© 2017 IJSRSET | Volume 3 | Issue 3 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Improved Apriori Algorithm with Pruning Unnecessary Candidate Set for Reducing Execution Time

Prof. Neha Khare, Priya Shrivastava
Takshshila Institute of Engineering and Technology, Jabalpur, Madhya Pradesh, India

ABSTRACT

Data mining which is also known as Knowledge Discovery in the databases (KDD) is an important research area in today's time. One of the important techniques in data mining is frequent pattern discovery. Finding co-occurrence relationships between items is the focus of this technique. The active research topic for KDD is association rule mining and many algorithms have been developed on this. This algorithm is used for finding associations in the item-sets. Its application areas include medicine, World Wide Web, telecommunication and many more. Efficiency has been an issue of concern for many years in mining association rules. Till date the researchers of data mining have worked a lot on improving the quality of association rule mining and have succeeded to a great extent. There are many algorithms for mining association rules. Apriori algorithm is the mostly used algorithm which is used to determine the item-sets, which are frequent, from a large database. It extracts the association rules which in turn are used for knowledge discovery. Apriori is based on the approach of finding useful patterns from various datasets. There are lot many other algorithms that are used from association rule mining and are based on Apriori algorithm. Although it is a traditional approach, it still has many shortcomings. It suffers from the deficiency of unnecessary scans of the database while looking for frequent item-sets as there is frequent generation of candidate item-sets that are not required. Also there are sub item-sets generated which are redundant and algorithm involves repetitive searching in the database. This work has been done to reduce the redundant generation of sets. The large dataset is scanned only once. As a result, the overall time of execution is reduced. Also the number of transactions to be scanned are reduced.

Keywords: Apriori, Datamining, Frequent Item, Association Rules, Transactions

I. INTRODUCTION

1.1.1 Knowledge Discovery in the Database (KDD)

The role of data mining (KDD) is very important in many of the fields such as analysis of market basket, classification, etc. If talk about data mining, the most important role presented by frequent item set which is used to find out the correlation between the various types of the field that is displayed in the database. Discovery of frequent item set is done by association rules. Retail store also use the concept of association rule for managing marketing, advertising, and errors that are presented in the telecommunication network.

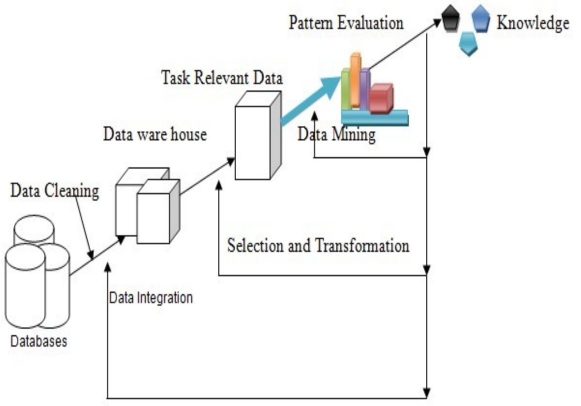


Figure 1.1 : The Process of Knowledge Data Discovery

As we know information technology is growing and databases generated by the companies or organizations like telecommunications, banking, marketing, transportation, manufacturing etc are becoming huge. It is important to explore the databases and efficiently and completely as data mining helps to identify information in large amount of databases. KDD is the process designed to generate data that shows the well-defined relationship between the variables. It is the process designed to generate data that show the well-defined relationship between the variables. KDD has been very interesting topic for the researchers as it leads to automatic discovery of useful patterns from the database. This is also called as Knowledge Discovery from the large amount of database. Many techniques have been developed in data mining amongst which primarily Association rule mining is very important which results in association rules. These rules are applied on market based, banking based etc. for decision making.

The relationship among the items is done by association rule. All type of relationship between items is totally based on the co-occurrence of item.

