Optimization of Execution Time using Association Rule Mining Algorithms

Smita R.Sankhe.
University of Mumbai
K.J.Somaiya College of Engineering

Kavita Kelkar.
University of Mumbai
K.J.Somaiya College of Engineering

ABSTRACT

The efficiency of mining association rules is an important field of Knowledge Discovery in Databases. The Apriori algorithm is a classical algorithm in mining association rules. This paper presents optimization of execution time for classicical apriori and an improved Apriori algorithm (DFR-Direct Fined and Remove) to increase the efficiency of generating association rules. This algorithm adopts a new method to reduce the redundant generation of sub-itemsets during pruning the candidate itemsets, which can form directly the set of frequent itemsetsand eliminate candidates having a subset that is not frequent in the meantime. This algorithm can raise the probability of obtaining information in scanning database and reduce the potential scale of item sets. Now a day's Hypermarket databases use data mining as a tool to optimize business solutions especially in the domain of sales and marketing. Common applications of data mining for any hypermarket include inventory management, tracking of customer behavior, finding frequent item sets and so on. The aim of this topic is to purpose efficiency analysis on data mining algorithms on aspects like Association Rule Mining. These aspects will help hypermarkets perform these functions effectively and hence increase their overall profit.

General Terms

Transactions, Knowledge Discovery, minimum support, minimum confidence.

Keywords

Data mining, association rule, Apriori algorithm, frequent itemset.

1. INTRODUCTION

Association rule mining, one of the most important and well researched techniques of data mining. It aims to extract interesting correlations, frequent interesting patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Association rule mining has a wide range of applicability such market basket analysis, medical diagnosis/ research, Website navigation analysis, homeland security and so on. The conventional algorithm of association rules discovery proceeds in two steps.[1]
In association rule mining algorithm, all frequent itemsets are found in the first step. The frequent itemset is the itemset that is included in at least *minsup* transactions. The association rules with the confidence at least *minconf* are generated in the second step. Data mining refers to extracting knowledge from large amounts of data by some technologies which include association rule mining, sequential pattern mining, clustering, and classification etc. Association rule mining which finds the relation between items or attributes is one of the most popular topics among these data mining technologies. For example, association rules mining can be applied to supermarkets. Managers of supermarkets can rearrange the positions between merchandises in supermarket and increase the profits.

2. RELATED WORK

Frequent Itemset Mining (FIM) is an important data mining problem which detects frequent itemsets in a transaction database.It plays a fundamental role in many data mining tasks that attempt to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers, clusters, etc. Many algorithms have been proposed to solve the problem,here here will be resenting optimization of execution time between classical apriori and improved aprirori algorithm (DFR-Direct Fined Remove)algorithm.
Apriori [Agrawal 1994], represents the candidate generation approach. Apriori is a Breadth First Search Algorithm (BFS) which generates candidate k+1-itemsets based on frequent k-itemsets.[2]
The frequency of an itemset is computed by counting its occurrence in each transaction.
(DFR) algorithm to mine a database consisting of remove and direct steps. When pruning the candidate's item sets, the algorithm eliminate non-frequent subset of candidates in the remove steps. In the direct steps, the algorithm directly generates the frequent item sets by computing and comparing the frequency of frequent k-item sets with k in the meantime.

2.1 Apriori Algorithm

Apriori algorithm (Agrawal et al. 1993), [7] is the most classical and important algorithm for mining frequent itemsets. Apriori is used to find all frequent itemsets in a given database DB. The key idea of Apriori algorithm is to make multiple passes over the database. It employs an iterative approach known as a breadth-first search (level-wise search) through the search space, where k-itemsets are used to explore (k+1)-itemsets. In the beginning, the set of frequent 1-itemsets is found. The set of that contains one item, which satisfy the support threshold, is denoted by L1. In each subsequent pass, begin with a seed set of itemsets found to be large in the previous pass. This seed set is used for generating new potentially large itemsets, called candidate itemsets, and count the actual support for these candidate itemsets during the pass over the data. At the end of the pass, determine which of the candidate itemsets are actually large (frequent), and they become the seed for the next pass. Therefore, L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. Then, a very significant property called Apriori property is employed to reduce the search space, where the Apriori property is described as ―"All nonempty subsets of a large itemset must also be large" or ―"If a set is not large, then its superset can’t be large either”. in the sense that if a set cannot pass a test, all of its supersets will fail the same test as well.

