IJITKM Volume 7 • Number 2 Jan– June 2014 pp. 185-187 (ISSN 0973-4414)

A Survey on Optimization of APRIORI Algorithm for high Performance

Sunita Sihag[1], Dr. Gundeep Tanwar[2]
[1]*M.Tech, BRCM College of Engineering , Bahal*
[2]*Associate Professor , BRCM College of Engineering , Bahal*

Abstract:-
There are several mining algorithms of association rules. One of the most popular algorithms is Apriori that is used to extract frequent itemsets from large database . Based on this algorithm, this paper indicates the limitation of the original Apriori algorithm of wasting time for scanning the whole database searching on the frequent itemsets, and presents an improvement on Apriori by reducing that wasted time depending on scanning only active transactions. The paper shows by experimental results with several groups of transactions, and with several values of minimum support that applied on the original Apriori and our implemented optimized Apriori that our optimized Apriori reduces the time consumed.
Keywords: Apriori, Improved Apriori, Frequent itemset, Support, Candidate itemset.

Data Mining
A process used by companies to turn raw data into useful information. By using software to look for patterns in large batches of data, businesses can learn more about their customers and develop more effective marketing strategies as well as increase sales and decrease costs. Data mining depends on effective data collection and warehousing as well as computer processing.
Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. 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.

APRIORI
Finding frequent itemsets is one of the most investigated fields of data mining. Association rule and frequent itemset mining became a widely researched area, and hence faster and faster algorithms have been presented. Numerous of them are APRIORI based algorithms or APRIORI modifications. Those who adapted APRIORI as a basic search strategy, tended to adapt the whole set of procedures and data structures as well. Apriori algorithm is easy to execute and very simple, is used to mine all frequent itemsets in database. The algorithm makes many searches in database to find frequent itemsets where k-itemsets are used to generate k+1-itemsets. Each k-itemset must be greater than or equal to minimum support threshold to be frequency. Otherwise, it is called candidate itemsets. In the first, the algorithm scan database to find frequency of 1-itemsets that contains only one item by counting each item in database. The frequency of 1-itemsets is used to find the itemsets in 2-itemsets which in turn is used to find 3-itemsets and so on until there are not any more k-itemsets. If an itemset is not frequent, any large subset from it is also non-frequent; this condition prune from search space in database.
This algorithm follows three steps :
1. For l from 1 to I do
2. For each set JI such that for each h JI occurs in at least k baskets do
3. Examine the data to determine whether the set JI oc-curs in at least k baskets
For this algorithm, most of its time has been spent in ac-cessing the database until it results into one frequent associa-tion match. Based on this work, the probability model was calculated with two quantities:
1. Success rate : the probability that the set is a success
2. Failure rate: the probability that the set is a failure

Optimized Apriori Algorithm:-
We will modify existing apriori algorithm by which it will take less time and also works with less memory. Following algorithm is used in paper we have taken. Our objective is to optimize this algorithm to improve performance.
In below Algorithm following work has been done to reduce the time:
They enhance the Apriori algorithm to reduce the time consuming for candidates itemset generation. We firstly scan all transactions to generate L1 which contains the items, their support count and Transaction ID where the items are found. And then we use L1 later as a helper to generate L2, L3 ... Lk. To generate C2, they make a self join L1 * L1 to construct 2-itemset C (x, y), where x and y are the items of C2. Before scanning all transaction records to count the support count of each candidate, use L1 to get the transaction IDs of the minimum support count between x and y, and thus scan for C2 only in these specific transactions. The same thing for C3, construct 3-itemset C

(x, y, z), where x, y and z are the items of C3 and use L1 to get the transaction IDs of the minimum support count between x, y and z, then scan for C3 only in these specific transactions and repeat these steps until no new frequent itemsets are identified.

//Generate items, items support, their transaction ID
(1) L1 = find_frequent_1_itemsets (T);
(2) For (k = 2; Lk-1 ≠Φ; k++) {
//Generate the Ck from the LK-1
(3) Ck = candidates generated from Lk-1;
//get the item Iw with minimum support in Ck using L1, (1≤w≤k).
(4) x = Get _item_min_sup(Ck, L1);
// get the target transaction IDs that contain item x.
(5) Tgt = get_Transaction_ID(x);
(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) }

In above algorithm, complete table has been scanned as Original Apriori but Candidate generation process han been changed as shown in step 3 and 4.

Our approach:-

We will focus on following steps of above algorithm to improve its performance

We will Optimize the Apriori algorithm to reduce the time consuming for candidates itemset generation. We will generate L1 without scanning of complete Table. Our Algorithm will filter Data Table on the basis of desired column. Construct Ck using L1, L2…Lk , L1,L2.. Lk tool less time for generation, so Ck will be generated fast.

Write Algorithm and program efficiently which will take less computation time.

We developed an implementation for original Apriori and our Optimized Apriori,

We will compare original Apriori with our Optimized Apriori and analyse time reduced by new Algorithm for different transactions.

