ISSN: 2321-7782 (Online)
Volume 3, Issue 6, June 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Priority Model of Apriori Algorithm

Rini Christopher[1]
Mtech in CSE
Marian Engineering College
Trivandrum - India

Merlin Shoerio[2]
Asst. Professor in CSE
Marian Engineering College
Trivandrum - India

Abstract: ***The need for storing high volume of data in database and processing those 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 proposes a Priority model of Apriori algorithm which increases the accuracy of the rules generated by mining infrequent itemsets of high priority along with the frequent itemsets. There are situations where combinations of itemsets which are not in abundance can make some remarkable effects in output generated in certain applications. Using Apriori algorithm it fails to detect these high priority rare items, as in the first step itself it gets pruned off due to its low frequency. This is where the proposed algorithm comes into play. First part of the paper describes the original Apriori algorithm and second half describes the implementation of the Priority model of Apriori algorithm. Last section describes a comparative study between the algorithms and in an application level, qualitative content analysis of water is done to affirm the results.***

Keywords: data mining; association rule mining; Apriori algorithm; support; confidence; frequent itemsets; item probability.

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 requires 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. 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 Apriori algorithm along with the implementation of a modified Apriori algorithm. This algorithm suggests solutions to market basket analysis, qualitative content analysis, predictive systems, medical applications etc which generate rules that shows the relationship between items in large databases. In the proposed algorithm we discuss methods for finding associations with improved accuracy by incorporating a probability matrix based on the priority of the items in the transaction.

II. CONCEPTS

The 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 $x \rightarrow y$ is the percentage of transaction in the database that contain $x \cup y$. [1]

Confidence: or strength (c) for an association rule $x \rightarrow y$ is the ratio of the number of transaction that contain $x \cup y$ 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. In other words the property can be described as; all nonempty subsets of a frequent itemsets must be frequent.

Antimonotone property: The property states that if the system cannot pass the minimum support test, then all its supersets will also fail to pass the test.

Above mentioned properties make Apriori unique and classical from other association rule learning algorithms. The concepts used in the priority matrix based apriori algorithm are as follows:

Item flag value (I_f): It is the value initially set for the high priority rare items inorder for the system to recognise them as important items in the transaction. Usually the value set ranges from 0.1 to 0.5, where highest priority item is set as 0.5 and lowest as 0.1. The higher the flag value, the more impact the item makes in the rules generated.

Item Probability (I_p): Where I_p is the probability of each item in the transaction, which is calculated by dividing the count of each item with the highest item count in the transaction.

$$I_p = (I_c / \text{highest } I_c) \qquad (1)$$

Where I_c is the item count in the transaction.

Matrix Probability (M_p) : This matrix probability is required in-order to confirm the propagation of probability of each itemsets. The equations used to calculate M_p are:

- Initial iteration

  For frequent items,

  $$M_p = I_p - \beta \qquad (2)$$

  For high priority items,

  $$M_p = I_p + I_f \quad 0 < Ip < 0.5 \qquad (3)$$

  $$M_p = I_p + \beta \quad 0.5 < Ip < 0.9 \qquad (4)$$

  $$M_p = \alpha \quad 0.9 \leq Ip \leq 1 \qquad (5)$$

Where α and β are two constants whose value depends on the item probability and the value of $\alpha+\beta = 1$ at all times.

- Second Iteration onwards

  For high priority items,

  $$M_p = I_p + \max(I_f) \quad 0 < Ip < 0.5 \qquad (6)$$

Since from frequent 2-itemset onwards contains more than one item in the set and if more than one item is a high priority item, then the flag value is chosen to be the highest of those items.

Support Priority (S_p) : It is the dynamic support threshold set in each iteration for the generation of next frequent itemset from the candidate itemset. In each iteration the support is calculated as,

$$S_p = (\Sigma M_p / |T|) \quad (7)$$

Where |T| is the number of items in the candidate itemset in the current iteration and ΣM_p is the sum of matrix probability of items in the current iteration. The S_p value calculated is rounded off to one decimal point.

