ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 3, September 2013

Improving the efficiency of Apriori Algorithm in Data Mining

Vipul Mangla, Chandni Sarda, SarthakMadra, VIT University, Vellore (632014), Tamil Nadu, India

Abstract: In computer science and Data mining, Data mining, an interdisciplinary subfield of computer science, is the computational process of discovering patterns in large data sets in database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. In Data mining, Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. Association rules are the main technique to determine the frequent item set in data mining. It is sometimes referred to as "Market Basket Analysis". This classical algorithm is inefficient due to so many scans of database. And if the database is large, it takes too much time to scan the database. In this paper we will build a method to obtain the frequent item-set by using a different approach to the classical Apriori algorithm and applying the concept of transaction reduction and a new matrix method, thereby eliminate the candidate having a subset that is not frequent.

Index Terms- **Apriori Algorithm, Association Rules, Candidate-Item sets, Data Mining.**

I. INTRODUCTION

Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. Other algorithms are designed for finding association rules in data having no transactions, or having no timestamps. Other algorithms are designed for finding association rules in data having no transactions, or having no timestamps. The purpose of the Apriori Algorithm is to find associations between different sets of data. It is sometimes referred to as "Market Basket Analysis". Each set of data has a number of items and is called a transaction. The output of Apriori is sets of rules that tell us how often items are contained in sets of data. Data mining also known as Knowledge Discovery in Database (KDD). The purpose of data mining is to abstract interesting knowledge from the large database. From the analysis of abstracted patterns, decision-making process can be done very easily. Association rule is based mainly on discovering frequent item sets. Association rules are frequently used by retail stores to assist in marketing, advertising, inventory control, predicting faults in telecommunication network.

Apriori algorithm represents the candidate generation approach. It generates candidate (k+1) item sets based on frequent k-item sets. Apriori is a Breadth First Search Algorithm (BFS). Now, a method to obtain frequent item-set by using a different approach to classical apriori algorithm, by making a matrix of given example by considering row as transactions and columns as items. By reducing rows and columns from matrix, we will finally produce a frequent item set without scanning database repeatedly.So; this method will increase the efficiency and reduce the time to generate the frequent item-sets.

II. PROPOSED METHOD

The method we propose involves the mapping of the I_n items and T_m transaction from the database into a matrix A with size mxn. The rows of the matrix represent the transaction and the columns of the matrix represent the items. The elements of matrix A are:

$$A= [a_{ij}] = 1, \text{ if transaction i has item j}$$

$$= 0, \text{ otherwise}$$

We assume that minimum support and minimum confidence is provided beforehand.

In matrix A,

The sum of the j^{th} column vector gives the support of j^{th}item.

And the sum of the i^{th} row vector gives the S-O-T, that is, size of i^{th} transaction (no. of items in the transaction).

Now we generate the item sets.

For, 1–frequent item set, we check if the column sum of each column is greater than minimum support. If not, the column is deleted. All rows with rowsum=1 (S-O-T) are also deleted. Resultant matrix will represent the 1-frequent item set.

Now, to find 2-frequent itemsets, columns are merged by AND-ing their values. The resultant matrix will have only those columns whose columnsum>=min_support. Additionally, all rows with rowsum=2 are deleted. Similarly the k^{th} frequent item is found by merging columns and deleting all resultant columns with columnsum<min_support and rowsum=k.When matrix A has 1 column remaining, that will give the k^{th} frequent item set.

III. ALGORITHM

A. Basic algorithm

1. Create matrix A
2. Set n=1

```
3.  While(n<=k)
        • If(columnsum(colj)<min_support)
        • If(rowsum(row i)==n)
          Delete row i;
        • Merge(col j, col j+1)
        • n=n+1
4.  end while
5.  display A
```

IV. NUMERICAL ILLUSTRATION

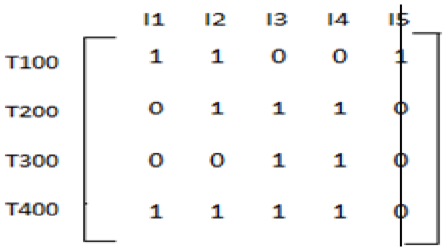
Consider the following example:

Transactions	Items
T100	I1,I2,I5
T200	I2,I3,I4
T300	I3,I4
T400	I1,I2,I3,I4

The above example shows the number of transactions and items in table. Consider minimum support to be given as 2. Now, we will draw the matrix from above table to show the occurrence of each item in particular transaction, i.e.:

$$\begin{array}{c|ccccc|} & I1 & I2 & I3 & I4 & I5 \\ T100 & 1 & 1 & 0 & 0 & 1 \\ T200 & 0 & 1 & 1 & 1 & 0 \\ T300 & 0 & 0 & 1 & 1 & 0 \\ T400 & 1 & 1 & 1 & 1 & 0 \end{array}$$

