Weblog and Retail Industries Analysis using a robust modified Apriori algorithm

Jayakumar Kaliappan, S. Mohan Sai, K. Shaily Preetham

Abstract**:** ***With the rapid development of the usage of the Internet. There is a lot of information on the Internet. The Weblogs and retail industries accumulate a large amount of transactions data. It is important that this data has to be properly used to find the hidden patterns and mine the association rules among these data. First part is the data pre-processing phase which is most important but makes the data of good quality and the next is the pattern discovery phase for that we use the Apriori algorithm. The Apriori algorithm is used to find association rules and commodities and could help in promoting sales and user interaction. In this paper, we propose a modified algorithm by reducing the dimensions of the items in the transaction set database Dk and Candidate set Ck. This also provides proof of the proposed optimization techniques. This paper introduces the hashing technique for finding the frequencies of items in the one step with O (1) time complexity. The buffer is introduced to keep track of all the unwanted sets which helps in increasing the memory efficiently and computation. This paper gives the application with an example and implementation with proposed optimization techniques.***

Index Terms**:** ***Apriori Algorithm; Candidate sets; Web Log Data; Data Cleaning; Hashing; Retailer Stores; Association Rules; Frequent Itemsets***

I. INTRODUCTION

A huge amount of data is available in the present world and this data such as from retailer store or any other business sectors. It data can be used find the hidden patterns and the relations among the itemsets in the transactions which could help the business to promote or improve the profits. Data mining, data pre-processing and data cleaning are figured out to be the most important part of our database in this technical word. This can be answered in the best way of knowing the value of data and storage space. The different techniques available for the above-stated mining, pre-processing and cleaning of data makes our work a bit easier. One of the most important and used technique in this particular area of data sets is the Apriori algorithm. This technique was introduced by Rakesh Agarwal in 1993. An Apriori algorithm plays a vital role in determining the frequent data sets of a given particular data. The important terms used in the algorithm are support value, confidence value, and associative rules. Support value is a threshold value on basis of which the item sets are sorted out.

Revised Manuscript Received on April 07, 2019.

Dr. Jayakumar Kaliappan, Associate Professor, School of Computer Science, VIT University, Vellore, India.

S. Mohan Sai, UG Student, School of Computer Science, VIT University, Vellore, India.

K. Shaily Preetham, UG Student, Department of computer Science and Engineering, Indian Institute of Technology, Bhubaneswar, India

Confidence value gives the probability of having an item in the set with respect to other elements in the set. Associate rules are the final output of the algorithm showing the relation level between different items in a set. The methodology behind the algorithm can be clearly explained using an example like a large dataset of store transactions. The stored transaction dataset is considered and

iterated in-order to find out the most frequently purchased combo or item set based on support value and carried forward using different iteration levels starting with calculating the frequency of each item and deleting the items from the list with a frequency value less than the support value and then a set of two elements and so on are iterated in the same manner. From the finally obtained sets, the Associative rules are figured out on basis of confidence value calculated. On the basis of above-obtained rules, we conclude dependency of an item with other and probability of having an item in presence of other items. This paper basically describes the implementation of hashing process to calculate the frequency of each item in data set at the easiest way possible and a method to remove unwanted data from the given set in order to free up space resulting in the increment of compilation speed.