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 and anti monotonic property. Therefore if the one set is frequent 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 that contains one item, which satisfy the support threshold, is denoted by L1. 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, L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found.[4] The basic steps to mine the frequent elements are as follows:

Generate and test: In this, first find the frequent 1-itemset elements L1, by scanning the database and removing all those elements from candidate itemset C1, which cannot satisfy the minimum support criteria.

Join step: To attain the next level candidate elements Ck; where $k \geq 2$, join the previous frequent items 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 frequent k-1 itemsets are included in Ck, thus prune the Ck to find frequent k-itemsets with the help of Apriori and anti monotonic property. I.e. This step eliminates some of the candidate k-itemsets using the Apriori property, which scans 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. Anti monotonic property is used to check the support count of each candidate item with the support threshold set earlier. If the candidate support is less than the minimum support threshold, then the item is pruned from Ck and doesn't belong in Lk. 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 Lk U Lk

A. Limitations of Apriori Algorithm

The major limitations of Apriori algorithm are: repeated number of database scanning makes the algorithm time consuming; large number of candidate itemset generation leads to high memory usage and since a fixed support value is used, early pruning of items having low frequency leads to reduced accuracy of the rules generated.

Inorder to overcome these limitations many modified versions of Apriori algorithm has been proposed. The modified versions which provides solutions to first two limitations are: Organised transaction selection approach [3], Corelation threshold based algorithm[4], Confabulation inspired algorithm[2], Map/reduce approach[5], Intersection and record filter approach [6], Algorithm based on cost and quantity[9], High definition oriented Apriori algorithm[10], Support matrix based algorithm [11] , Apriori growth algorithm [13] etc. To generate stronger rules, optimization through genetic algorithm [8] was proposed. The main goal of the proposed algorithm mentioned in this paper is to generate more accurate rules by considering the high impact infrequent items in the transaction.

IV. PRIORITY MODEL

Along with the usual concepts of original Apriori algorithm, we introduce a new concept of priority matrix in our proposed algorithm. A priority matrix is created to calculate the probability of item occurrence of frequent and high priority non frequent items in each transaction. This matrix probability is required in-order to confirm the propagation of probability to each itemsets. The standard support and confidence measures of original Apriori works well in many cases, although not so ideal when dealing with rare items which has an important role in the rules generated. These rare items must be considered since it can be important in many real- world applications, such as fraud detection, fault prediction, rare disease diagnosis, qualitative content analysis and network intrusion. Hence selection of a static support threshold for pruning items can lead to generation of undesirable rules. This is because setting a high support threshold may remove an important rare item early, there by missing the generation of some interesting rules and likewise setting a low support threshold can lead to the generation of many uninteresting rules. Therefore a dynamic support threshold is set for each iteration using the matrix value calculated inorder to generate stronger, accurate and desired rules in the proposed algorithm. Priority matrix generation finds its application in candidate item set generation. It helps in finding strong association rules between the itemsets thereby increasing its accuracy.

Like in classical Apriori, priority model also employs level wise search. Assume the values of β; the minimum probability, as 0.1 and $\alpha = 1-\beta$; the maximum probability of item in the transaction. Algorithm begins by scanning the database D initially to find the count of each item in the transaction. Using this count, the item probability is calculated using equation (1) and then the matrix probability for both common and high priority items are calculated. The matrix probability is a value between 0 and 1. If the value is nearer to 1, then the attributes are highly related to each other. While a value close to zero shows the itemset as independent. This probability value confirms the presence of itemsets appearing in traditional Apriori in this algorithm too and also includes those items which are rare but has a significance in the rule generated. Then the support value is calculated using equation (7) for the iteration and is checked upon each calculated matrix probability and prune step is done to find the frequent 1-itemset. For generating next candidate itemset, join step is implemented and the process continues until no further frequent itemset can be generated. The steps included in priority model of Apriori algorithm are:

Matrix probability (M_p) calculation: The M_p of frequent items are calculated using equation (2) and that of high priority rare items are calculated using (3) to (5) for generating frequent 1-itemset. For generating rest frequent itemset, equation (6) is used instead of equation (3). Rest are all the same.