Now, to find 1-frequent item set, remove those columns whose sum is less than minimum support i.e. 2 and those rows that sum is equal to finding frequent item set which is 1 for above case. So, the matrix after removing particular row and column would be:

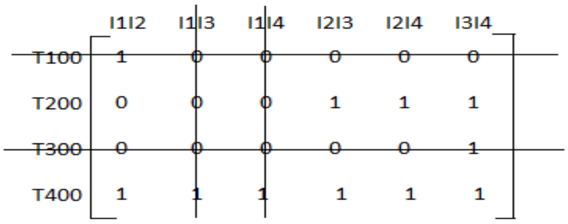


$$\begin{array}{c|ccccc|} & I1 & I2 & I3 & I4 & I5 \\ T100 & 1 & 1 & 0 & 0 & 1 \\ T200 & 0 & 1 & 1 & 1 & 0 \\ T300 & 0 & 0 & 1 & 1 & 0 \\ T400 & 1 & 1 & 1 & 1 & 0 \end{array}$$

So, the above matrix represents the items present in 1-freq item set. Combine the item by taking AND to get matrix of 2-freq item set, which can be represented as:

$$\begin{array}{c|cccccc|} & I1I2 & I1I3 & I1I4 & I2I3 & I2I4 & I3I4 \\ T100 & 1 & 0 & 0 & 0 & 0 & 0 \\ T200 & 0 & 0 & 0 & 1 & 1 & 1 \\ T300 & 0 & 0 & 0 & 0 & 0 & 1 \\ T400 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Now, after removing rows and columns following the above method, the reduced matrix would be like:

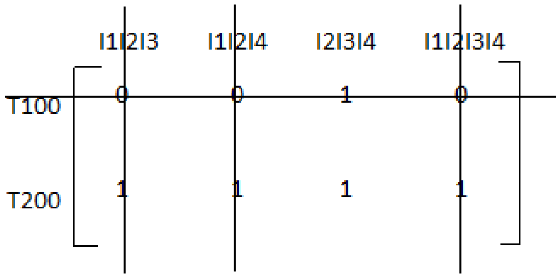


$$\begin{array}{c|cccccc|} & I1I2 & I1I3 & I1I4 & I2I3 & I2I4 & I3I4 \\ T100 & 1 & 0 & 0 & 0 & 0 & 0 \\ T200 & 0 & 0 & 0 & 1 & 1 & 1 \\ T300 & 0 & 0 & 0 & 0 & 0 & 1 \\ T400 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

For finding 3-frequent set, follow the same procedure and combine item sets as follow:

$$\begin{array}{c|cccc|} & I1I2I3 & I1I2I4 & I2I3I4 & I1I2I3I4 \\ T100 & 0 & 0 & 1 & 0 \\ T200 & 1 & 1 & 1 & 1 \end{array}$$

Remove those columns whose sum is less then 2(min support) and those rows whose sum is less than 3, so the reduced matrix is:



$$\begin{array}{c|cccc|} & I1I2I3 & I1I2I4 & I2I3I4 & I1I2I3I4 \\ T100 & 0 & 0 & 1 & 0 \\ T200 & 1 & 1 & 1 & 1 \end{array}$$

So, this is the final reduced matrix for above given example. The final frequent item set (3-freq item set) is **I2I3I4**.

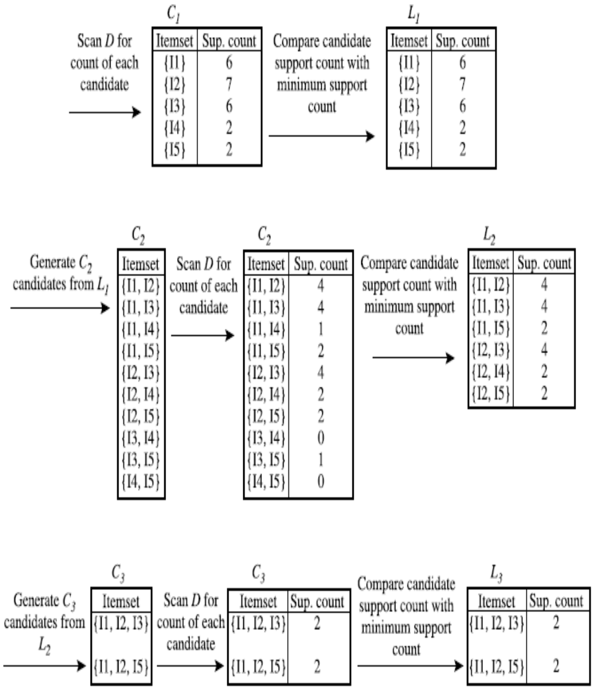
V. ANALYSIS OF APRIORI ALGORITHM

In this part, We will compare time efficiency in finding frequent item set by normal apriori algorithm and by method proposed in this paper i.e. Transaction Reduction and Matrix method. Now, consider the following example and calculate time to generate frequent item sets by using basic apriori algorithm.

