MODIFIED VERSIONS OF APRIORI ALGORITHM: A SURVEY

Rini Christopher[#1], M.Tech in CSE, Marian Engineering College, Trivandrum,India

Sini.S.Raj[2], Asst.Professor in CSE, Marian Engineering College, Trivandrum, India

[#]rinichristopher@gmail.com

Abstract: The need for storing high volume of data in database and processing that information has led to the increased demand for efficient data mining techniques. Association rule mining is one of the important steps in data mining. Apriori algorithm is the first and best known algorithm for association rule mining. Even though Apriori is effectively used in various applications in the area of data mining, it possesses many limitations. This paper present a survey on the various modifications done in the classical Apriori algorith by several researchers and concludes the paper by proposing a priority matrix based technique for mining infrequent itemsets of high priority along with the frequent itemsets for more accurate and stronger rule generation as its future work.

Keywords: *data mining; association rule mining; apriori algorithm; support; confidence; itemsets*

I.INTRODUCTION

In recent years, as the computer technology advances large amounts of data have been collected routinely in the course of day to day management in business, administration, banking, the delivery of social and health services, environmental protection, security and in politics. Such data is primarily used for accounting and for management of the customer base. Typically, management data sets are very large and constantly growing and contain a large number of complex features. One require robust, simple and computationally efficient tools to extract information from such data sets. The development and understanding of such tools is the core business of data mining. [1] . Hence it is an important area of study, to know the extent of association between such attributes. This is why association rule mining is crucial. There are various algorithms which fall under this. Major association rule mining algorithms include Apriori algorithm, Tertius algorithm, Frequent pattern growth algorithm and Eclant algorithm. All these algorithms provide ways to create rules on associated attributes. Apriori algorithm is the first and best-known for association rules mining. [4] This paper discusses the classical rule mining algorithm: Apriori. This algorithm suggests solutions to various applications such as market basket analysis, qualitative content analysis, fault prediction systems, wave prediction systems etc. This paper provides a survey report on the various modified Apriori algorithms and the techniques implemented in them.

II.CONCEPTS

Two major concepts used while working with the Apriori algorithm is Support and Confidence. Let's define what exactly these terms are:

Association rule: It is defined in the form of $x \rightarrow y$, indicating that if a transaction contains item set *x* then it will likely contain item set *y* as well.[2] Association rule mining is such a process which provides numerous ways to find association between variables.

Support: Support(s) of an association rule x=>y is the percentage of transaction in the database that contain x∪y. [1]

Confidence: or strength (α) for an association rule x=>y is the ratio of the number of transaction that contain x∪y to the number of transactions that contain x.

Apriori property: The superset of all frequent itemset will be frequent [2]. This is the major property used while calculating the frequent data.

Above mentioned properties make Apriori unique and classical from other association rule learning algorithms.

III.APRIORI ALGORITHM

Apriori algorithm is, the most classical and important algorithm for mining frequent itemsets, proposed by R.Agrawal and R.Srikant in 1994. 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!, the set of frequent 2-itemsets, which is used to find L , 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 Ck join the previous frequent elements by self join i.e. Lk-1* Lk-1 known as Cartesian product of Lk-1. I.e. This step generates new candidate k-itemsets based on joining Lk-1 with itself which is found in the previous iteration. Let Ck denote candidate k-itemset and Lk be the frequent k-itemset.

Prune step: Ck is the superset of Lk so members of Ck may or may not be frequent but all K ' 1 frequent itemsets are included in Ck thus prunes the Ck to find 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 Ck would result in the determination of Lk (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to Lk). Ck, however, can be huge, and so this could involve grave computation. To shrink the size of Ck, 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 Lk-1 then the candidate cannot be frequent either and so can be removed from Ck. Step 2 and 3 is repeated until no new candidate set is generated.

