

Mining Frequent Itemset by Reduced Transaction Strategic Algorithm and its Comparative Analysis with Classical Apriori Algorithm

Manisha Bhargava¹, Kapil Mehta²
^{1,2} (CSE, Gian Jyoti Group of Institutions/ PTU, India)

ABSTRACT: Association rule mining is used to find all subsets of items which frequently occur and the relationship between them. Apriori algorithm is used to find out the frequent itemset. But this algorithm is not efficient as it scans the database number of times and takes a lot of time to give the results. In order to solve this problem a new technique is put forward in this paper. The improved algorithm first creates the index of database. It also uses a temporary table technique in the generation of frequent item-set. Also the comparison of new improved algorithm is done with the existing Apriori algorithm. For the simulation the graph is plotted with time variant and data bases. The graph depicts that the improved algorithm shows better performance as compared to the existing algorithm. The improved Apriori algorithm will effectively decrease the system overhead and reduce the running time

Keywords- Data Mining, Association rules, frequent itemset, Apriori Algorithm.

I. INTRODUCTION

Data Mining is the extraction of hidden predictive information from large databases. Data mining [1] is an important tool for analyzing data. It can automatically transform the data into useful information. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified [1]. Association rule mining [2] is a very important tool in data mining [1]. A particular association rule mining algorithm is Apriori algorithm and its generally preferred algorithm. Apriori algorithm [4] scans the database and generates the candidate itemset. It uses minimum support and confidence to calculate the frequent itemset.

II. ALGORITHMS

2.1 Apriori Algorithm

For mining frequent item sets [2] and strong association rules [3] R. Agrawal and R Srikant introduced Apriori algorithm [4] in 1994. Apriori algorithm is, the most classical and important algorithm for mining frequent item sets [1] & [2]. It Assume all data are Categorical. It is used for Market Basket Analysis to find how items purchased by customers are related. Apriori is used to find all frequent item sets in a given database. The key idea of Apriori algorithm is to make multiple passes over the database [5].

It employs an iterative approach known as a breadth-first search through the search space, where k-item sets are used to explore (k+1) item sets[6]. The algorithm terminates when no further successful extensions are found. The algorithm finds all frequent item sets with size 1 in the first pass. In second pass, the algorithm generates a set of candidate item sets of size 2 based on the result of first pass [6]. Then the algorithm scans the dataset and counts the supports and confidence for each generated candidate [5] & [7].

$$\text{Supp}(A \Rightarrow B) = \frac{\text{support containing both A and B}}{\text{total of tuples}}$$
$$\text{Conf}(A \Rightarrow B) = \frac{\text{support containing both A and B}}{\text{tuples containing A}}$$

1.2 Mechanism of Apriori Algorithm

To understand the basic working of Apriori algorithm [7], we take a database of 15 transactions containing an item set $I = \{I1, I2, I3, I4, I5\}$ of six items. We assume the absolute support count of 3

Table1: Database (D)

<u>TID</u>	<u>ITEMS</u>
T1	I1,I3,I5
T2	I1,I4
T3	I4,I5
T4	I2,I3,I5
T5	I1,I2,I3
T6	I2,I4,I5
T7	I2,I5
T8	I2,I3,I4,I5
T9	I4
T10	I2,I3,I4,I5
T11	I3,I4
T12	I1
T13	I2,I4,I5
T14	I4,I5
T15	I1,I2,I3,I4,I5

- In the first step of algorithm we take a candidate set of one item and scan the database to count the support of each member. After scanning we Compare the support count with minimum support count (i.e.3) and write only those items in the item set which has support count greater than or equal to 3and remove the other item. This process shows in Table 2.
- After determining the frequent set of 1 item, we generate the candidate set of 2 items by merging the frequent set of 1 item. Then again we scan the database D to count the support of each element of candidate set and generate the frequent set of 2 items by comparing support count with minimum support count (i.e. 3) and write only those items in the item set which has support count greater than or equal to 3.This process shows in Table 3.
- Further we generate a candidate set of 3 items by using frequent 2 item sets and pruning technique. After that we again scan all the transactions in database D to count the support of each element of candidate set in order to get the frequent set by comparing them with the minimum support count (i.e. 3) and write only those items in the item set which has support count greater than or equal to 3.This process shows in Table4.
- In the next step we generate candidate set of 4 items by using frequent 3 item sets and pruning technique and determine the support of candidate set by scanning all the transactions available in the database in order to get frequent set of 4 items. This process shows in table 5.
- In this way Apriori [7] discover all frequent item set by scanning all the transactions in each repetitive scan and this is the final result.