Join Step: Ck is generated by joining Lk-1 with itself. This step is similar to the join step of original Apriori.

Prune Step: The M_p values less than S_p is removed from the candidate itemset to generate frequent itemset.

The Priority model of Apriori algorithm is given in Algorithm 2.

Algorithm 2 Function Priority Model - Apriori:

Input: Database D with n elements

Output: Frequent item, I and Rule S

Assumptions: If , α, β

Data structure Used: Probabilistic Array, PA[n]

while (n!=NULL)

Scan the database D

and compute the item count (I_c)

Calculate the item probability (I_p) of each item

input I_p of the n elements into PA[n]

for(i=0; i<n; i++)

calculate the matrix probability (M_p) from PA[n]

calculate the support priority (S_p) of the transaction

for (k=0; k< n; k++)

if ($M_p(k) < S_p$)

prune the item

else input I in Lk

Lk = frequent itemset generated of size L

for (i =1; i < L; i++) do begin

Join Lk with itself to create Ck+1

Ck+1 = candidates generated from Lk

Continue finding Lk until no further join possible

ie; until Lk = ∅

end

return U Lk

V. IMPLEMENTATION EXAMPLE

Inorder to show the stepwise implementation of the modified Apriori algorithm, a real life application to determine the quality of water by generating rules which contains mineral and other contents found in water was used. Drinking water contains various minerals and dissolved solids. By using original Apriori algorithm the most commonly occurring elements are generated as rules. But sometimes presence of rare elements such as mercury, lead, arsenic etc in water can make the water unusable. Using Apriori algorithm it fails to detect these high priority rare items, as in the first step itself it gets pruned off due to its low frequency. This is where the proposed algorithm comes into play. The final rules generated using priority model of Apriori contains these rare items along with the common elements. Table I denotes a sample real time database used for the content prediction and Table II is the 5-transaction database used to affirm the comparison results.

TABLE I DATABASE-WATER CONTENTS

Transaction	CaCO3	Cl2	NO3-	SO4	HgCl2	Pb(NO3)2
1	1	1	1	1	0	0
2	1	0	1	1	0	1
3	1	0	1	1	0	0
4	0	1	1	1	0	1
5	1	1	1	1	1	0

TABLE II 5-TRANSACTION DATABASE

Transaction	item1	item2	item3	item4	item5	item6
1	1	2	3	4	0	0
2	1	0	3	4	0	6
3	1	0	3	4	0	0
4	0	2	3	4	0	6
5	1	2	3	4	5	0

Stepwise process of the algorithm is described using this database (Table I). The items under consideration is calcium carbonate (CaCO3), chlorine (Cl2), nitrate (NO3-), sulphate (SO4), mercuric chloride (HgCl2) and lead II nitrate (Pb(NO3)2).Here mercuric chloride and lead nitrate are the rare elements present and since both are toxic chemicals, they have higher priority than the others elements. Using Apriori, these rare items are removed early because of their low frequency and thus making the rules generated unuseful for the current situation.

A. Observation of 5-transaction database

The working of the Priority model of Apriori is explained using Table II, which is a sample transaction set generated similar to that of the water contents in Table I. Initially, the flag value (I_f) of high priority items and the values of α, β are assumed. Here item5 and item6 are the high priority rare items whose flag values are set as 0.5 and 0.3 respectively. Then each transaction is scanned one after another and the count of each item (I_c) is taken to calculate the item probability (I_p) using equation (1). Using this I_p, the matrix probability (M_p) of high priority and common items are calculated using equations (2) to (6). Then the Support Priority (S_p) is calculated using equation (7).Then the Prune step is implemented. The resulting is the rule generated from the initial iteration. Inorder to generate next candidate itemset, Join set is implemented and the process continues until no further candidate itemset can be generated. The example is explained using a diagrammatic approach.

