An Improved Apriori Algorithm for Market Basket Analysis

Swati Gupta
Assistant Professor, Department Of CSE
Amity University Haryana

Abstract*—Data mining is an essential process to discover an interesting pattern from the given data. It is an influential algorithm to find out frequent item set. We propose an improved Apriori algorithm which find out the frequent item set in an efficient and less time .In order to prove the correctness of proposed approach an example has being explained which shows that an efficient apriori can be achieved by pruning the item set from the initial database whose support count is less than minimum support count.

***Keywords*—Frequent Itemsets, Support_Count.**

I. INTRODUCTION

Data mining is one of the essential process of discovering some hidden and interesting patterns from the massive amount of data where data can be stored in data warehouse, OLAP (on line analytical process), databases and other repositories of information [11]. This data may reach to more than terabytes[1,3,4]

The name of Data mining is (KDD) knowledge discovery in databases [3], as it includes series of steps to find the useful data. The KDD involves a series of steps which are performed to extract patterns to user, such as data cleaning, data selection, data transformation, data preprocessing, data mining and pattern evaluation [4].

The architecture of data mining system as shown in Fig 1 has the following main components [6]: data warehouse, database or other repositories of information, a server that fetches the relevant data from repositories based on the user's request, knowledge base is used as guide of search according to defined constraint, data mining engine include set of essential modules, such as characterization, classification, clustering, association, regression and analysis of evolution. Pattern evaluation module that interacts with the modules of data mining to strive towards interested patterns[7,8,9] Finally, graphical user interfaces from through it the user can communicate with the data mining system and allow the user to interact[10,11]

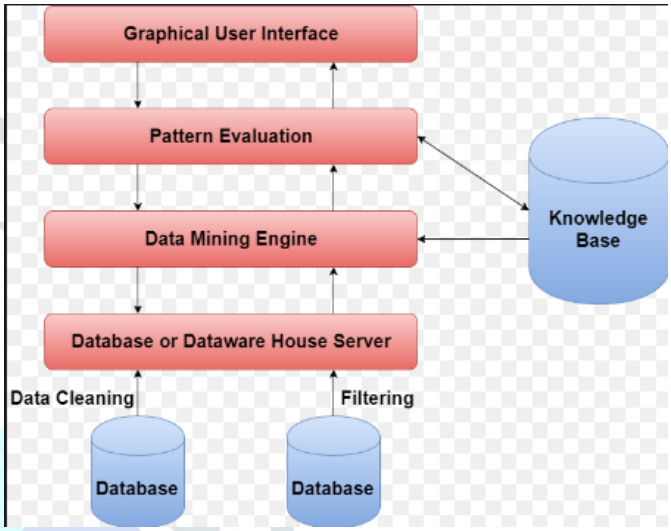

Figure 1: Architectural Components of Data Mining

There are numerous data mining techniques one of the most prominent technique is Association Mining[12]. Association rules mining are used to discover the associations and relations among item sets of large data. It is an important branch of data mining research, and association rules is the most typical style of data mining. Presently, association rules mining problems are highly valued by the researchers in database, artificial intelligence, statistic, information retrieval, visible, information science, and many other fields. Many incredible results have been found out [14,15]. What can efficiently catch the important relationships among data is simple form of association rules and easily to explanation and understanding. Mining association rules problems from large database has become the most mature, important, and active research contents[16].

II. APRIORI ALGORITHM

Apriori algorithm is the originality algorithm of Boolean association rules of mining frequent item sets, raised by R. Agrawal and R. Srikan in 1994. The core principles of this theory are the subsets of frequent item sets are frequent item sets and the supersets of infrequent item sets are infrequent item sets. This theory is regarded as the most typical data mining theory all the time [3].The figure 2 represents the Association rule data classification.