II. LITERATURE REVIEW

In this paper, the authors selected some data mining algorithms i.e., Éclat, Apriori, and FP-Growth. These algorithms are tested by using varies cases for estimating the performance as per both performance and memory usage. The database used is a server access log file of three different sets. And they came to a conclusion that for big databases the performances of Apriori is better than the remaining two algorithms. And they also concluded that these algorithms have their own area of implementation where they can be used efficiently and can be executed to their full potential [1]. This paper proposed a new web mining algorithm named Efficient Web Miner or E-Web Miner and checked the efficiency by comparing it with the Improved Apriori algorithm. In this paper, the new web mining algorithm is used for web log analysis. The new algorithm is proposed to improve the time complexity of the Apriori algorithm and also provides an improved candidate set pruning. Here the main objective was to reduce the elements in every candidate set without repeating for changes in larger sets and by pruning, the numbers in the candidate sets are decreased to reduce the repetitive database is eliminated. This paper showed that the new algorithm consumes less time than the Improved Apriori algorithm with graphs and tables [2]. This paper well-focussed on the method of using weblog technique in the Apriori algorithm to find out the associate rules of the data set. It shows the clear example of the proposed algorithm in figuring out a weblog showing the IP address, website searched, userid, time and date of search in a clear modified manner. And finally makes an interactive means of using that weblog the technique to process the Apriori and find out the Associate rules [3]. This paper is developed on the basic view of explaining different mining techniques like- Web mining. Web structure mining- Web content mining; Multi-media mining. Text-mining. The explanation of the most important and frequently used algorithms in the above-stated areas and also the applicable standards of these algorithms. Some clustering techniques are also well defined and explained. The paper arrives at a conclusion of proving all the different parts of our life where these techniques play a vital role and the way it's clearing out human problems [4]. This paper introduced an improved Apriori algorithm for Mining associations. This new algorithm improves the efficiency of generating association rules. The improvement of the algorithm is mainly produced by reducing the query frequency and storage i.e. the algorithm will not create a new candidate when the same itemset is introduced. They concluded that due to the improvement reduces the number of database scanning and the redundancy while generating subtests and verifying them in the database efficiently [5]. This paper proposed an improved Apriori algorithm as count based pruning of the candidate itemsets. They used cloud computing to increase the efficiency of the existing Apriori algorithm. They created three tables i.e. table of transaction, items record in transaction and items associating Items record in the transaction. From these, it becomes memory effective and this algorithm is best used for data mining because its records are save in the cloud for future calculation [6]. The current study presents the most efficient computation for finding the association rules among the large dataset. The proposed methodology was compared with the traditional Apriori algorithm by using the retailer data and the weblog data for the analysis of the computational expenses and performance.

III. PROPOSED METHOD

A. Data Pre-processing

An important method before making way to the adopted data mining method is data pre-processing. The techniques used to modify or regulate the dataset from the whole data as per the requirement in the mining process are called Data pre-processing technique. This also includes the transformation of data from one format to the other format, like from raw format to an understandable form required in the real-life implementation of data. Different data pre-processing techniques are data cleaning, Data Integration, Data Transformation, Data Reduction.

Data Cleaning

A technique where inconsistency is cleaned out is data cleaning. As per the requirement in the Apriori process, all the transaction are to be with a properly defined value. In the grocery store due to errors while taking a written bill, noisy data ie. Improper format, incomplete data or any missing values are cleared and cleaned in this process.

Data Integration

Combining the data from different sectors like cosmetics, dairy products, cereals etc. on different dates are brought under the same roof. Because all the data are to be considered to arrive at a final view of the store overview. Data redundancy problem is also solved here if any.

Data Reduction

This is the most important method because it is where the data is cut down. All the data sets that do not make an impact on the final output even on deleting them from the consideration set. In the grocery store, a prescribed threshold value is defined and all those data items and sets having a frequency less than the threshold are deleted as they do not make a visible change in the overall output. A lossless reduction is done.

B. Classical Apriori Algorithm

Apriori Algorithm is one of classical Algorithm. Apriori Algorithm is used to find the associations for the given dataset of transactions of the items. The objective of mining strong association rules is to satisfy minimum support and the minimum confidence from the database transactions (D). This Algorithm is a level-wise search. The Main challenge in the classical Apriori algorithm is searching for the itemsets pattern in the whole database. In this process first, the frequencies of each element are found and stored in candidate set C1 the values which satisfy the threshold values are then kept in the Reduced set (R1). Then after the permutations of the 2-items are searched in the database D. Then these frequencies values of the itemsets are kept in the candidate set of 2-itemset (C2). The values which satisfy the minimum support are placed in a reduced set of 2-itemset (R2). This algorithm is supposed to scan the whole database for the frequencies of the itemsets for every iteration. After the generation of the reduced set of 2-itemsets, Then for the 3rd iteration the value of k is 3 the permutation in candidate set C3 is formed with (k-2) elements (.i.e. if (k>3)) and then the reduced sets are formed those which satisfy the minimum support and the iteration are carried and the Ck and Rk are found in iteration until the Ck is null (φ). The association rules are formed using the minimum confidence. In the Apriori Algorithm, the