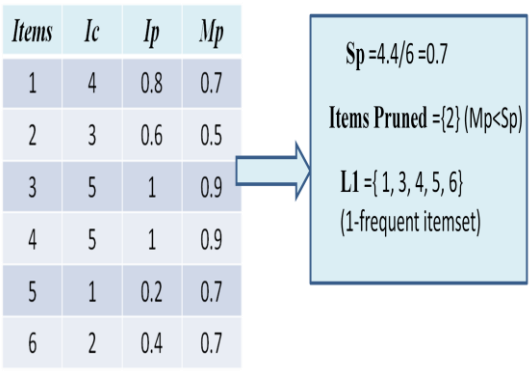

Fig 1 (a). Initial iteration

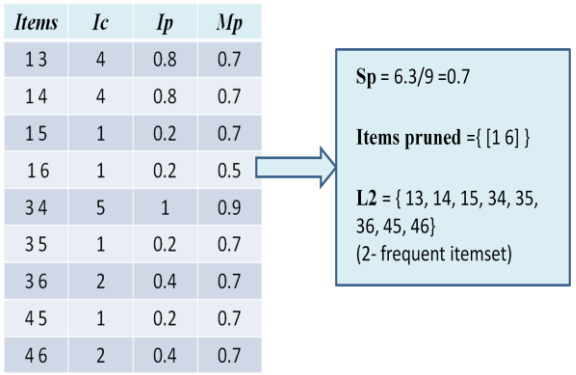

Fig 1(b). Second iteration

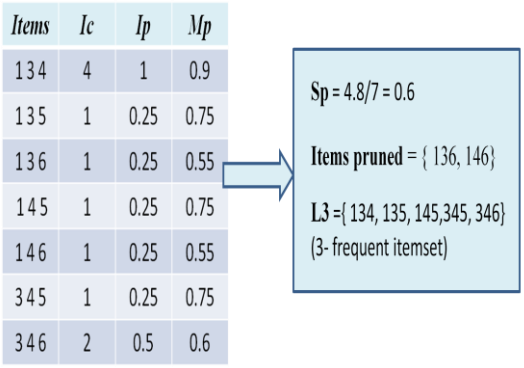

Fig 1 (c). Third iteration

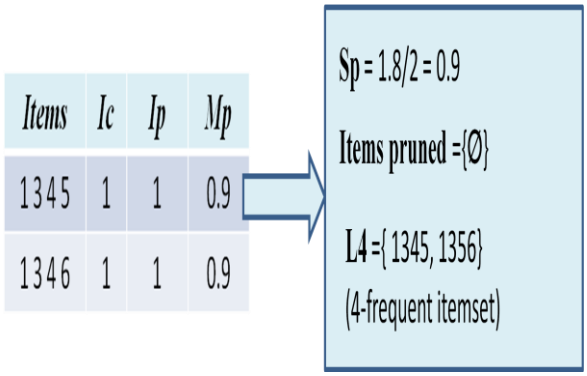

Fig 1(d). Fourth iteration

The algorithm stops at fourth iteration since further candidate itemset cannot be created. From the rules generated, it is evident that the accuracy has been increased and thereby increasing the performance of the algorithm. Table III draws clear distinction on the number of itemsets generated by original Apriori algorithm to the proposed priority model of Apriori algorithm. The Apriori algorithm was implemented in the same database with a support threshold of 0.5.Table IV shows a comparison between the two algorithms.