The knowledge discovery in data can be achieved by following steps:

- *Data Cleaning*: In this step, the data that is irrelevant and if noise is present in database then both irrelevant and noisy data is removed from the database.
- *Data Integration*: In this step the different types of data and multiple data sources are joined in a common source.
- *Data Selection*: In this stage, the application analyzes that what data, what type of data is retrieved from the collection of data.
- *Data Transformation*: In this stage, the selected data is changed into accurate form for the procedure of data mining.
- *Data Mining*: This is the important step in which the techniques used to extract the pattern is clever.
- *Pattern Evaluation*: In this step severely needed patterns represent acquaintances based on measures parameters.
- *Knowledge Representation*: This is the final step in which knowledge is visually represented to the user. Knowledge representation use visualization techniques to help in understanding of user and taking the output of the KDD**.**

II. REALTED WORK

In [3], they proposed high dimension Apriori algorithm. Unnecessary generated sub item-sets are removed by this method. As a result higher efficiency of mining can be obtained when data dimension is high as compared to original Apriori algorithm.

In [4], a new algorithm is proposed that reduces number of times the database is scanned. This algorithm is Apriori algorithm. Also the procedure of joining frequent item-sets is optimized. This results in the reduction in the size of candidate item-set. This newly proposed algorithm performs better than the classical Apriori algorithm.

An Apriori algorithm is can also be improved with improvement of pruning operation. In [8], the same approach is adopted with the introduction of count based method in an improved algorithm named IAA. According to this if L is an item-set of dimension *k* and is a frequent set then its subsets of dimension *k-1* will also be frequent. Since each of the two subsets will generate L only once, the total time will be Lk2. If the count of each subset of Lk is less than Lk2 then it is considered to be infrequent. Other deficiency of Apriori algorithm is scanning the database multiple times. This is also improved in IAA. In this data is stored as <Item set, TID> [3] which is similar to WDPA with only difference is that each candidate item-set is counted only once. Thus candidate sets are generated using count occurrence step that is based on records that were produced in prune operation. The frequent item-sets and association rules are generated synchronously the advantage of synchronized generation is that operations can be stopped before all large frequent item-sets are found in case the existing results are not according the expectation. This deviation may arise if minimum support is inappropriate or there is restriction on the number of association rules.

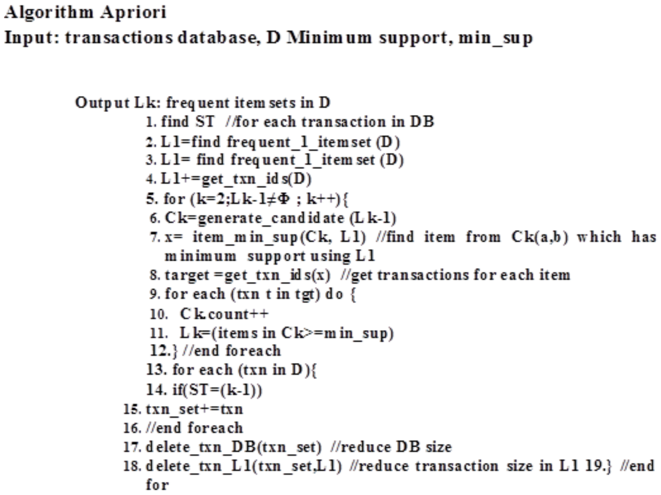
In [17], an enhanced algorithm is designed and discussed which puts forward only those items that the user is interested in. These items are called seed items. The database is then scanned and all the items which are in the same transaction with the seed item are added as a part of item-set. The count structure is used to keep a record of the state of items so that item is not repeatedly visited every time database is scanned. If count structure is updated after every scan of the database, then keep scanning otherwise scanning is stopped and item-set of user interest are recorded. The support value of each item is calculated taking importance of item-sets into consideration. The importance is measured by using weight as an indicator which is calculated using price of the item as a parameter. The algorithm is efficient as the data is compressed which increases the speed of the operation and the computational efficiency. The generation of frequent item-sets is faster and memory is also saved.

In [18], an Apriori algorithm is discussed that consist of three areas of improvement. Firstly the reduction in number of judgements, secondly: reduction in the number of candidate frequent item-sets and lastly the database optimization. It is assumed that the item-sets are ordered. Thus if two item-sets cannot be connected then all the item- sets after these two items cannot satisfy the condition to form a connection. Thus judgements are reduced. Secondly all the item-sets with frequency less than k-1are found and saved in I, which is the set of items, and then those frequent item-set that contain subset of I are removed. Lastly database optimization is achieved by maintain a delete tag for all those transactions which do not contain Ck. These items are not considered further as the algorithm proceeds.