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

support represents the occurrence of the item in the whole database. Let the item X in the database.

supp(X) = (Number of Transactions in which X appears)/(Total no of Transactions) (1)

The X, Y are the items present in the dataset. The confidence represents the likely occurrence of the item Y in a transaction when the element X occurs in the transaction. This is like a kind of conditional probability P (X|Y).

$$\text{Conf}(x) = (\text{supp}(X \cup Y))/(\text{supp}(X)) \quad (2)$$

C. Improvement in Apriori Algorithm:

The main objective of the improvement of the algorithm are to reduce the search space and to use in the large databases. The efficiency of the algorithm can be improved by considering 3 aspects.

1. In the first step, the frequencies of each item can be found by the hashing instead of searching for the whole database (D1).
2. For every iteration, the database (Dk) is updated by removing the unwanted relations these are of two types.
   i) While the iteration in k=2 the elements in the buffer is removed from the database (D1). The buffer consists of the elements of C1 which don't satisfy minimum support (which will no longer usable in the further operations.).
   ii) In Dk-1 the itemsets with the size <= k-1 can be deleted from the database and updated.
3. To remove the unwanted permutation of candidate set Ck which are formed from Rk-1 with (k-2) elements in common. i.e. if (k>=2).

D. Description of Optimized Apriori Algorithm

The Apriori algorithm generates the frequent itemset which results in a reduced set (Rk) from the candidate itemset Ck and the transactions of the itemsets are present in the database D1. In this process of 1st iteration the all transaction frequencies can be found using the hashing function which will retrieve the frequencies of the items in O(1) time complexity. The candidate set (C1) is found using the hashing method. And the values of C1 which satisfies minimum support are used in R1 and the remaining values are present in Buffer (B). Then after the database is updated by removing items which are present in the buffer in the 2nd iteration. And some unwanted itemsets are the size of the itemsets in Dk-1 are less than k in the kth iteration. The candidate set (Ck) is formed with the permutations of the elements in Rk-1 with (k-2) items in common. The permutations in candidate set(Ck) can be further reduced by the itemsets in the buffer as the subset of the itemset in the possible permutations from the Rk-1. Because the itemset present in the buffer is the itemsets which don't satisfy the minimum support so, the permutation will no longer lead to frequent itemset. And the reduced set (Rk) formed using the elements which satisfy the minimum support from the Ck. And this process will be iterated until the Ck = φ.

Statement: The itemset in the transactions of database Dk at kth iteration with itemset of the item less than k items in the Dk-1 then this itemset can be ignored in Dk. And the itemset in candidate set Ck-1 which doesn't satisfy the minimum support and this is the sub itemset of the possible permutation candidate set (Ck) then these permutations can be removed. Proof: In the first iteration the items in the buffer are those which doesn't satisfy minimum support it means that the items are not frequent in the given dataset and the items which are not frequent will no longer be used in the further iterations and thus we can delete these items from the database. The itemsets in the database at the kth iteration in the candidate set will contain k-itemset combination. So, the database in the kth iteration (Dk) doesn't require the itemsets of a number of items less than k. So, these all permutations can be removed from the database and will lead to fast computing and will deduce the I/O spending.The itemset in the buffer and which is the subset of the combination of the Ck and this can be removed from the candidate set. Because the itemset present in the buffer are which doesn't satisfy the minimum support these itemsets are less frequent. And this itemset as the subset of the permutation in Ck¬ will also not satisfy the minimum support. So, these values can be deleted from the candidate set Ck. This operation will improve the search space by removing the unwanted relations in the entire process.

E. Pseudo Code of the Optimized Algorithm

Here the D1 is the database of transactions and the Ck represents the candidate set of all the kth iteration. Rk this is the reduced set at the kth iteration. And 'min_support' is the minimum support (threshold value). Buffer B to store the value of not required from the database.