We will implement this code in Weka Tool. In this we are working only with the clustering algorithms because it is most important process, if we have a very large database. I am using WEKA tools for clustering. The main thing, why I am chooses WEKA, because we can work in WEKA easily without having the deep knowledge of data mining techniques.

Related work:-

Arpna Shrivastava[1] introduced Fast implementation of Apriori algorithm analyzed and modified it to find frequent item sets and association rules of level-3. The modification is done in three steps. In first step, the transaction database is coded using a new coding scheme and in second step the cleaning of database is done if required and at the third step code of implementation modified and a new module is added to facilitate the third level association rules generation. This algorithm is based on fast implementation of Apriori algorithm and generating the third level of association rules.

Hassan M. Najadat[3], this paper indicates the limitation of the original Apriori algorithm of wasting time for scanning the whole database searching on the frequent itemsets, The paper shows by experimental results with several groups of transactions, and with several values of minimum support that applied on the original Apriori and our implemented improved Apriori that our improved Apriori reduces the time consumed by 67.38% in comparison with the original Apriori, and makes the Apriori algorithm more efficient and less time consuming.

Sheila A. Abaya[4] There are several ways to improve the database access of Apriori algorithm thereby improving also the efficiency of the execution. Based on the modified code, set size and set size frequency were introduced. These factors helped in a more rapid generation of possible association of frequent items. In terms of database passes, the modified apriori provides less database access compared with the original one that makes its execution faster.

Ms Shweta[5] In this paper, author considers data (bank data) and tries to obtain the result using Weka a data mining tool. Here author consider three association rule algorithms: Apriori Association Rule, PredictiveApriori Association Rule and Tertius Association Rule. Author compares the result of these three algorithms and presents the result. According to the result obtained using data mining tool author find that Apriori Association algorithm performs better than the PredictiveApriori Association Rule and Tertius Association Rule algorithms.

Rachna Somkunwar[6] This paper discusses an enhanced version of Apriori algorithm that is concentrated on four characteristics namely, First data preparation and chooses the desired data, second produce itemsets that decides the rule constraints for knowledge, third mine k-frequent itemsets using the new database and fourth produce the association rule that sets up the knowledge base and offer better results. Another approach discussed in this paper are the HASH MAPPING TABLE and HASH_TREE tactics used to optimize space complexity and time complexity.

Conclusion:

In this paper, an Optimized Apriori is proposed through reducing the time consumed in transactions scanning for candidate itemsets by reducing the number of transactions to be scanned. Whenever the k of k-itemset increases, the gap between our Optimized Apriori and the original Apriori increases from view of time consumed, and whenever the value of minimum support increases, the gap

between our Optimized Apriori and the original Apriori decreases from view of time consumed.

References

[1]. Arpna Shrivastava1, R. C. Jain and A. K. Shrivastava,(2014), Generating 3rd Level Association Rules Using Fast Apriori Implementation, British Journal of Mathematics & Computer Science.

[2] Ferenc Bodon Informatics Laboratory,2010, A fast APRIORI implementation, Computer and Automation Research Institute, Hungarian Academy of Sciences.

[3].Hassan M. Najadat1, Mohammed Al-Maolegi2, Bassam Arkok3[2013] , An Improved Apriori Algorithm for Association Rules, International Research Journal of Computer Science and Application

[4]Sheila A. Abaya[2012] , Association Rule Mining based on Apriori Algorithm in Minimizing Candidate Generation ,International Journal of Scientific & Engineering Research

[5] Ms Shweta1 Dr. Kanwal Garg2 ,1M Tech. Scholar, 2Assistant Professor,

Dept. of Computer Science and Applications,Kurukshetra, India Kurukshetra, India[2013]

Mining Efficient Association Rules Through Apriori Algorithm Using Attributes and Comparative Analysis of Various Association Rule Algorithm, International Journal of Advanced Research in Computer Science and Software Engineering.

[6] Rachna Somkunwar Computer Department, Nagpur University India [2012]

A study on Various Data Mining Approaches of Association Rules, International Journal of Advanced Research in Computer Science and Software Engineering.

[7]Sandhu, P.S. ; Dept. Of CSE, Rayat & Bahra Inst. of Eng. & Bio-Tech., Mohali, India ; Dhaliwal, D.S. ; Panda, S.N. ; Bisht, A. An Improvement in Apriori Algorithm Using Profit and Quantity,Ieee2010.

[8]R. Agrawal and R. Srikant.(1994) Fast algorithms for mining association rules. The International Conference on Very Large Databases.

[9]R. Agrawal and R. Srikant. Mining sequential patterns(1995). In P. S. Yu and A. L. P. Chen, editors, Proc. 11th Int. Conf. Data Engineering, ICDE.

[10]N. F.Ayan, A. U. Tansel, and M. E. Arkun(1999). An efficient algorithm to update large itemsets with early pruning. In Knowledge Discovery and Data Mining.