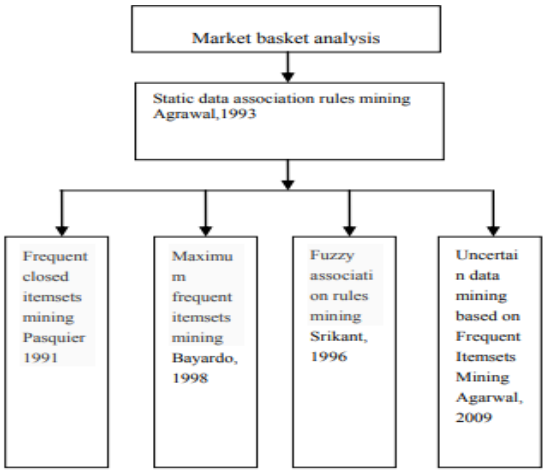

Fig 2: Association Rule Classification

The algorithm is used to find out all the frequent item sets and to find out association between different items to carry out market basket analysis. Basically an Apriori is an influential algorithm of association mining which works with several iterations in order to produce results. In the first iteration, item set P directly constitutes the first candidate item set C1. Let us assume that P= {p1, p2,…, pm}, then C1={{a1},{a2},…,{am}}. In the last Kth iteration, firstly, the candidate item set Ck of this iteration emerges according to the frequent item set Lk-1 found in the last iteration. (The candidate item set is the potential frequent item set and is the superset of the K-1th frequent item set. Item set with k candidate item sets is expressed as Ck, which was consisted by k frequent item sets Lk .)

Then we take a counter and we distribute the counter which has an initial value equals to zero to every item set and scan the entire database D in proper order. We ensure that every affairs belongs to each item sets and the counter of these item sets will increase[4,5]
When all the item sets have been scan, the support level can be induced according to the actual value of |D| and the minimum support level of the certain Ck of the frequent item set. We will repeat this process until no mew item occurs. [4]
The Apriori algorithm comprises of two key processes:

- Joining Step
- Pruning Step

The functionality of each step is as follows:
Joining Step: In this step in order to get Lk , connect Lk-1 with itself. Set this candidate as Ck and assume L1 and L2 are the item sets of Lk-1. Li[j] is the jth item of Li . Assume the affairs and items of the item set are in the dictionary order. Execute the Lk-1, in which the elements of Lk-1, L1 and connection Lk-1<> L1, are connectable, if they have the same first (k-2)th items.
That is, the elements of Lk-1, L1 and L1, are connectable, if (L1∧……(L1[k-2]=L2[k-2])∧(L1[2]=L2[2])∧(L1[1]=L2[1]) [k-1]=L2[k-1]).
Pruning step: In Pruning Step the Ck is the superset of Lk , that is that the members of it could be frequent or not, but all the k frequent item sets are all include in Ck . Scan the database, clear the counters of every candidate item sets of Ck to assure Lk . However, Ck might be very large, and then the amount of calculation will be huge [1,6,8]. In order to decrease Ck. Consequently, if the (k-1) subset of a candidate item set with k items are not in Lk-1, the candidate item set is not frequent and can be deleted in Ck .

Apriori algorithm employs the bottom up strategy which follows the following property that all subset of a frequent item sets should also be frequent. However, once this kind of algorithms meet dense database (such as, telecom, population census, etc.), as large amounts of long forms occur, the properties sharply drop. The Apriori algorithm suffers from some weakness in spite of being clear and simple[6,9,10]. The main limitation is costly wasting of time to hold a vast number of candidate sets with much frequent item sets, low minimum support or large item sets.

For example, let us assume if there are 104 from frequent 1-item sets, it need to generate more than 107 candidates into 2-length which in turn they will be tested and accumulate [2]. Furthermore, to detect frequent pattern in size 100 (e.g.) v1, v2… v100, it have to generate 2100 candidate item sets [1] that yield on costly and wasting of time of candidate generation. So, it will check for many sets from candidate item sets, also it will scan database many times repeatedly for finding candidate item sets. Apriori will be very low and inefficiency when memory capacity is limited with large number of transactions.