Input of the Algorithm: The database of transactions are given as input .i.e D_1

Output of the Algorithm: This will produce the frequent subsets of the whole transactions.

```
(1) C1 = hash_1-itemset(D1);
(2) for (i=0; i<C1.length; i++)
          // C1.length➔no of sets in the candidate set
(3)         If (frequency (C1 [i]) >= min_support)
          //C1[i] ➔ gives the sets in the position i
(4)         R1.append (C1 [i].value)
(5)         else if (frequency (C1 [i]) < min_support)
(6)       B.append(C[i].value)
(7) for (k=2; $C_k \neq \varphi$; k++)
(8)      $D_k$= update_database (B,k)
(9)      $C_k$= permutation ($R_{k-1}$,B)
(10)         empty(B)
(11)    for (i=0; i<Ck.lenght; i++)
```

```
(12)        if (frequency(Ck[i])> min_support))
(13)        $R_k$.append (C1 [i].value)
(14)        else if (frequency (Ck[i]) < min_support)
(15)        B.append (C1 [i].value)
(16) Return $R_k$;
```

Procedure for function **hash_1-itemset(D1)**

While entering the value into the database these values are passed through the hash function such that just by placing the keyword in the hash function gives the frequency in O(1) time. In the items are hashed and stored in the array X.

The values are stored in the array by consider the any item can be represented in the numeric form.

```
(1) for (i=0; i<$D_1$.length; i++)
(2)   for (j=0; j<$D_1$ [i].length; j++)
(3)       X [$D_1$ [i] [j]] ++;
          For retrieving the values from the hash table of the
          database $D_1$.
 (4) freq (o) =X[o];
      // o is the item which the frequency has to find.
```

Function for the **database_update (B, k)**

```
(1) if (k-1 = 1)
(2)   x € B
(3)   for (i=0; i<$D_1$.lenght; i++)
(4)       for (j=0; j<$D_1$ [i].length; j++)
(5)           if(x€$D_1$ [i] [j])
(6)           remove the item $D_1$ [i] [j] from the set.
(7)       if ($D_1$ [i].length<2)
(8)       remove the itemset from the dataset.
(9)  else if (k>2)
(10) for (i=0; i<$D_{k-1}$.length; i++)
(11)   if (setsize ($D_{k-1}$[i]) <k)
(12)   remove from the database.
(13) return database.
```

Function for the **permutation (R_{k-1}, B)**

```
(1) from the x € B
(2) from the $R_{k-1}$ sets if ((k-2) items in common)
(3)   if(x != subset of the any combinations in $R_{k-1}$)
(4)        These combinations are considered in $R_k$.
(5)   return permutations.
```

F. Usage of Apriori Algorithm for Weblog Data

In the process of finding the pattern in the weblog data. We need to find the frequent website links navigation pattern in the whole datasets. The general weblog dataset consists of IP Address, time of navigation of the website, link of the website and status number. From the database, we don't require the time and status number which are helps us to reduce the computational cost which will improve the performance of the model. In this process of analysing the weblog data in the whole transactions, one single IP address represents one single person navigation of websites. Now have to convert the whole raw data into standard format i.e. we keep track of all the websites navigated by the one single user. Now we have the whole set of data with different users with different websites.

G. Demonstration of Conversion of Whole raw weblog data into the Standard format

The conversion of the dataset into the proper format for finding the frequent websites patterns experienced by the user. The weblog data after gathering all the websites navigated by a single user. The user with single IP Address navigates all the websites as shown in the URL which is shown in Table 3.

IP Address	Time	URL(Hyperlink)	Status
10.128.28.1	12/Feb/2018	http://abc1.html	200
	12/Feb/2018	http://abc2.html	200
	12/Feb/2018	http://abc1.html	200
	13/Feb/2018	http://abc3.html	200
	14/Feb/2018	http://abc4.html	200
	14/Feb/2018	http://abc5.html	200
10.128.36.1	12/Feb/2018	http://abc6.html	200
	12/Feb/2018	http://abc3.html	200
	14/Feb/2018	http://abc7.html	200
	14/Feb/2018	http://abc2.html	200
	15/Feb/2018	http://abc8.html	200
10.128.45.1	12/Feb/2018	http://abc3.html	200
	12/Feb/2018	http://abc9.html	200
	12/Feb/2018	http://abc8.html	200
	12/Feb/2018	http://abc6.html	200
	13/Feb/2018	http://abc1.html	200
10.128.33.1	12/Feb/2018	http://abc1.html	200
	12/Feb/2018	http://abc3.html	200
	14/Feb/2018	http://abc7.html	200
	16/Feb/2018	http://abc8.html	200
	16/Feb/2018	http://abc5.html	200

Table 3: Gathering all the websites navigated by the user

Then after gathering all the websites navigated by the user, we can remove status column and time. This will help in improving the efficiency of the model. Now, this can be represented in the standard format by that each link can be represented with a unique number and they are represented in Table 4.

