all frequent elements if all the elements included in sample and if elements missed in sample then second pass are needed to find the itemsets missed in first pass or in sample [7].
Thus sampling approach is beneficial if efficiency is more important than the accuracy because this approach gives the result in very less scan or time and overcome the limitation of memory consumption arises due to generation of large amount of datasets but results are not as much accurate.

E. Dynamic Itemset Counting (DIC):

The DIC algorithm [4] was also used to reduce the number of database scan. It is also based upon the downward disclosure property in which adds the candidate itemsets at different point of time during the scan. In this algorithm the dynamic blocks are formed from the database marked by start points and unlike the previous techniques of Apriori it dynamically changes the sets of candidates during the database scan. It cannot start the next level scan at the end of first level scan then start the scan by starting label attached to each dynamic partition of candidate sets.
In this way DIC reduces the database scan for finding the frequent itemsets by just adding the new candidate at any point of time during the run time. But DIC generates the large number of candidates and computing their frequencies are the bottleneck of performance while the database scans only take a small part of runtime.

F. Improved Apriori algorithm

It was observed in [8] [7] that the improved Apriori algorithm is based on the combination of forward scan and reverse scan of a given database. If certain conditions are satisfied then the improved algorithm can greatly reduce the iteration, scanning times required for the discovery of candidate itemsets.
Suppose the itemset is frequent then all of its nonempty subsets are frequent. It is contradictory to the given condition that one nonempty subset is not frequent then the itemset is not frequent.

III. CONCLUSION

In this paper, we presented a literature survey of the popular frequent item set mining algorithms. The working of each method is also discussed in brief. The merits and demerits of each method are also described in summarized way.

REFERENCES

[1] 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] 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 “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. 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] Park. J. S, M.S. “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.
[6] Toivonen “Sampling large databases for association rules”. In Proc. Int’l Conf. Very Large Data Bases (VLDB), Sept. 1996, Bombay, India, pages 134–145.
[7] By Jiawei Han, Micheline Kamber, “Data mining Concepts and Techniques” by Morgan Kaufmann Publishers, 2006.
[8] Shaohua Teng, Wei Zhang, Haibin Zhu. “An Algorithm to Improve the Effectiveness of Apriori”. In Proc. Int’l Conf. on 6th IEEE Int. Conf. on Cognitive Informatics (ICCI'07), 2007.