Sometimes it may occur that a transaction in the *k+1* pass does not contain frequent *k* item-set. Identifications of such transactions are necessary so that these transactions are no read and processor time is saved. Proposed approach in [7] identifies the transactions that contain the frequent set and checks whether that transaction should be scanned further or not. This is done in first scan. In order to achieve the efficiency, the data is distributed among parallel processors. This distribution is equal. Each processor is utilized to the maximum in order to maximize the efficiency.

III. PROPOSED WORK AND RESULTS

Proposed Algorithm:

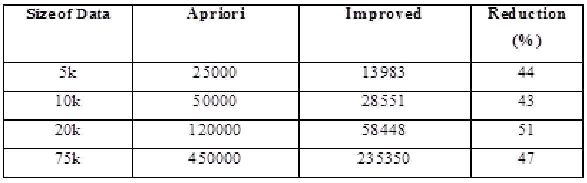


```
Algorithm Apriori
Input: transactions database, D Minimum support, min_sup

    Output Lk: frequent item sets in D
        1. find ST  //for each transaction in DB
        2. L1=find frequent_1_itemset (D)
        3. L1= find frequent_1_item set (D)
        4. L1+=get_txn_ids(D)
        5. for (k=2;Lk-1≠Φ ; k++){
         6. Ck=generate_candidate (Lk-1)
         7. x= item_min_sup(Ck, L1) //find item from Ck(a,b) which has
            minimum support using L1
         8. target =get_txn_ids(x) //get transactions for each item
         9. for each (txn t in tgt) do {
         10.  Ck.count++
         11.  Lk=(items in Ck>=min_sup)
         12.} //end foreach
        13. for each (txn in D){
        14. if(ST=(k-1))
      15. txn_set+=txn
      16. //end foreach
      17. delete_txn_DB(txn_set) //reduce DB size
      18. delete_txn_L1(txn_set,L1) //reduce transaction size in L1 19.} //end
          for
```

(a) Reduction in the Number of Transactions

The number of transactions to be scanned is reduced by taking intersection of the transactions as already discussed. This intersection gives the support value. The comparison of the improved algorithm is done with the Apriori algorithm in terms for number of transactions. In Apriori in every pass the original database is scanned therefore the total transactions to be scanned will be equal to the product of the total number of transactions and number of passes whereas in improved Apriori transactions are reduced at every pass. Only in the first pass, it is equal to the transactions in the original dataset. In the subsequent passes it reduces. Both the algorithms are tested on datasets of different sizes and results are depicted in the table and graphically represented in figure below:



Size of Data	Apriori	Improved	Reduction (%)
5k	25000	13983	44
10k	50000	28551	43
20k	120000	58448	51
75k	450000	235350	47

Table 3.1: Comparison of Number of Transactions

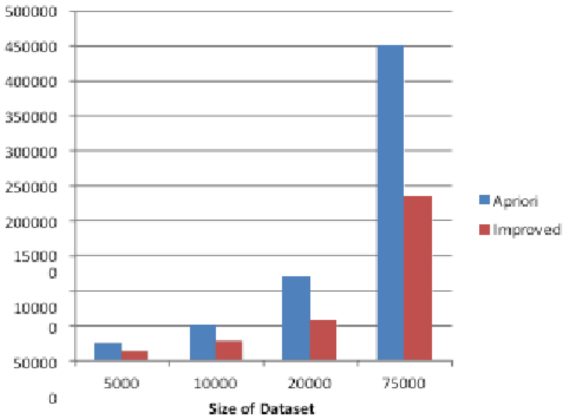


Figure 3.1 : Comparison of Number of Transactions

IV. CONCLUSION

After implementing the proposed approach, we come to the conclusion that the improved Apriori algorithm proposed is an effective algorithm to reduce the number of transactions. The work is carried out on transactions rather than items which have improved its efficiency and to achieve the same the dataset is taken in a transposed manner. Instead of repeated scan of the original database, it is scanned only once to form large 1 item-set from which further computations are carried out. This reduces the time involved in scanning the dataset which in turn reduces the overall time to a greater extent. The minimum support value is also calculated at each pass which removes the unnecessary formed sets. Although the algorithm is simple, it carries out more effective pruning.