User(IP Address)	Hyperlinks Navigated
User 1(10.128.28.1) ➔ TID0001	1,2,1,3,4,5
User2(10.128.36.1) ➔ TID0002	6,3,7,2,8
User3(10.128.45.1) ➔ TID0003	3,9,8,6,1
User4(10.128.33.1) ➔ TID0004	1,3,7,8,5

Table 4: Representation each user with their corresponding hyperlinks

Then after bringing the data into the standard format now, we can find the frequent itemsets using the Modified Algorithm which is discussed in the Demonstration of modified Algorithm.

H. Demonstration of Modified Algorithm

The sample transaction of the retail store is shown in Table 5. Table 6 describes the representation of each item with its corresponding representation I_i the minimum support for the sample are taken as min_support= 3 and the min_cofd =50% is the minimum confidence. The Algorithm steps are shown as follows. The results of the algorithm are shown in the following steps.

ITEM NO	ITEM
I1	Onion
I2	Sugar
I3	Cup
I4	Coffee
I5	Toothpicks
I6	Soap
I7	Brush

TABLE 1

ITEM ID	ITEMS
TID1001	Sugar, Cup, Coffee, Toothpick
TID1002	Sugar, Cup, Brush
TID1003	Onion, Cup
TID1004	Onion
TID1005	Onion, Cup, Brush
TID1006	Toothpick, Sugar
TID1007	Sugar, Cup, Coffee, Soap
TID1008	Onion, Sugar, Cup
TID1009	Onion, Sugar, Cup, Coffee, Soap
TID1010	Sugar, Cup

TABLE 2

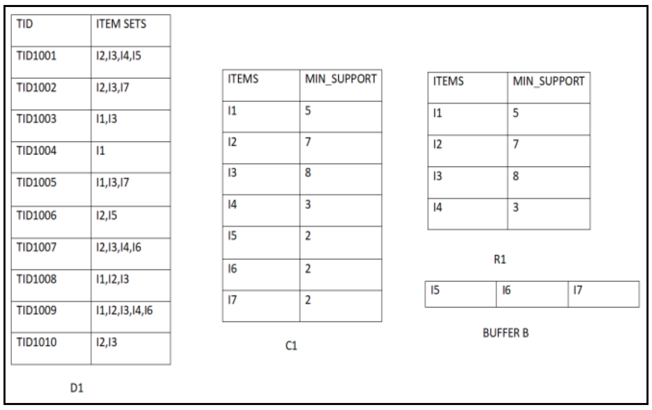
Table 5: Describes a representation of each item of the dataset

Table 6: Sample Transaction off retailer store

Step 1: The first step of the algorithm is the candidate set (C_1) has to be produced from the Dataset of transactions (D_1). To obtain the Candidate set instead of simply scanning in the whole database we applying the hashing technique to retrieve the frequencies of each item in the dataset from the D_1.

Step 2: In the next step we generate the reduced set (R1) in the 1st iteration. Here in this step, we evaluate the itemsets in the candidate set (C1) and the values which crosses the minimum support (min_support) then we place the sets in the reduced set (R1). And the status is shown the Figure 1.

Step 3: In this previous step the I_5, I_6, I_7 are the items with less than the threshold value .i.e. min_support. These items are taken into the buffer B. In the 2nd iteration while generating the candidate set (C2) these are formed using the combinations of the R1. In this set, none of the combinations consists of the elements which are in the buffer B. So, these elements can be deleted from the database. So, from T1001 I_5 is removed. And from T1002 I_7 is removed and all the transaction which contain I_5, I_6, I_7 are removed and in T1004 contains only one item (I_1) and in T1006 after the removal of I_5 which contain only one item (I_2) here in the next step which contain 2 items in a set so these can be removed from the database D1 and updated to D2. This D2 can be seen in the Figure 2 for generating C2 we use self-joining (R1⋈R1) and these leads in 6 combinations {I_1I_2, I_1I_3, I_1I_4, I_2I_3, I_2I_4, I_3I_4} these are itemsets of C2.