Algorithm 1 Function Apriori

Join Step: Ck is generated by joining Lk-1 with itself
Prune Step: Any (k-1) - item set that is not frequent cannot be a
subset of a frequent k - item set.
Ck: Candidate item set of size k
Lk: frequent item set of size k
L1 = frequent items
for (k = 1; k < L; k++) do begin
Ck+1 = candidates generated from Lk
for each transaction t in database **do**
increment the count of all candidates in
Ck+1 that are contained in t
Lk+1= candidates in Ck+1 with min support
end
return Ek U Lk

IV.MODIFIED APRIORI ALGORITHMS

Organised Transaction Selection Approach:

In this method the bottom-up approach is replaced by organized transaction selection approach. This algorithm uses organized transaction selection approach[3], where in the rules are generated by picking up the transactions according to the highest order first basis and hence avoiding generation of un-necessary patterns that are not a part of the original database. The major advantage of this approach is that, the number of database scans is massively reduced, to the order of O(n) i.e., of the order of number of transactions available for frequent patterns generation. Hence overcomes the relatively higher time complexity of original Apriori algorithm which is of the order $O(e^n)$.

Applying Correlation Threshold:

In addition to the usual concepts a new term Correlation threshold is introduced, which is implemented in the proposed Apriori algorithm.[4] Correlation threshold is a factor which transfers the probability from single itemset to n-itemset. Correlation threshold finds its application in candidate item set generation. The modified Apriori algorithm incorporates correlation threshold for finding strong association rules between the itemsets. The correlation threshold is a value between 0 and 1. If the value is 1, then the attributes are highly related to each other. While a value close to zero shows the dataset as independent. This correlation confirms the presence of all itemset appearing in traditional Apriori in proposed algorithm. Performance of the redesigned algorithm is evaluated and is compared with the traditional Apriori algorithm. The evaluation shows a peak improvement in the mining result. The time complexity of the newly designed algorithm is reduced into O (n).

Confabulation Inspired Algorithm:

The next approach proposes a CARM [Confabulation-inspired Association Rule Mining] [2] approach using cogency inspired measure for generating rules. Cogency inspiration can lead to more intuitive rules. Moreover, cogency-related computations only need pair wise item co-occurrences; hence finding rules can be done by only one file scan. In large data sets, the number of file input/output operations (data complexity) affects their performance, so data complexity is one way of measuring performance of algorithms. Apriori-based algorithms need $\min(k + 1, m)$ scans of input data set, where k is the size of the largest frequent itemset. CARM can mine association rules by only one scan of the data set. Although it is a nested loop, it uses data stored in main memory. Since file access, particularly for large files can be significantly time consuming, it can be concluded that the proposed algorithm should be much faster than the Apriori due to one-time file access.

Map/Reduce Approach:

The approach presents a Apriori-Map/Reduce Algorithm [5] that implements and executes Apriori algorithm on

Map/Reduce framework. The proposed Apriori-Map/Reduce Algorithm runs on parallel Map/Reduce framework such as Apache Hadoop. The algorithm starts with calculating frequent item set for each map. Then collect the frequent item set and remove items that does not meet the minimum support in reduce nodes. Then calculates frequent item set with an additional item by joining, sorting, and eliminating the duplicated items in each map and again collect the frequent item set at the reduce nodes. Now it counts the item frequencies that do not meet the minimum support at the map nodes and removes them. The time complexity is p times less than the sequential Apriori algorithm; where p is the number of map and reduce nodes assuming the node sizes are the same.

Intersection and Record Filter Approach:

In Record filter approach, count the support of candidate set only in the transaction record whose length is greater than or equal to the length of candidate set, because candidate set of length k , cannot exist in the transaction record of length k-1 , it may exist only in the transaction of length greater than or equal to k. In Intersection approach, to calculate the support, count the common transaction that contains in each element's of candidate set. This approach requires very less time as compared to classical Apriori. In Proposed Algorithm [6] , set theory concept of intersection is used with the record filter approach. In proposed algorithm, to calculate the support, count the common transaction that contains in each element's of candidate set. In this approach, constraints are applied that will consider only those transaction that contain at least k items. The main disadvantage of this approach is that memory optimization is done but still it needs much optimization.

Improvement based on Set Size Frequency:

The improved algorithm for Apriori [7] takes for the set size which is the number of items per transaction and set size frequency which is the number of transactions that have at least set size items. Initially a database is given with set size and second database is provided with set size frequency of the initial database. Remove items with frequency less than the minimum support value initially and determine initial set size to get the highest set size whose frequency is greater than or equal to minimum support of set size. Set sizes which are not greater than or equal to min set size support are eliminated. The main disadvantage of this approach is that ideal starting size of combination size for pruning candidate keys is not given.