The results of the improved Apriori algorithm are satisfactory on single processor. The future work may include the implementation of the improved Apriori algorithm on the data distributed on parallel processors. The results are expected to be different in that case. We also wish to see whether the database scans reduce on each processor using this approach.

V. REFERENCES

[1]. R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules", pp. 487- 499.

[2]. X. Liu, P. He, "The Research of Improved Association Rules Mining Apriori ALgorithm", Proceedings of the Third International Conference on Machine Learning and Cybermetics, Shanghai, 26-29 August 2015, pp. 1577-1579.

[3]. J. Lei, B. Zhang, J. Li, "A new improvement on Apriori Algorithm", International Conference on Computational Intelligence and Security, Vol. 1, IEEE, 2015, pp. 840-844.

[4]. Y. Xie, Y. Li, C. Wang, M. Lu, "The Optimization and Improvement of the Apriori Algorithm", Education Technology and Training, International Workshop on Geoscience and Remote Sensing, ETT and GRS, Vol. 2, IEEE, 2015, pp. 663- 665.

[5]. Z. Changsheng, L. Zhongyue, Z. Dongsong, "An Improved Algorithm for Apriori", First International Workshop on Education Technology and Computer Science, 2015, pp. 995-998.

[6]. L. Jing et. al, "An Improved Apriori Algorithm for Early Warning of Equipment Failure", 2015, pp. 450-452.

[7]. K. Shah, S. Mahajan, "Maximizing the Efficiency of Parallel Apriori Algorithm", International Conference on Advances in Recent Technologies in Communication and Computing, 2015, pp. 107-109.

[8]. H. Wu, Z. Lu, L. Pan, R. Xu, W. Jiang, "An Improved Apriori-based Algorithm for Association Rules Mining", Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2014, pp. 51-55.

[9]. Y. Liu, "Study on Application of Apriori Algorithm in Data Mining", Second International Conference on Computer Modelling and Simulation, 2013, pp. 111-114.

[10]. L. Wu, K. Gong, Y. He, X. Ge, J. Cui, "A Study of Improving Apriori Algorithm", 2013, pp. 1-4.

[11]. P. Sandhu, D. Dhaliwal, S. Panda, A. Bisht, "An Improvement in Apriori algorithm Using Profit And Quantity", Second International Conference on Computer and Network Technology, 2012, pp. 3-7.

[12]. L. Lu, P. Lu, "Study On An Improved Apriori Algorithm And Its Application In Supermarket", the research on Uncertain Reasoning Mechanism of Fuzzy Concept Map, pp. 441-443.

[13]. G. Wang, X. Yu, D. Peng, Y. Cui, Q. Li, "Research of Data Mining Based on Apriori algorithm in Cutting Database", 2012, pp. 3765-3768.

[14]. V. Sharma, M. Beg, "A Probabilistic Approach to Apriori Algorithm", International Conference on Granular Computing, IEEE, 2013, pp. 225-243.

[15]. Y. Shi, Y. Zhou, "An Improved Apriori Algorithm", International Conference on Granular Computing, IEEE, 2013 pp. 759-762.

[16]. Y. Shaoqian, "A kind of improved algorithm for weighted Apriori and application to Data Mining", The 5th International Conference on Computer Science & Education Hefei, China, August 24–27, 2013, pp. 507-510.

[17]. D. Ping, G. Yongping, "A New Improvement of Apriori Algorithm for Mining Association Rules", International Conference on Computer Application and System Modelling (ICCASM), 2013, pp. V2-529.

[18]. Y. Zhou, W. Wan, J. Liu, L. Cai, "Mining Association Rules Based on an Improved Apriori Algorithm", 2012, pp. 414-418..