TID	ITEM SETS
TID1001	I2,I3,I4,I5
TID1002	I2,I3,I7
TID1003	I1,I3
TID1004	I1
TID1005	I1,I3,I7
TID1006	I2,I5
TID1007	I2,I3,I4,I6
TID1008	I1,I2,I3
TID1009	I1,I2,I3,I4,I6
TID1010	I2,I3

D1

ITEMS	MIN_SUPPORT
I1	5
I2	7
I3	8
I4	3
I5	2
I6	2
I7	2

C1

ITEMS	MIN_SUPPORT
I1	5
I2	7
I3	8
I4	3

R1

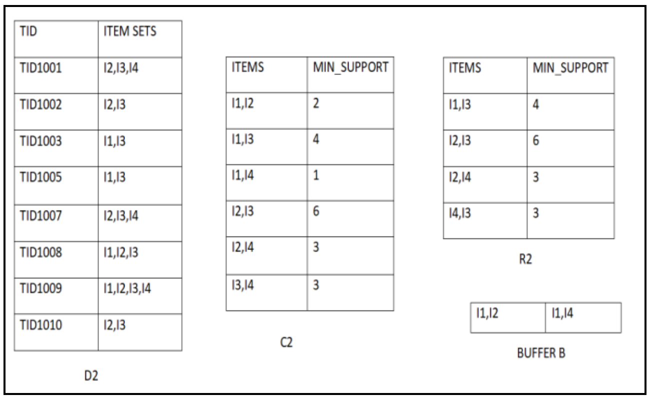
I5	I6	I7

BUFFER B

Fig 1: Status of the tables after the second iteration

Step4: After these combinations (C2) are generated the frequencies of the itemsets has to be counted from the D2 which helps rather than searching the whole database. Then after the min_support value is 3 and the itemsets less than the threshold value are (I_1I_2, I_1I_4) all the remaining values go to the Reduced set (R2).

Step5: Now the buffer is updated with the new values less than the min_support of C2 (.i.e. I_1I_2, I_1I_4). This is to keep track of the unwanted elements. The status of the tables is shown in Figure 2.



TID	ITEM SETS
TID1001	I2,I3,I4
TID1002	I2,I3
TID1003	I1,I3
TID1005	I1,I3
TID1007	I2,I3,I4
TID1008	I1,I2,I3
TID1009	I1,I2,I3,I4
TID1010	I2,I3

D2

ITEMS	MIN_SUPPORT
I1,I2	2
I1,I3	4
I1,I4	1
I2,I3	6
I2,I4	3
I3,I4	3

C2

ITEMS	MIN_SUPPORT
I1,I3	4
I2,I3	6
I2,I4	3
I4,I3	3

R2

I1,I2	I1,I4

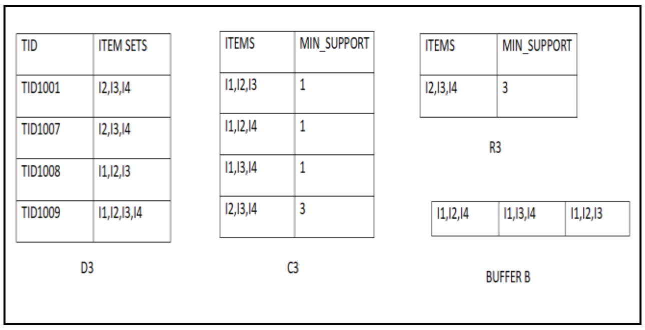
BUFFER B

Fig 2: Status of the tables after the second iteration

Step 6: Now the database has to be updated the no of itemsets less than value 3(3rd iteration) in D2 are removed from the database and the updated to D3.

Step 7: The permutation of the table C3 were formed from the R2 combinations were made with one item in common using the operation self-joining (R1⋈R1). And all the combination made are {I_1I_2 I_3, $I_1I_3I_4$, I_2I_3 I_4} in combination I_1I_2 I_3 the I_1I_2 is present in Buffer. So, this permutation can be ignored. In the combination $I_1I_3I_4$ the I_1I_4 is there in the buffer. So, this can also be ignored.