Problem :(Ref.) This problem is taken from the book 'Data Mining concepts and techniques by Jiawei Han and MichelineKamber'

TID	List of item_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	
I1,I2,I3,I5	

Solution.


Scan D for count of each candidate
Compare candidate support count with minimum support count
Generate C_2 candidates from L_1
Scan D for count of each candidate
Compare candidate support count with minimum support count
Generate C_3 candidates from L_2
Scan D for count of each candidate
Compare candidate support count with minimum support count

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Fig. Generation of candidate itemsets and frequent itemsets, where the minimum support count is 2.

(**Ref.** of image: this image is available in 'Data Mining concepts and techniques by Jiawei Han and MichelineKamber')

So, this is the frequent item set obtained by normal apriorialgorithm.We can see this method takes a lot of time to solve the problem by scanning database frequently. Now, we will solve this problem by our method of transaction reduction using matrices which is as follow:

VI. EXAMPLE OF OUR ALGORITHM

Step 1: Draw matrix from the given table with rows as transactions and columns as items.

Step2: check row_sum and column_sum of generated matrix. Now, remove those rows whose row_sum is less than or equal to value of k (in k-frequent item set generation, like for finding 1-frequent item set, value of k=1) and remove those columns whose column_sum is less than min_support(given).

Step 3: Now, combine all two possible rows starting from initial row, to generate 2 frequent-items and take AND of values of these rows.

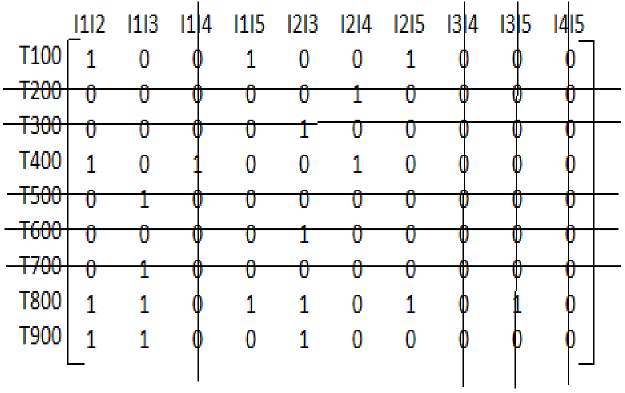
Step 4: Now again find the rum_sum and column_sum values of generated matrix and follow the same procedure as done above.

Step 5: combine 3 rows (all possible) to generate 3-frequent item set and follow the same procedure until no further row combination is possible.

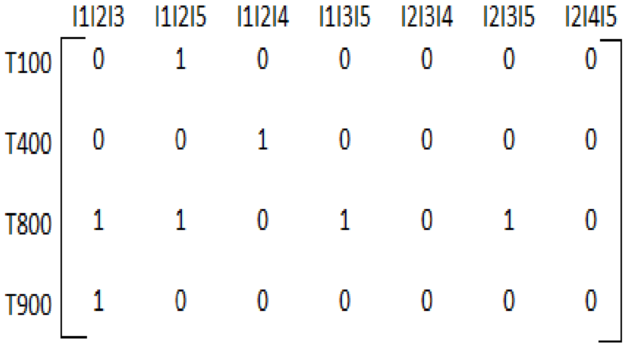
Step 6: Finally, we will get the possible frequent item sets of given table.

	I1	I2	I3	I4	I5
T100	1	1	0	0	1
T200	0	1	0	1	0
T300	0	1	1	0	0
T400	1	1	0	1	0
T500	1	0	1	0	0
T600	0	1	1	0	0
T700	1	0	1	0	0
T800	1	1	1	0	1
T900	1	1	1	0	0

	I1I2	I1I3	I1I4	I1I5	I2I3	I2I4	I2I5	I3I4	I3I5	I4I5
T100	1	0	0	1	0	0	1	0	0	0
T200	0	0	0	0	0	1	0	0	0	0
T300	0	0	0	0	1	0	0	0	0	0
T400	1	0	1	0	0	1	0	0	0	0
T500	0	1	0	0	0	0	0	0	0	0
T600	0	0	0	0	1	0	0	0	0	0
T700	0	1	0	0	0	0	0	0	0	0
T800	1	1	0	1	1	0	1	0	1	0
T900	1	1	0	0	1	0	0	0	0	0