TABLE III FREQUENT ITEMS GENERATED

Itemset	***Apriori***	***Priority model***
L1	{ 1,2, 3, 4}	{ 1,3,4,5,6}
L2	{13,14, 23, 24, 34}	{13, 14, 15, 34, 35, 36, 45, 46}
L3	{134, 234}	{134, 135, 145, 345, 346}
L4	-	{1345, 1346}

TABLE IV COMPARISON OF ALGORITHMS

Apriori Algorithm	***Priority model***
Total candidate items generated = 17	Total candidate items generated= 24
Total frequent itemset generated = 11	Total frequent itemset generated = 20
Number of iteration= 3	Number of iteration = 4
The rules generated doesn't include the high priority rare items.	The rules generated contain the high priority rare items.
Rules declare the water as usable	Rules declare the water unusable

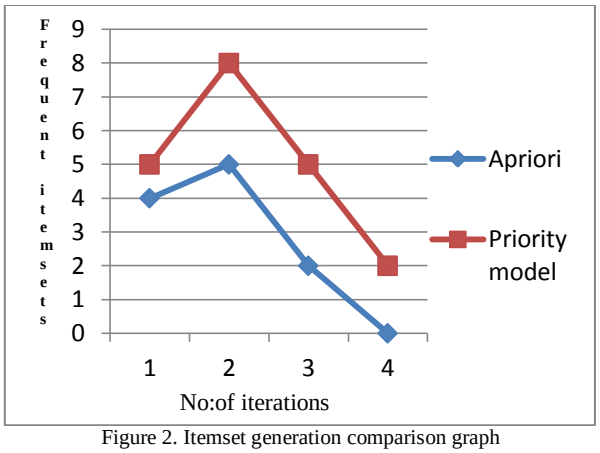

Figure 2. Itemset generation comparison graph

Inoder to generate the comparison graph the same dataset is used. The vertical axis shows the number of itemset generated and the horizontal axis shows the iteration number.

VI. Conclusion

The proposed algorithm suggests a priority model approach to the traditional Apriori algorithm. Method enhances the accuracy of the rules generated in applications where infrequent items make an impact in the output. Here a dynamic support value is used opposed to the static support value used by the classical Apriori algorithm. From the qualitative content analysis of water, it is clear that the rules generated with Apriori will establish the water as usable, which does contain toxic rare elements because the rules generated doesn't include compounds of mercury and lead. Whereas the proposed scheme declares the water as unusable because of the presence of the high impact rare items in the final rules generated. Results confirm that, with extended inter-transactional association, absolute and accurate relations were able to mine from the database. 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 towards the Institute of Technology and Water authority of Kerala 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. 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.
9. New Improvement on Apriori Algorithm-Lei Ji, Baowen Zhang, Jianhua Li Information Security Engineering School, Shanghai Jiaotong University, Shanghai-C2006 IEEE
10. 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.

11. 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.

12. 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

13. An Improved Apriori Algorithm based on Matrix Data Structure-Shalini Dutt, Naveen Choudhary & Dharm Singh Maharana Pratap University of Agriculture and Technology, India, Global Journal of Computer Science and Technology: CSoftware & Data Engineering Volume 14 Issue 5 Version 1.0 Year 2014.

14. An improved Apriori algorithm based on the matrix-feng wang, Yong-hua LI School of Computer Science and Technology,University of Technology Wuhan, China, 2008 International Seminar on Future BioMedical Information Engineering.

15. An Improved Apriori Algorithm Based On the Boolean Matrix and Hadoop- Honglie Yua, Jun Wenb, Hongmei Wangc ,Li Jun , School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu, China Procedia Engineering 15 (2011) 1827 – 1831.

16. Modified Versions Of Apriori Algorithm: A Survey-Rini Christopher, M.Tech in CSE, Sini.S.Raj, Asst.Professor in CSE, Marian Engineering College, Trivandrum, India-International Journal of Emerging Trends in Engineering and Development Issue 5, Vol. 3 (April.-May. 2015)