Step 8: Now the only permutation present in C3 are {$I_2I_3I_4$} and this has min_support value 3. The min_support is 3. So, this value is carried in R3. The status of the tables is shown in figure 3.



TID	ITEM SETS
TID1001	I2,I3,I4
TID1007	I2,I3,I4
TID1008	I1,I2,I3
TID1009	I1,I2,I3,I4

D3

ITEMS	MIN_SUPPORT
I1,I2,I3	1
I1,I2,I4	1
I1,I3,I4	1
I2,I3,I4	3

C3

ITEMS	MIN_SUPPORT
I2,I3,I4	3

R3

I1,I2,I4	I1,I3,I4	I1,I2,I3

BUFFER B

Fig 3: Status of the table after the third iteration

Step 9: Now the R3 has only 1 itemset of itemset sized 3. So that the C4 = φ by this step the algorithm will terminate and the stop giving the frequent itemsets.

Step 10: After finding the frequent elements the association rules can be found and the rules, in this case, I_2^I_3=>I_4 which has the confidence value (3/6) =0.5*100=50% so this is one of our association rules.

From this above example based on the transaction set and the association rules, we can conclude that those who ever buy sugar and cup will also buy the coffee in this way we can find the frequent itemsets in an optimized manner.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

IV EXPERIMENTAL RESULT AND ANALYSIS

For the experiment of analysis of the retailer store data we have used a dataset from a supermarket, Vaishnavi Supermarket database consists of around 100K transactions. And this data is helpful for analysing the data of the retailer store. The algorithm was implemented and applied on the same dataset which has been for the minimum support count 10, 20, 30, 40, 50 and 60. Apriori algorithm works fine with a smaller data set. But the real-time data is not small and the Apriori algorithm is not that efficient and the implementation of the proposed algorithm has been done so that we could analyse the results and can see which works best. Now we can find the responses of the dataset with the minimum support count of 10 and the time taken to find the frequent itemsets and which can be viewed in Table 5.

Various Parameters	Apriori Algorithm	Proposed Algorithm
Size of the Data	78MB	78MB
Number of Transactions	100K	100K
Number of different items	267 items	267 items
Number of Candidate sets	9687	9687
Time for computation(ms)	8756	860

Table 5: Responses of both general and proposed algorithm over the dataset

Now the comparison of the dataset with different minimum support itemsets are shown in table 6. In table 6 we can find that the time taken for the computation for finding the frequent itemsets decreases as the minimum support decreases and that is due as the minimum support represents the frequency of occurrence of the itemset in the whole database. As the minimum support count increases the chance of occurring the items decreases and hence the time for the computation of the data set decreases. All the values of time taken for the computation of finding the frequent itemsets and the optimization rate is shown in table 6.

Minimum Support	Apriori Algorithm (time)	Proposed Algorithm (time)	Optimization Rate (%)
10	8756 ms	863 ms	90.1
20	6635 ms	760 ms	88.5
30	5992 ms	591 ms	90.0
40	3844 ms	422 ms	89.0
50	3201 ms	390 ms	87.8
60	2695 ms	242 ms	91.0

