International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 05 | May-2016 www.irjet.net p-ISSN: 2395-0072

Reducing Redundant dataset in Association based Technique

Chetanay Gupta, Prateek Dubey, Dharmveer Singh Rajput

Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India

Abstract: *Frequent item set generation is the basic need for data mining to see the frequently occurring data set. Data mining or Data analysis is based on frequent item set generation based on minimum support principle. Mining can be done by various algorithms but the most important thing is choosing the most efficient algorithm. Since the Apriori algorithm proposed to generate frequent item set there have been several researches to improve its efficiency which has improved but can be improved further by using some logistic approach. The main problem with this algorithm is time constraint due to multiple pass to database. The K-Apriori algorithm as proposed in this paper tries to reduce time by reducing the number of passes.*

Key Words: Kapriori, Classification, Clustering

Overview:

Mining is a process of extracting useful or interesting information from various information sources such as xml files, database, data warehouse etc. There are basically two classes' broad classes of data mining: Descriptive and Prescriptive. Descriptive mining is summarizing general properties of data in data repository and on the other hand prescriptive mining performs interference on current data by making prediction based on analysis of the old data [2]. There are three types of data mining techniques: Association, Classification and Clustering [8]. Our paper is based on concepts of association in data mining.

Association Rule

This rule is mainly used to make relevant links between items in data set. It's an important branch of data mining. Let us assume a expression of the form: **z->v**

Where a z and v are item sets. There is a rule evaluation metrics in which we calculate support and confidence. Support refers to "Ratio of transactions that contain both z as well as v " and confidence refers to "probability of occurrence of v in transactions that contains z ".Association Rule Mining is the study of finding rules whose support as well as confidence are more than the limit, min support and min confidence values. It proceeds further in two steps:

The first step is to find all item sets with adequate supports and the second step is to generate association rules by combining these frequent or large item-sets. Applications

Applications

Market Basket Analysis: Market basket databases consist of a large number of records and in each record all items bought by a customer on a single purchase transaction are listed. Managers would be paying attention to know that which groups of items are constantly purchased together. This data is used to analyze which items are to kept together, which items should come under discounts, which item has maximum sale along with its subordinate item. For example, a shopkeeper observes that customer buying bread is always buying milk. Association rule mining is of high importance for commercial use as they try to analyze their sales and see if specific type of relation between items to detect the relationship between items and to keep stock of items according to it. This "market basket analysis" result can be used to suggest combinations of products for special discounts or sales .Market basket can be defined as collection of items purchased by a customer (e.g. supermarket, web).

Medical diagnosis: Association rules can be used in medical analysis for assisting physicians to cure patients. The common problem with indicating relationship between diseases is there is no theoretical proof that a disease will occur with other diseases for sure. There is no surety of the disease occurring together. It only helps to identify the probability of illness in a certain disease. This intersection can help in adding some new symptoms of a disease and also defines new relations between these symptoms. Certain symptoms are always associated with specific diseases.

CRM of credit card business: Customer relationship management through the banks tries to improve their services to customers. They divide their customers in different groups by analyzing their services, their liking, to enhance the cohesion between customer and the bank. Song provided with a method to analyze changing

customer at different time taking snaps form customer profiles. The main idea is analyzing changing dataset and makes changes in rule according to each dataset.

Algorithm

Apriori Algorithm: Apriori algorithm is used for generating frequent data set and applying association rule to find relation between the given data. It uses "bottom up" approach. Apriori algorithm works in a very easy manner it first collects all the transactions from the dataset one by one after doing this it finds unique entries present in it and calculate support and confidence as mentioned above for each unique item. Now after calculating support and confidence if finds items which have support and confidence greater than minimum support. The items qualifying this forms pair and then support of these pairs is calculated this process continues till single transaction is left or no further pairing can be done. The complexity of this algorithm is very high as it involves scanning of database at various stages, therefore it is very expensive.

How Apriori Works:

- Find all the unique item having support greater than minimum support
- Generate paired sets from frequently occurring dataset.
- Prune the results to find the frequent item sets.
- Generate association rule between item from frequent item sets
- Rules which satisfy the Min_support and min_confidence threshold.

Problems:

- Apriori is Snell like moving algorithm and bottleneck in frequent item set generation.
- In case of huge amount of data, use of Apriori can be a disaster.
- Apriori algorithm requires large amount of scans of dataset again and again.
- All transaction is made for each candidate.

FP Tree Algorithm

It uses shrieked representation of database in form of FP-growth tree. Once the tree is constructed it recursive makes frequent item set by using greedy approach.

How the algorithm works: Starting of FP-tree is same as that of Apriori it first calculates support for the unique items in dataset after that it the item qualifying the min support criteria are taken and constructed based on their occurrence in a particular transaction. After achieving this much start forming conditional pattern and conditional FP-tree pattern and calculating frequent item set for each item from the leaf node of tree.