Table2: Step one

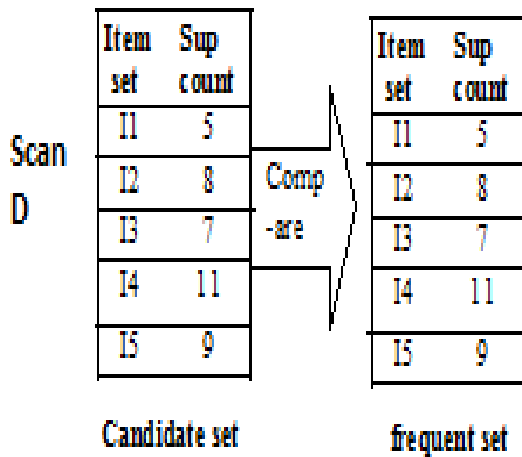


Table3: Step two

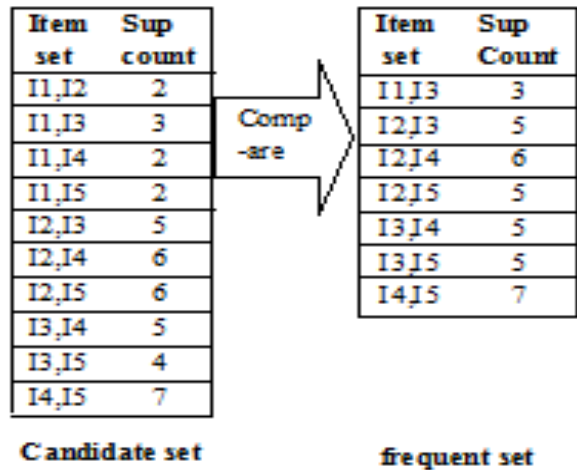


Table4: Step four

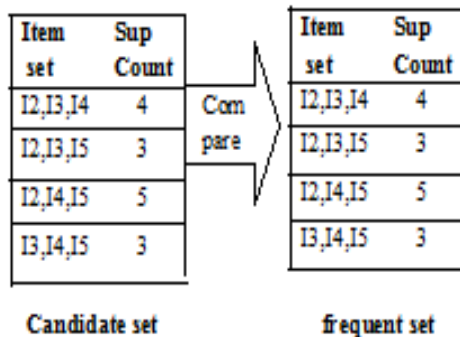
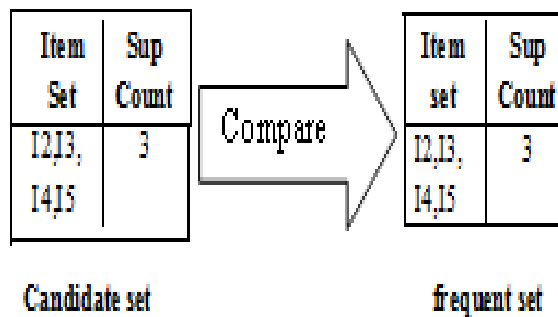


Table5: Step five



III. IMPROVED ALGORITHM:REDUCTION STRATEGIC APPROACH

The improved algorithm takes transaction reduction strategy to overcome the problems of apriori algorithm [4]. Apriori algorithms do not return result in a reasonable time. It only tells the presence and absence of an item in transactional database [4] & [8]. It does not work perfectly when database is very large [8]. The reduction strategic approach is based on apriori algorithm. This proposed algorithm takes transaction reduction strategy to compress the transaction of database in order to reduce the scale of transaction database. This algorithm will effectively decrease the system overhead and reduce the running time. In proposed algorithm there are two step added which are also depicted in the flow chart presented in Fig 1.

- Select the Data base (DB) and then create the index of database (DB).
- Make temporary table and count the number the no of frequent items that are present.