III. An Improved Apriori Algorithm

A. Realization of the Algorithm

According to the property of Apriori to discover o frequent item sets, our algorithm decreases the number of scans to our database D by eliminating the items which lies below the minimum support count. It declines the number of candidate item sets further by eliminating those item sets whose support count is less than the given support count.

The figure 3 depicts the series of steps that are being followed. The improved algorithm has being designed for the following purpose.

- To improve the Efficiency
- To reduce the scan this is being made to transactional database D.

In other words, prune Lk-1 before Ck occur using Lk-1. This algorithm can also be described as following: Count the number of the times of items occurs in Lk-1 (this process can be done while scan data D); Delete item sets with this number less than k-1 in Lk-1 to get Lk-1. To distinguish, this process is called Prune 1 in this study, which is the prune before candidate item sets occur; the process in Apriori algorithm is called Prune 2, which is the prune after candidate item sets occur. Thus, to find out the k candidate item sets, the following algorithm can be taken

which is being discussed in next section.

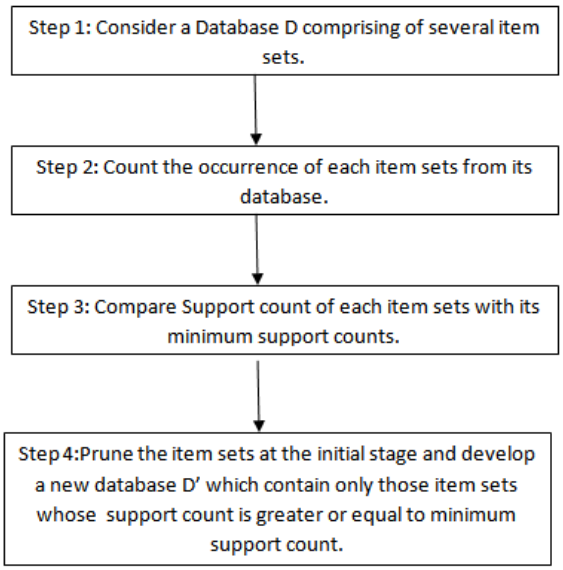

Fig 3: Series of Steps for Improved Apriori Algorithm

B. The Improved Apriori Algorithm

The improved Apriori algorithm is as follows:

//Generate items, items support, their transaction ID
(1) Scan the database D and prune out the itemsets whose sup_count<min_Sup_count to create a new database D'.
(2) For a new database D' we find L1 = find_frequent_1_itemsets (T);
(3) For (k = 2; Lk-1 ≠Φ; k++) {
//Generate the Ck from the LK-1
(4) Ck = candidates generated from Lk-1; //get the item Iw with minimum support in Ck using L1, (1≤w≤k).
(5) Q= Get _item_min_sup(Ck, L1); // get the target transaction IDs that contain item x.
(5) Tgt = get_Transaction_ID(Q);
(6) For each transaction t in Tgt Do
(7) Increment the count of all items in Ck that are found in Tgt;
(8) Lk= items in Ck ≥ min_support;
(9) End;
(10) }

C. An Illustrative Example of Improved Apriori Algorithm

In order to prove the correctness of our proposed Apriori algorithm we analyze the following scenario. Let us consider an example to demonstrate the working of our improved Apriori algorithm. There is a transactional database D as shown in table 1 which comprises of 10 Transactions and the minimum support count=3.

The series of steps are being carried out to find the frequent item set in highly efficient manner.

Table 1: Initial database D

T_ID	List of Items
T1	A1,A2,A5
T2	A2,A4
T3	A2,A5
T4	A1,A2,A4
T5	A1,A3
T6	A2,A3
T7	A1,A3
T8	A1,A2,A3
T9	A1,A2,A3
T10	A1,A2

Firstly we will scan the database D and count the support count of each individual data item which is shown below.

Table 2:Support_Count of Item set

Item sets	Support_Count
A1	7
A2	8
A3	5
A4	2
A5	2

Since there are several item set such as A4 and A5 in the table 2 whose minimum support_count is less than 3, so we will prune those item set before finding out the frequent item sets. Thereby we get a new database D' as shown in table 3.

