International Journal of Computer Applications Technology and Research
Volume 4– Issue 5, 371 - 373, 2015, ISSN:- 2319–8656

Data Mining using Improved Association Rule

Arpit Tripathi
Thakur College of Engineering and Technology
Mumbai, India

Shefali Singh
Thakur College of Engineering and Technology
Mumbai,India

Devika Prasad
Thakur College of Engineering and Technology
Mumbai,India

Abstract: Data Mining plays an important role in extracting patterns and other information from data. The Apriori Algorithm has been the most popular techniques infinding frequent patterns. However, Apriori Algorithm scans the database many times leading to large I/O. This paper is proposed to overcome the limitaions of Apriori Algorithm while improving the overall speed of execution for all variations in 'minimum support'. It is aimed to reduce the number of scans required to find frequent patters.
Keywords: Aprior, association, candidate sets, data mining

1. INTRODUCTION

Data Mining has become a great field of interest in this era of online shopping and web malls. Although most data mining systems work with data stored in flat files, it is beneficial to implement data mining algorithms using SQL in DBMS that allow us to discover patterns in data. Association rules have been used to find realtionships between itemsets in large datasets. In this paper we discuss a method to find frequent itemsets in datasets faster than traditional algorithms, per se Apriori Algorithm. This algorithm reduces the number of scans done to find frequent patterns in large datasets. Apriori Algorithm creates large candidate itemsets for smaller 'minimum supports'. The main goal of the system is to reduce the execution time for finding frequent patterns.

2. RELATED WORK

Several attempts were made by researchers to improve the efficiency:

1. Krishna Balan, Karthiga, Sakthi Priya suggested using Hash table and finding frequent itemsets in dataset. They proposed a algorithm that does a three stage process where the first process is a hash based step is used to reduce the candidate itemsets generated in the first phase, create a 2-itemset combination of itemsets in a transaction and include it in Hashtable. Finally, removing the itemsets with support less than minimum support.[0]

2. Mahesh Balaji and G Subrahmanya VRK Rao et al in their paper for IEEE proposed Adaptive Implementation Of Apriori Algorithm for Retail Scenario in Cloud Environment which solves the time consuming problem for retail transactional databases. It aims to reduce the response time significantly by using the approach of mining the frequent itemsets.

3. ALGORITHM

4. Apriori Algorithm

R. Agrawal and R. Srikant in 1994 presented the apriori algorithm for mining frequent itemsets which is based on the generation of candidate itemset.One of the first algorithms to evolve for frequent itemset and Association rule mining was Apriori. Two major steps of the Apriori algorithm are the join and prune steps. The join step is used to construct new candidate sets. A candidate itemset is basically an item set that could be either Frequent or infrequent with respect to the support threshold. Higher level candidate itemsets (Ci) are generated by joining previous level frequent itemsets are Li-1 with it. The prune step helps in filtering out candidate item-sets whose subsets (prior level) are not frequent. This is based on the anti-monotonic property as a result of which every subset of a frequent item set is also frequent.Thus a candidate item set which is composed of one or more infrequent item sets of a prior level is filtered(pruned) from the process of frequent itemset and association mining.

Algorithm: The Apriori Algorithm

Input:

T//Transaction Dataset

m //Minimum support

Output:

Frequent Itemsets

Steps:

1.Ck: Candidate itemset of size k

2.Lk : frequent itemset of size k

3.L1 = {frequent items};

4.for (k = 1; Lk != □; k++) do begin

Ck+1 = candidates generated from Lk;

5.for each transaction t in database do

increment the count of all candidates in Ck+1 that are contained in t

Lk+1 = candidates in Ck+1 with min_support

6.end

7.return □k Lk

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis. 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. Each transaction is seen as a set of items (an itemset). Given a threshold , the

Apriori algorithm identifies the item sets which are subsets of at least transactions in the database.

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the source dataset.

5. Improved Algorithm

In this algorithm, the first step is finding the support of all itemset is same as Apriori algorithm. Any items that have support less than minimum support is less are discarded. For next step, 2-itemsets combination for items in each transaction is created. Count for 2-itemset is found and itemsets with count less than minimum support are deleted. The database is reduced only using these distinct itemsets, this is called 'Transaction Reduction'. Support for all items is found and frrequent itemset are found.