Table 6: Analysis of performance of algorithms over different minimum support

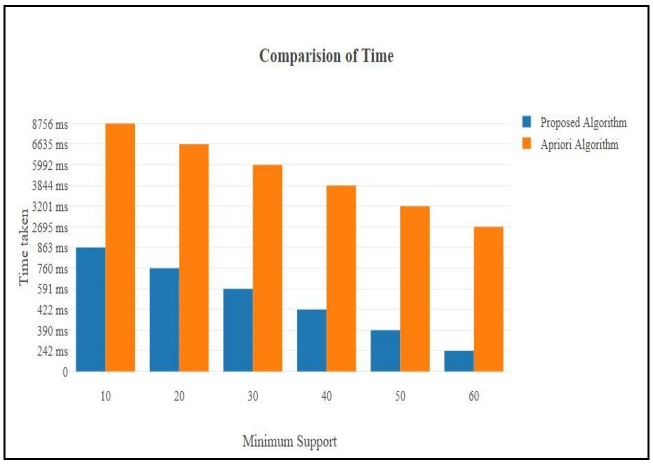


Fig 4: Analysis of Computational time of the Proposed and general Algorithm

V. CONCLUSION

In this paper we have proposed a new modified algorithm for finding the association rules among the large datasets for both the retailer data and weblog data and this process could be further extended by converting any dataset and converting them to the standard format and through computational analysis was conducted for the analysis of performance of proposed algorithm over the general Apriori algorithm. In this process, we have found that our proposed algorithm is 89.4% efficient than the normal algorithm. In this algorithm, we have used hashing to find the frequency of the elements in the first iteration which requires space complexity rather than computational power and then we also introduced buffer to keep track of all the unwanted itemsets which helps the system to improve the performance. Hence, the results of the Proposed Algorithm show more efficient results with less time complexity than the general Apriori algorithm.

REFERENCES

1. Gashaw, Y., & Liu, F. (2017, October). Performance evaluation of frequent pattern mining algorithms using web log data for web usage mining. In Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2017 10th International Congress on (pp. 1-5). IEEE.
2. Yadav, M. P., Keserwani, P. K., & Samaddar, S. G. (2012, March). An efficient web mining algorithm for Web Log analysis: E-Web Miner. In RAIT (pp. 607-613).
3. Sathya, M., & Devi, P. I. (2017, March). Apriori algorithm on web logs for mining frequent link. In Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), 2017 IEEE International Conference (pp. 1-5). IEEE.
4. Li, Y. (2017, July). Research on Technology, Algorithm and Application of Web Mining. In Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), 2017 IEEE International Conference on (Vol. 1, pp. 772-775). IEEE.
5. Chai, S., Yang, J., & Cheng, Y. (2007, June). The research of improved apriori algorithm for mining association rules. In Service Systems and Service Management, 2007 International Conference on (pp. 1-4). IEEE.

6. Zhou, Y., Wan, W., Liu, J., & Cai, L. (2010, November). Mining association rules based on an improved Apriori Algorithm. In Audio Language and Image Processing (ICALIP), 2010 International Conference on (pp. 414-418). IEEE.
7. Nidhi Khurana, Dr. R.K. Datta – "Pruning Large Data Sets for Finding Association rule in cloud: CBPA (Count Based Pruning Algorithm)" 2013 International Journal of Software and Web Sciences (IJSWS)
8. Sai, S. M., Naresh, K., RajKumar, S., Ganesh, M. S., Sai, L., & Nav, A. (2018, April). An Infrared Image Detecting System Model to Monitor Human with Weapon for Controlling Smuggling of Sandalwood Trees. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 962-968). IEEE.
9. Balaji, M., & Rao, G. S. V. (2013, May). An adaptive implementation case study of apriori algorithm for a retail scenario in a cloud environment. In Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on (pp. 625-629). IEEE.
10. Somasundaram, S., Khandavilli, P., & Sampalli, S. (2010, December). An intelligent RFID system for consumer businesses. In Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing (pp. 539-545). IEEE Computer Society.
11. Vijayalakshmi, V., & Pethalakshmi, A. (2015, January). A new enriched exploration of modified algorithm for generating single dimensional fuzzy itemsets. In Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on (pp. 1-6). IEEE.
12. Chen, Y. L., Tang, K., Shen, R. J., & Hu, Y. H. (2005). Market basket analysis in a multiple store environment. Decision support systems, 40(2), 339-354.
13. Khurana, K., & Sharma, S. (2013). A comparative analysis of association rule mining algorithms. International Journal of Scientific and Research Publications, 3(5), 0.
14. Zhang, C., & Ruan, J. (2009, June). A modified apriori algorithm with its application in instituting cross-selling strategies of the retail industry. In Electronic Commerce and Business Intelligence, 2009. ECBI 2009. International Conference on (pp. 515-518). IEEE.
15. Ye, Y., & Chiang, C. C. (2006, August). A parallel apriori algorithm for frequent itemsets mining. In Software Engineering Research, Management and Applications, 2006. Fourth International Conference on (pp. 87-94). IEEE.
16. Chen, Y. L., Tang, K., Shen, R. J., & Hu, Y. H. (2005). Market basket analysis in a multiple store environment. Decision support systems, 40(2), 339-354.
17. Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).