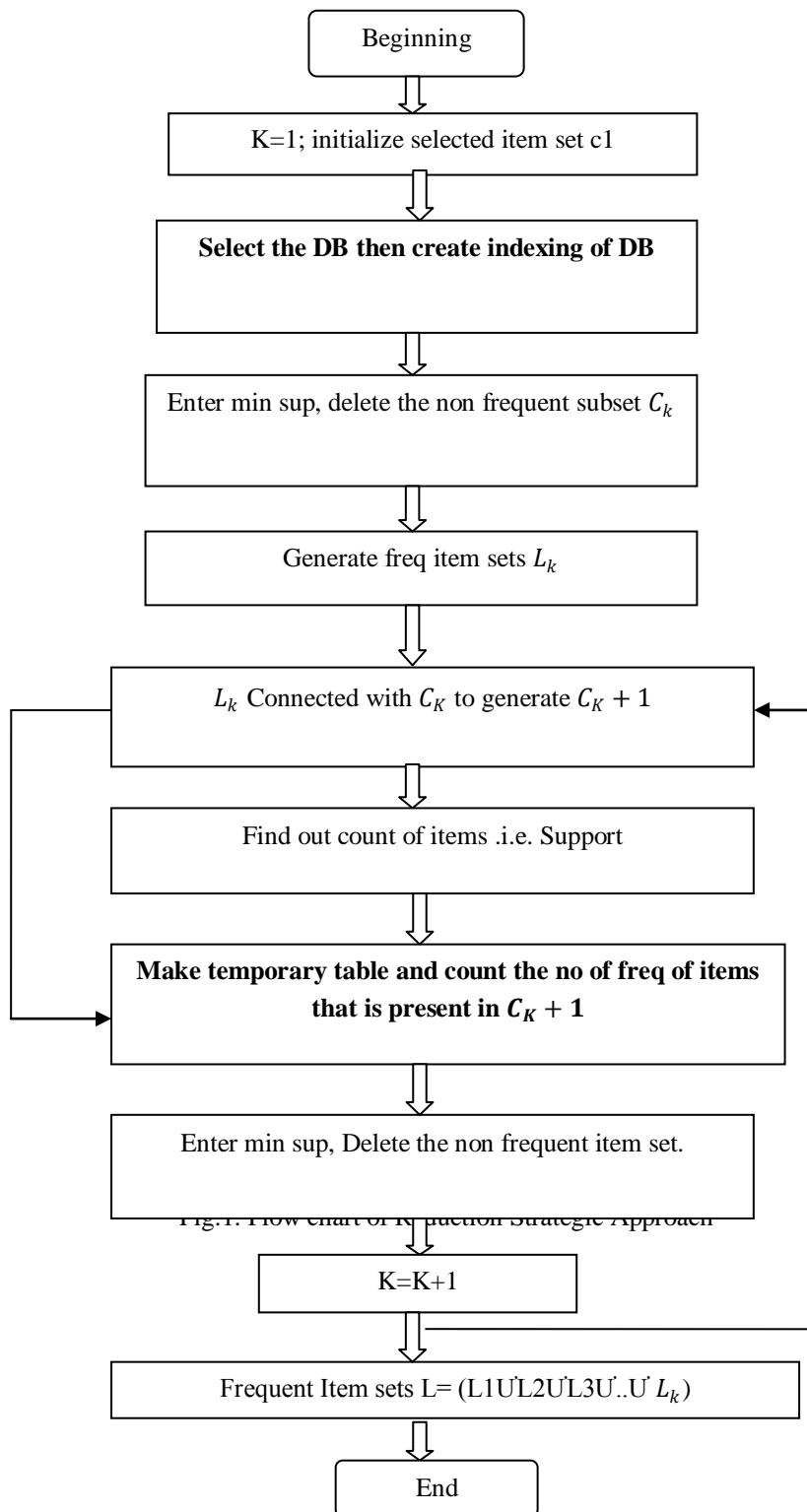


Fig.1. Flow chart of Apriori algorithm

IV. EXPERIMENTAL RESULTS AND CONCLUSION

In order to compare performance difference of Apriori and reduction strategic algorithms in large amounts of data, the experimental test data is the data set randomly generated. To show the experimental results take a database of different sizes. After performing the algorithms on different size of database record the time required to generate the frequent item sets. Then compare the results with the implementation efficiency of the apriori algorithm and the proposed algorithm. The proposed algorithm takes very less time as compared to the apriori algorithm. If we take a database of approximate 1554 kb, apriori algorithm takes 111.835 seconds while proposed algorithm takes very less time i.e. 18.918 seconds. The time taken by different sizes of database is shown in below Table 6.