In large databases, due to more consumption of memory it's not feasible to implement FP-tree. To solve this problem we first divide our database in smaller databases and construct FP-tree for each one of them.

Problems:

1. There is too much database scanning to calculate frequent item (reduce performance)

2. Assumes that there is no memory shortage of database thus wastage of memory.

3. Condition pattern and condition FP tree generation is time consuming

4. Support counting is expensive

• Subset checking (computationally expensive)

• Multiple Database scans (I/O)

Problem Statement: Removal of repeated dataset traversal thus reducing scanning time.

Authors [7], research use soft set theory along with the traditional used Apriori algorithm. Soft set is a way for mining of data by association rule from transactional dataset. This method converts all the transactional data to a Boolean value in the system. There is no limit of parameterization tool and they can deal with Boolean valued-information system.

Authors [6], proposed an Apriori algorithm based on the matrix, uses matrix to describe the each affair in the database, so we just deal with the matrix to count the support of the candidates of frequent item sets, and there is no further need to scan the database. Here, count the support quickly using the "AND operation". The "AND

operation" is to AND the rows according to the items in the candidate of frequent item sets in the matrix, then add the result of the AND, and the result is the support. For example, we just AND the row "a" and row "c" in the matrix when we want to count the support of item sets {a, c}, and add the result, so we can get the support of it. Advantages: The algorithm based on the matrix is better than the Apriori in two aspects, both time and space. It needn't scan the database time and again to lookup the affairs, and also greatly reduce the number of candidates of frequent item sets. The algorithm based on the matrix is better than the Apriori both in time and space. It needn't scan the database time and again to lookup the affairs, and also greatly reduce the number of candidates of frequent item sets. This algorithm is good at finding the frequent item sets which support is small.

Author's [5], presented a new method for generating frequent item sets is introduced in this section, which is based on the existing mining algorithms. This will generate item sets progressively hence named as PAPRIORI algorithm. If minimum support set to around 50% execution time is less with comparison to Apriori algorithm. Proposed mining algorithm for incremental generation of large item sets. The no transactions K should be intermediate. It performs unsatisfactory if value of k becomes very high as the time of running this algorithm will increase and along with it if the value is small it will perform similar to APRIORI algorithm.

Authors [3], focused on enhanced version of Apriori Algorithm. It is focused on four process: for choosing the appropriate data it forms the data , then, it couples up the item to form a relation for distingusment, third is mining of the data in new database and ,fourth involves forming associative rule for mining the data and forming new relations for analysis . In this, mapping-table was to convert the items into compressed data set. To count the support, HASH_TREE had been used, which disperses the matching of candidate item-sets to reduce the time complexity of algorithm. This approach used HASHMAPPING TABLE and HASH_TREE, to optimized both space and time complexity. Time complexity was reduced to some extent. More memory utilized. It works on guess so it's good to cross check that the it's right.

PROPOSED METHOD:

This algorithm, also uses the Apriori-gen function (used to generate candidate set during first pass)to determine the candidate item sets before the pass begins(Fk). The interesting feature of this algorithm is that the database B is not used for counting support after the rest of the pass. Rather, the set Fk is used for this purpose. Each member of the set Fk is of the form < TD; {Zk} >, where each Zk is a potentially large k-item set present in the transaction with identier TD. For k = 1, F1 corresponds to the database B, although conceptually each item is replaced by the item set . For k > 1, Fk is generated by the candidate set generation. If a transaction does not contain any candidate k-item set, then Fk will not have an entry for this transaction. Thus, the number of entries in Fk may be smaller than the number of transactions in the database, especially for large values of k

```
Ek = flag 1-setsg;
 F1 = database B;
 for ( k = 2; Ek-1 =! NULL ; k++ ) do begin
 Fk = generation(Ek-1 ); // New candidates
 Fk = NULL;
 all entries at 2 Fk-1 do begin
 #  candidate itemsets in Fk contained
#in the transaction with identifier t.TD
Ft = t belongtoFk | (t - t[k]) 2 t:set-of-itemsets  intersects
(t- t[k-1]) 2 t.set-of-itemsets };
 For all candidates t belonging to Ft do
 t:count++;
 if (Ft =! NULL ;) then Fk += < t:TD;Ct >;
 end
 Ek = {t belonging to  Ek | t:count >= minsup }
 end
 Answer = Ek;
```

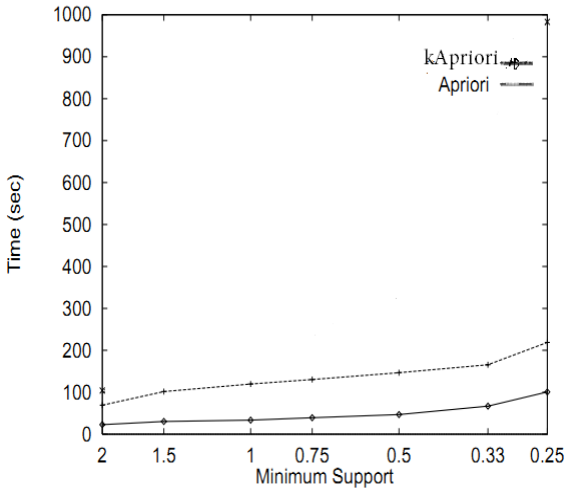
EXPERIMENTAL ANALYSIS:

For the comparative study of Apriori and Kapriori algorithm we have taken a database of 100 transaction of 10 items. In the process we considered 200 transactions to generate the fp with the support count 0.1 .This process was performed again and again by increasing data set ,the output was analyzed on proposed and basic algorithm summarized a result in the following table .Looking to the increasing rate of the Apriori we see it increases linearly with increasing transactions where as in the Kapriori (after a time period the consistency of the time was maintained) take less time as no of passes increases. This concludes our experiment on saying that if 1000 transactions are taken time by Apriori is 1000 whereas of proposed algo is 250 saving great amount of time[4].

Table1: Comparative Results

TRANSACTIONS	APRIORI(s)	KAPRIORI(s)
10	7	6
30	9	7
50	14	10
70	20	15
90	24	18
100	28	20

Kapriori is effective in later passes as it has to pass only the data stored in memory. If memory overhead is not there Kapriori performance is far better as it reduce passing data set from database again and again. The data stored in memory becomes smaller than large dataset thus reducing time. Below we have constructed a graph changing the min support of dataset and then checking execution time on dataset set 1000 transaction and 10 items each.



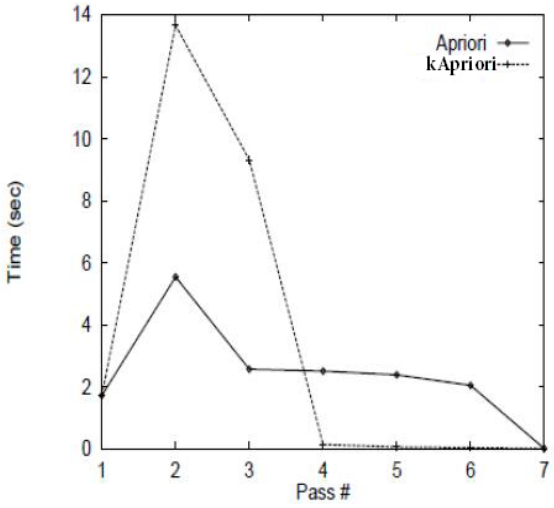


Figure 1: Graphical Result of Comparison

QUALITY MEASURES

False Rate: Sometimes during accessing data some misinterpretation occurs which results in some association rule generation which does not even qualify the minimum confidence requirement these relations are removed using false rate quality measure.

CONCLUSION

A lot of researches have been made trying to decrease time in Apriori but with minimum output. The researches were helpful in building this algorithm .Like other developed algorithm this to has some drawbacks which we would emphasis upon in our next research paper and try to solve to .As of now we have proved above how the Kapriori take less time than that of classical algorithms and how this is really going to be fruitful in saving the time in case of large database.

REFERENCES:

[1] Miss. Nutan Dhange et.al,"Scalable and Efficient Improved Apriori Algorithm", International Journal of Innovative Research in Computer and Communication Engineering, vol.1, Issue 2, April 2013.

[2] D. S. Rajput, P. K. Singh and M. Bhattacharya, "*PROFIT: A Projected Clustering Technique*", "Real World Data Mining

Applications", "Annals of Information Systems 17, Springer", pp 51-70, 2015.

[3] Apurwa Sahu, Mradul Dhakar, Pushpi Rani,2015, Comparative Analysis of Apriori Algorithm based on Association Rule, International Journal of computer science,2015.

[4] K.Geetha et.al,"An Efficient Data Mining Technique for Generating Frequent Item sets", International Journal of Advanced Research in Computer Science and Software Engineering,vol.3,Issue 4,April 2013.

[5] Sujatha kamepalli et.al,"Apriori Based: Mining Infrequent and Non-Present Itemsetsfrom Transactional Data Bases", IJECS-IJENS, vol.14,no.03, June 2014.

[6] Komal Khurana1, Mrs. Simple Sharma2,A Comparative Analysis of Association Rules Mining Algorithum,International Journal of Scientific and Research Publications, Volume 3, Issue 5, May 2013

[7] Jeetesh Kumar Jain, Nirupama Tiwari, Manoj Ramaiya / International Journal of Engineering
Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 1, January -February 2013, pp.2065-2069.

[8] D. S. Rajput, P. K. Singh and M. Bhattacharya, "IQRAM: A High Dimensional Data Clustering Technique", International Journal of Knowledge Engineering and Data Mining, Inderscience, Volume 2, Issue 2/3, pp. 117-136, 2012.

[9] T. Karthikeyan1 and N. Ravikumar2,A Survey on Association Rule Mining, International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 1, January 2014.

[10] Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New Algorithms for Fast Discovery of Association Rules.