Algorithm:

1 Take inputs from user for minimum support.

2. Find count for all itemsets in the database.

3. Delete itemsets from Databse having support less than minimum support .

4. Create all possible 2-itemset candidate itemset for each transaction.

5. Modify the transaction database to include only these candidate pairs

6. Then the candidate itemsets which has less frequent are then removed from the transaction database.

7. The database is scanned for minimum support threshold, frequent items are selected and sorted..

6. Example.

TID	Items
T1	I1,I3,I7
T2	I2,I3,I7
T3	I2,I3,I1
T4	I2,I3
T5	I2,I3,I4,I5
T6	I2,I3
T7	I1,I2,I3,I4,I6
T8	I2,I3,I4,I6
T9	I1
T10	I1,I3

Reducing the Database:

TID	Items
T1	I1,I3
T2	I2,I3
T3	I2,I3,I1
T4	I2,I3
T5	I2,I3,I4
T6	I2,I3
T7	I1,I2,I3,I4
T8	I2,I3,I4
T9	I1
T10	I1,I3

Hash Table-

TID	Items
T1	I1I3
T2	I2I3
T3	I1I2,I2I3,I1I3
T4	I2I3
T5	I2I3,I4I3,I2I4
T6	I2I3
T7	I1I2,I2I3,I3I4,I1I3,I2I4,I1I4
T8	I2I3,I3I4,I2I4
T9	I1
T10	I1I3

HASH COUNT:
{I1I3}=4,{I2I3}=7, {I1I2}=2,{I1I3}=3,
{I2I4}=3,{I3I4}=3,,{I1I4}=1

Reducing the Database:

TID	Items
T1	I1,I3
T2	I2,I3
T3	I2,I3,I1
T4	I2,I3
T5	I2,I3,I4
T6	I2,I3
T7	I1,I2,I3,I4
T8	I2,I3,I4
T9	I1
T10	I1,I3

Item Count-

Items	Count
I1	**5**
I2	**7**
I3	**8**
I4	**3**

7. Experimental Results

The data sets given to the Apriori and the improved algorithm are same. The results of the experiment are listed in table 1.

In this section we have taken the market basket analysis and compare the efficiency of the proposed method to the existing algorithms which is mentioned above. Both algorithms are coded using Visual Studio that uses Visual Basic and SQL programming language. The data sets have been generated for testing these algorithms. Two case studies have been done in analyzing the algorithm i) the execution time of the algorithm is tested to the number of transactions, ii) The execution time is executed to the number of the support.

Case i:

In this case where we are comparing the execution time of the transaction where any transaction may a contain more than one frequent itemsets. Here the minimum support is made constant.

Transaction	Apriori (mm:ss:ms)	Improved Algorithm
1000	00:15:82	00:14:10
2000	00:26:00	00:24:56
3000	00:36:77	00:35:58
4000	00:45:80	00:43:77
5000	00:50:07	00:46:23

Case ii:

Now the execution time of different algorithms is compared by varying the minimum support.

Support	Apriori (mm:ss:ms)	Improvised Algorithm
30	01:32:37	01:21:03
40	01:22:70	01:18:44
50	01:23:06	01:20:16
10	14:27:64	02:21:65

8. Conclusion

This new algorithm proposed for association rule mining is for finding the frequent itemsets. The present apriori algorithm has some bottlenecks we need to optimize and the proposed algorithm will give a new way for association rule where it reduces the candidate item sets. And we have also done some case studies about the existing algorithm above and we also listed the demerits of the existing systems and our proposed work is assured to overcome these bottlenecks we mainly concentrated to reduce the candidate itemset generation and also to increase the execution time of the process.

This algorithm works really efficiently against Apriori where support is low. Since it scans the database fewer times, I/O cycles are reduced and thereby decreasing the time of execution.

On another hand, it uses lesser memory than Apriori, saving crucial storage space. The increase in performance for small support (more transactions) is very good compared to Apriori, the runtime is reduced by 6 times.

The major limitation of the algorithm is that it has very slight increase in performance when the support is 30% or above.

9. REFERENCES

[1] Krishna Balan, Karthiga, Sakti Priya. An improvised tree algorithm for association rule mining using transaction reduction. International Journal of Computer Applications Technology and Research Volume 2– Issue 2, 166 - 169, 2013, ISSN: 2319–8656. .

[2] Feng WANG. School of Computer Science and Technology, Wuhan University of Technology Wuhan, China. 2008 International Seminar on Future BioMedical Information Engineering.

[3] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: Proc. of the l993ACM on Management of Data, Washington, D.C, May 1993. 207-216

[4] Chen Wenwei. Data warehouse and data mining tutorial [M]. Beijing: Tsinghua University Press. 2006.