Optimization through Genetic Algorithm:

This paper [8] explains that Strong rule generation is an important area of data mining. In this paper authors design a novel method for generation of strong rule. In which a general Apriori algorithm is used to generate the rules after that authors use the optimization techniques. Genetic algorithm is one of the best ways to optimize the rules. In this direction for the optimization of the rule set they design a new fitness function that uses the concept of supervised lear**n**ing then the GA will be able to generate the stronger rule set.

Based On Cost And Quantity:

This paper [9] proposes an efficient approach based on weight factor and utility for effectual mining of significant association rules. Initially, the proposed approach makes use of the traditional Apriori algorithm to generate a set of association rules from a database. The proposed approach exploits the anti-monotone property of the Apriori algorithm, which states that for a k-itemset to be frequent all (k-1) subsets of this itemset also have to be frequent. Subsequently, the set of association rules mined are subjected to weightage (W-gain) and utility (U-gain) constraints, and for every association rule mined, a combined Utility Weighted Score (UW-Score) is computed. Ultimately, it determines a subset of valuable association rules based on the UW-Score computed. The experimental results demonstrated the effectiveness of the proposed approach in generating high utility association rules that can be lucratively applied for business development.

HDO Apriori Algorithm:

In this paper [10], the High-Dimension Oriented Apriori algorithm, the HDO Apriori, is proposed for mining the association rules in the high-dimensional data. Based on the classical Apriori, the new algorithm can cut down the redundant generation of identical sub-itemsets from candidate itemsets, by means of pruning the candidate itemsets with the infrequent itemsets with lower dimension. So it can obtain a higher efficiency than that of the original algorithm when the dimension of data is high. Meanwhile, for different data, it still need further research to find methods to estimate how much improvement the HDO Apriori algorithm can implement.

Support Matrix based Algorithm:

The proposed algorithm [11] replaces arbitrary user defined minimum support with functional model based on Standard Deviation. In proposed algorithm Minimum support value is calculated based upon Standard Deviation value of support counts of all transactions. This approach make this algorithm more comfortable for somebody non expert in data mining. Presented algorithm uses Bottom up Approach to find the frequent item set from largest frequent Item set to smallest frequent item set which help in mining long pattern easily. This algorithm works in 2 phases, Support Matrix Generation and Bottom Up approach to mine frequent items set based upon calculated minimum support. The major advantages of this method are reduction in number of scans and time required to mine the frequent item set.

Fast Completion Apriori:

This algorithm [12] starts out like the regular Apriori Algorithm, proceeding in a level-wise manner, generating and testing candidates as it goes, but as soon as it determines

that the number of candidates for all remaining levels is not too large, it generates candidates for the remaining levels based on the currently available information. So, in fact, it runs the Apriori Algorithm but stops at a certain level n. From that level on, the algorithm uses the frequent itemsets of size n to generate candidates for all remaining higher levels. Like the name indicates the algorithm performs much faster than Apriori.

Apriori-Growth Algorithm

A new algorithm based on Apriori and the FP-tree structure is presented, which is called Apriori-Growth [13]. This method only scans the data set twice and builds FP-tree once while it still needs to generate candidate itemsets. The Apriori-Growth mainly includes two steps. First, the data set is scanned one time to find out the frequent 1 itemsets.Then the data set is scanned again to build an FP-tree. At last, the built FP-tree is mined by Apriori- Growth instead of FP-Growth.There are many advantages of this method. First, Apriori-Growth works much faster than Apriori. It uses a different method FPtreeCalcualte to calculate the support of candidate itemsets. Second, Apriori-Growth works almost as fast as FP-Growth. But it consumes less memory than FPGrowth because it doesn't need to generate conditional pattern bases and build sub-conditional pattern tree recursively.