Table.6. Experimental Results

Data set	Size of Db (in KB)	Time Take (in seconds)	
		Apriori Algorithm	Reduction Strategic Algorithm
Ds1	98	6.981	1.186
Ds2	195	13.897	2.38
Ds3	389	27.863	4.855
Ds4	777	55.875	9.418
Ds5	1554	111.835	18.918

According to the above results, it can be seen that the time taken by reduction strategic algorithm is less than the time taken by classical Apriori algorithm. Minimum support for performing the results of these two algorithms is set to 50 % as presented in Fig 2.

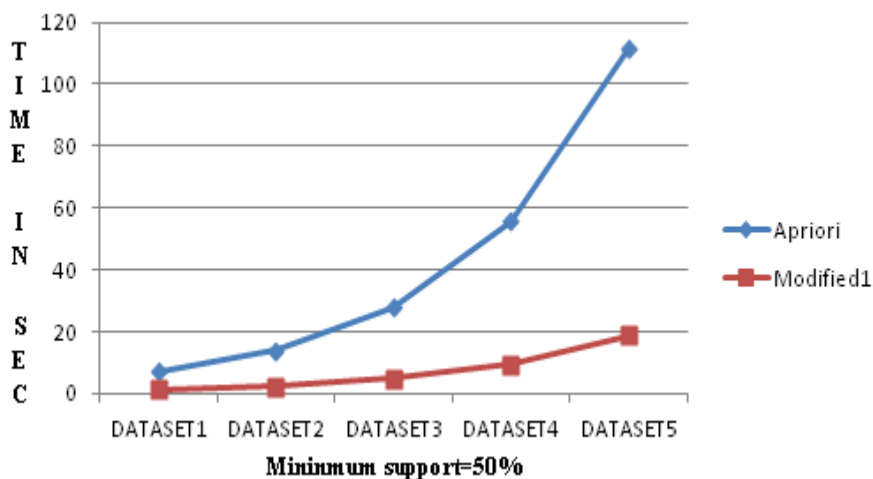


Fig.2. Comparison of performance in terms of seconds

For the future research and development, this algorithm can be applied on other association rule mining algorithms like FP- Growth, Eclat. This work can also be extended by sorting dataset into an order and then apply indexing and hashing technique.

REFERENCES

- [1]. Jiawei Han, Micheline Kambar, Jian Pei, data mining concepts and techniques(Morgan Kaufman, 2000)
- [2]. Agrawal, R., Imielinski, T., and Swami, A. N., Mining association rules between sets of items in large databases, International Conference on Management of Data, May 1993. pp. 207-216.
- [3]. Agrawal, R., Srikant, R., Fast Algorithms for mining association rules in large databases, International Conference on Very Large Databases, Santiago de Chile, 1994. pp 487-489.
- [4]. Mamta Dhanda, Sonali Guglani and Gaurav Gupta, Mining Efficient Association Rules through Apriori Algorithm Using Attributes IJCST Vol. 2, Issue 3, 2011, pp. 2229 – 4333.

- [5]. M Suman, T Anuradha, K Gowtham, A Ramakrishna, A Frequent Pattern Mining Algorithm Based On FP-Tree Structure And Apriori Algorithm, International Journal of Engineering Research and Applications (IJERA) ISSN, Vol. 2, Issue 1 2012, pp.114-116.
- [6]. Manisha Bhargava ,Arvind Selwal, Association Rule mining using Apriori Algorithm: A Review, International Journal of Advanced Research in Computer Science(IJARCS), 2014 vol4, Issue no2.
- [7]. Yubo Jia, Guanghu Xia, Hongdan Fan, Qian Zhang, Xu Li, An Improved Apriori Algorithm Based on Association Analysis, IEEE- Third International Conference on Networking and Distributed Computing, 2012 pp.208-211.
- [8]. Rui Chang, Zhiyi Liu, *An Improved Apriori Algorithm*, IEEE- International Conference on Electronics and Optoelectronics, 2011, pp.476-478.
- [9]. Huiying Wang, Xiangwei Liu, *The Research of Improved Association Rules Mining Apriori Algorithm*, IEEE- Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),2011,pp.961-964.
- [10]. Shuo Yang, *Research and Application of Improved Apriori Algorithm to Electronic Commerce* ,IEEE- 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science,2012,pp.227-231