Table 3: A new database D'

T_ID	List of Items
T1	A1,A2
T2	A2
T3	A2
T4	A1,A2
T5	A1,A3
T6	A2,A3
T7	A1,A3
T8	A1,A2,A3
T9	A1,A2,A3
T10	A1,A2

We will start with finding out the frequent item sets

Step 1: Finding Frequent-2 –item sets

Table 4: Frequent-2-itemsets

List of Items	Support_Count
A1,A2	5
A1,A3	4
A2,A3	3

We compare the support_count with minimum_support count and finds that the item sets in table 4 are frequent and so we will not prune any item sets and will proceed to next step.

Step 2: Finding frequent-3-itemsets

We follow the property of Apriori to find frequent item set which says that all subset of a frequent item set must also be frequent.

Now, we will consider the pair

A1, A2, A3 and we find the sub set as

A1, A2
A1, A3
A2, A3

We notice that all the subset are part of frequent-2-itemset.Hence we prove that Frequent-3-itemset is A1, A2, A3.
in much efficient manner.

CONCLUSION

Apriori is an influential algorithm for finding out the frequent item set which is being used for finding out the association between the different item sets.
In this paper, we have discussed an improved Apriori Algorithm which reduces the time consumed in transactions scanning for candidate item sets by reducing the number of transactions to be Scanned. The series of steps are being carried out to illustrate the working of our proposed approach by creating a new transactional database which prunes the item set whose support count is less than minimum support_count. The proposed algorithm is superior than the Apriori algorithm.

REFERENCES

[1] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu,P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, Dec. 2007.
[2] S. Rao, R. Gupta, "Implementing Improved Algorithm Over APRIORI Data Mining Association Rule Algorithm", *International Journal of Computer Science And Technology*, pp. 489-493, Mar. 2012
[3] H. H. O. Nasereddin, "Stream data mining," *International Journal of Web Applications*, vol. 1, no. 4, pp. 183–190, 2009.
[4] F. Crespo and R. Weber, "A methodology for dynamic data mining based on fuzzy clustering," *Fuzzy Sets and Systems*, vol. 150, no. 2, pp. 267–284, Mar. 2005.
[5] R. Srikant, "Fast algorithms for mining association rules and sequential patterns," UNIVERSITY OF WISCONSIN, 1996.
[6] J. Han, M. Kamber,"Data Mining: Concepts and Techniques*", Morgan Kaufmann Publishers*, Book, 2000.
[7] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996.
[8] F. H. AL-Zawaidah, Y. H. Jbara, and A. L. Marwan, "An Improved Algorithm for Mining Association Rules in Large Databases," Vol. 1, No. 7, 311-316, 2011
[9] T. C. Corporation, "Introduction to Data Miningand Knowledge Discovery*", Two Crows Corporation*, Book, 1999.
[10] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, vol. 22, pp. 207–216, 1993
[11] M. Halkidi, "Quality assessment and uncertainty handling in data mining process," in Proc, EDBT Conference, Konstanz, Germany, 2000.
[12] J. Han and M. Kamber, *Conception and Technology of Data Mining*, Beijing: China Machine Press, 2007.
[13] J. N. Wong, translated, *Tutorials of Data Mining*. Beijing. Tsinghua University Press, 2003.
[14] Y. Yuan, C. Yang, Y. Huang, and D. Mining, *And the Optimization Technology and Its Application*. Beijing. Science Press, 2007.
[15] Y. S. Kon and N. Rounteren, "Rare association rule mining and knowledge discovery:technologies for frequent and critical event detection. H ERSHEY," PA: *Information Science Reference*, 2010
[16] W. Sun, M. Pan, and Y. Qiang, "Inproved association rule mining method based on the statistical," *Application Research of Computers*. vol.28, no. 6, pp. 2073-2076, Jun, 2011.