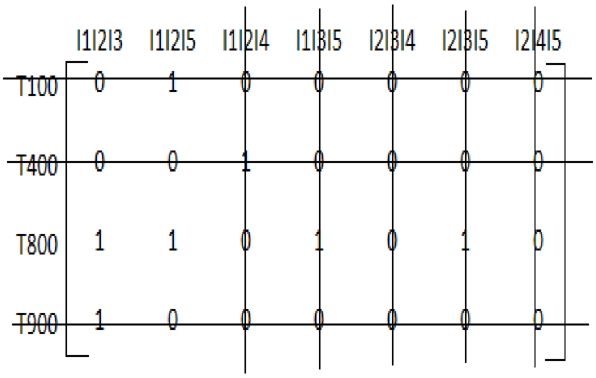


	I1I2	I1I3	I1I4	I1I5	I2I3	I2I4	I2I5	I3I4	I3I5	I4I5
T100	1	0	0	1	0	0	1	0	0	0
~~T200~~	0	0	0	0	0	1	0	0	0	0
~~T300~~	0	0	0	0	1	0	0	0	0	0
T400	1	0	1	0	0	1	0	0	0	0
~~T500~~	0	1	0	0	0	0	0	0	0	0
~~T600~~	0	0	0	0	1	0	0	0	0	0
~~T700~~	0	1	0	0	0	0	0	0	0	0
T800	1	1	0	1	1	0	1	0	1	0
T900	1	1	0	0	1	0	0	0	0	0



	I1I2I3	I1I2I5	I1I2I4	I1I3I5	I2I3I4	I2I3I5	I2I4I5
T100	0	1	0	0	0	0	0
T400	0	0	1	0	0	0	0
T800	1	1	0	1	0	1	0
T900	1	0	0	0	0	0	0

(Ref. all these matrix these images are not available anywhere, have been drawn on paint and been pasted over here)



	I1I2I3	I1I2I5	I1I2I4	I1I3I5	I2I3I4	I2I3I5	I2I4I5
~~T100~~	0	1	0	0	0	0	0
~~T400~~	0	0	1	0	0	0	0
T800	1	1	0	1	0	1	0
~~T900~~	1	0	0	0	0	0	0

So, the generated 3-frequent item of the given problem are**I1I2I3, I1I2I5.**

VII. CONCLUSION AND FUTURE SCOPE

In this paper, Apriori algorithm is improved based on the properties of cutting database. The typical Apriori algorithm has performance bottleneck in the massive data processing so that we need to optimize the algorithm with variety of methods. The improved algorithm we proposed in this paper not only optimizes the algorithm of reducing the size of the candidate set of k-itemsets, but also reduce the I / O spending by cutting down transaction records in the database. The performance of Apriori algorithm is optimized so that we can mine association information from massive data faster and better. Although this improved algorithm has optimized and efficient but it has overhead to manage the new database after every generation of Matrix. So, there should be some approach which has very less number of scans of database. Another solution might be division of large database among processors.

REFERENCES

[1] "Data Mining - concepts and techniques" by Jiawei Han and MichelineKamber.

[2] "Improved Apriori Algorithm using logarithmic decoding and pruning" paper published by SuhaniNagpal, Department of Computer Science and Information Technology, Lovely Professional University, Punjab (INDIA).

[3] "Improving Efficiency of Apriori Algorithm Using Transaction Reduction" by Jaishree Singh*, Hari Ram**, Dr. J.S. Sodhi, Department of Computer Science & Engineering, Amity School of Engineering and Technology, Amity University, Sec-125 NOIDA, (U.P.),India.

[4] Wanjun Yu; Xiaochun Wang; Erkang Wang; Bowen Chen;, "The research of improved apriori algorithm for mining association rules," Communication Technology, 2008. ICCT 2008 11th IEEE International Conference on, vol., no.,pp.513-516, 10-12 Nov. 2008.

[5] SixueBai, Xinxi Dai, "An Efficiency apriori Algorithm: P_Matrix Algorithm," isdpe, pp.101-103, The First International Symposium on Data, Privacy, and E-Commerce (ISDPE 2007), 2007.

[6] "Introduction to data mining and its applications" S. Sumathi, S. N. Sivanandam.

[7] "Introduction to Data Mining with Case Studies", G.K. Gupta.

AUTHOR BIOGRAPHY

Vipul Mangla is pursuing B.Tech (CSE) from VIT University, Tamil Nadu, India.His research areas include Data Mining, Graph Theory and Software Engineering. He has published one more paper in IJETTCS journal in Graph Theory.

Chandni Sardais pursuing B.Tech (CSE) from VIT University, Tamil Nadu, India.Her research areas include Data Mining and Data Warehousing. She is very keen in learning and has done many other research works.

Sarthak Madra is also pursuing B.Tech (CSE) from VIT University, Tamil Nadu, and India. .His research areas include Data Mining and Data Warehousing.