V.CONCLUSION

Association rule mining is an interesting topic of research in the field of data mining. We have presented a survey of most recent research work on the first and best known association rule mining algorithm-Apriori. Furthermore we are proposing a priority matrix based Apriori algorithm that finds the infrequent itemsets of high priority in a transaction along with the frequent itemsets as its future work. In the existing algorithm the infrequent itemsets of high priority are removed during pruning step which reduces the accuracy of the algorithm. The new technique incorporates a ranking method so that the high priority itemsets are not removed even if it occurs infrequently. The proposed algorithm works in two steps: the priority matrix generation and pruning based on support value calculated during each iteration to generate stronger rules. Here a dynamic support value is used rather than a static support value used by the classical Apriori algorithm. However association rule mining is still in a stage of exploration and development. There are still some essential issues that need to be studied for identifying useful association rules.

ACKNOWLEDGEMENT

The authors take this opportunity to express their gratitude to the Institute of Technology for providing necessary documents which helped in the successful completion of this paper.

REFERENCES

[1] The Apriori Algorithm - a Tutorial; Markus Hegland *CMA, Australian National University*-March 30, 2005.
[2] Confabulation-Inspired Association Rule Mining for Rare and Frequent Itemsets Azadeh Soltani and M.R. Akbarzadeh.T., Senior Member, IEEE-*IEEE Transactions On Neural Networks And Learning Systems-2014.*
[3] Advanced Version of Apriori Algorithm- K.R.Suneetha, R.Krishnamoorti - Bharathidasan Institute of Technology-*First International Conference on Integrated Intelligent Computing*-2010.
[4] Applying Correlation Threshold on Apriori Algorithm- Anand H.S, Dept. of Computer Science, College of Engineering, Trivandrum, Vinodchandra S.S.,Computer Center, University of Kerala-*IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology*-2013.
[5] Apriori-Map/ReduceAlgorithm-Jongwook Woo-Computer Information Systems Department, California State University, Los Angeles, *CA- 2013.*
[6] Survey on several improved Apriori algorithms-Ms. Rina Raval, Prof. Indr Jeet Rajput , Prof. Vinitkumar Gupta-Department of Computer Engineering ,H.G.C.E.,-*IOSR Journal of Computer Engineering (IOSR-JCE) e*-ISSN: 2278-0661,Vol.9, Issue 4 - 2013.
[7] Sheila A. Abaya, "Association Rule Mining based on Apriori Algorithm in Minimizing Candidate Generation",-*International Journal of Scientific & Engineering Research*-Volume 3, Issue 7, July-2012.
[8] Rupali Haldulakar, Prof. Jitendra Agrawal, Optimization of Association Rule Mining through Genetic Algorithm, *International Journal on Computer Science and Engineering (IJCSE),* Vol. 3, Issue.3, Mar 2011.
[9] An Improvement in Apriori algorithm Using Profit And Quantity-Parvinder S. Sandhu, Professor (Deptt. Of CSE), Dalvinder S. Dhaliwal, Asst Prof. (Deptt. Of CSE), S. N. Panda ,Director & Professor, Regional Institute of Mgmt. & Tech., *Second International Conference on Computer and Network Technology*- 2010 IEEE.
[10] New Improvement on Apriori Algorithm-Lei Ji, Baowen Zhang, Jianhua Li *Information Security Engineering School, Shanghai Jiaotong University*, Shanghai-C2006 IEEE
[11] Educational Data Mining using Improved Apriori Algorithm Jayshree Jha and Leena Ragha-Department of Computer Engineering, Ramarao Adik Institute of Technology,Navi Mumbai, India-*International Journal of Information and Computation Technology*-ISSN 0974-2239 Volume 3, Number 5 (2013), pp. 411-418.
[12] A Probability Analysis for Candidate Based Frequent Itemset Algorithms- Nele Dexters University of Antwerp Middelheimlaan, Paul W. Purdom Indiana University Computer Science Bloomington, Dirk Van Gucht Indiana University, Computer Science-2006.
[13] An Efficient Frequent Patterns Mining Algorithm based on Apriori Algorithm and the FP-tree Structure-Bo Wu, Defu Zhang, Qihua Lan, Jiemin Zheng *Department of Computer Science, Xiamen University, Xiamen 361005, China-Third 2008 International Conference on Convergence and Hybrid Information Technology*-2008 IEEE