2.1.1 Apriori Algorithm

Apriori Algorithm can be used to generate all frequent itemset. A Frequent itemset is an itemset whose support is greater than some user-specified minimum support (denoted Lk, where k is the size of the itemset). A Candidate itemset is a potentially frequent itemset (denoted Ck, where k is the size of the itemset).[4]

STEP1:. Pass 1
 Generate the candidate itemsets in C1
 Save the frequent itemsets in L1
STEP2: Pass k
 2.1. Generate the candidate itemsets in Ck from the frequent itemsets in Lk-1 Join Lk-1p with Lk-1q, as follows: insert into Ck.
2.2 .select p.item1, p.item2, ., p.itemk-1, q.itemk-1 .
from Lk-1p, Lk-1q where p.item1 = q.item1, p.itemk-2 = q.itemk-2, p.itemk-1<q. itemk-1
STEP 3: Generate all (k-1)-subsets from the candidate itemsets in Ck.
3.1 Prune all candidate itemsets from Ck where, some (k-1)-subset of the candidate itemset is not in the frequent itemset Lk-1
STEP 4: Scan the transaction database to determine the support for each candidate itemset in C save the frequent itemsets in Lk.

2.1.2 The Apriori Algorithm Analysis

The association rule mining algorithm includes two key steps:

1. Find out all frequency itemsets-their support is greater than minimum support.
2. Find the association rule whose confidence is greater than min_conf given by user in each frequent large itemset.

The manipulation with redundancy result in high frequency in querying, so tremendous amounts of resources will be expended in time or in space.

2.1.3 The Drawbacks of Classical Apriori Algorithm

1. It may need to generate a huge number of candidate generations.
2. Each time when candidate generations are generated, the algorithm needs to judge whether these candidates are frequent item sets.
3. The manipulation with redundancy result in high frequency in querying, so tremendous amounts of resources will be expended whether in time or in space.

2.2 DFR: A New Improved Algorithm for Mining Frequent Item sets [1]

Efficiency has been concerned in the research of association rules mining. This paper presents an improved method called Direct-Fined-Remove (DFR) algorithm to mine a database consisting of remove and direct steps. When pruning the candidates itemsets, the algorithm eliminate non-frequent subset of candidates in the remove steps. In the direct steps, the algorithm directly generates the frequent itemsets by computing and comparing the frequency of frequent k-itemsts with k in the meantime.[1]

The contributions include:

Proposes an algorithm to raise the probability of obtaining information in scanning database and reduce the potential scale of itemsets.

2.2.1 Algorithm1: Direct-Fined-Remove for frequent **itemsets**

```
STEP1:L1 = {large 1-itemsets};
STEP2: for (k=2; Lk-1 ; k++) {
 Ck=apriori-gen(Lk-1,min_sup);
//new candidategeneration; }
 return L=UkLk;
Algorithm2: apriori-gen (Lk-1).Generate candidate generation
STEP1:For each itemset x∈ Lk-1 and each itemset y∈Lk-1
{
 if x.item1=y.item1,…, x.itemk-2=y.itemk-2,
x.itemk-1 < y.itemk-1 then {
 c= x∞y; // Put the k-1 elements in y join to
the back of y
 if has_infrequent_subset(c, Lk-1) then
 delete c; // Delete the candidate
generation that include non-frequent subitemsets
 else
 for each transactions t∈D {
// Scan D and accumulate itemsets
 if Count(t)<k then delete t;
 else {
 Ct=subset (Ck,t);// Take out the subsets
of candidate generation included in transaction t
 for each candidates c ∈ Ct
 c.count++;}}
 Lk={c∈ Ck |c.countεminsup}}
 return Ck;
Algorithm 3: has_infrequent_subset(c, Lk-1).
 Judge the elements of candidate generation
 for all (k-1)-subset s of c
 if s∈Lk-1 then
 return TURE;
 return FALSE;
```

Since DFR algorithm mines frequent itemsets
Without candidate generation, the query frequency
falls to about half level comparing with Apriori Algorithm. The size of database is reduced; meanwhile, the storage space and computing time are saved.

3. DISCUSSION

Table1: Comparative Study of Apriori and Direct-Fined Remove Algorithm

Algorithm	Advantages	Disadvantages
Apriori algorithm	Finding out all frequent itemsets which are greater than or equal to user-specified minimum support threshold;	It may need to generate a huge numberof candidate generations.
	Generating association rules from frequent itemsets.	Each time when candidate generations are generated, the algorithm needs to judge whether these candidates are frequent item sets.

	Apriori repeatedly uses above two steps to find frequent itemsets until there are no frequent itemsets to be found	The manipulation with redundancy result in high frequency querying, so tremendous amounts of resources will be expended whether in time or in space.
DFR(Direct-Fined-Remove) Algorithm	Adopts a new method to reduce the redundant generation of sub-itemsets during pruning the candidate itemsets	It still need further research to find methods to estimate how much improvement this Apriori algorithm can implement
	It eliminates candidates having a subset that is not frequent in the meantime	
	It can raise the probability of obtaining information in scanning database & reduce the potential scale of itemsets	

4. EXPERIMENTAL RESULTS

This section compares the experimental performance of DFR with classical Apriori algorithm. Two algorithms are implemented with java language running under JDK1.6 environment .All experiments are performed on Intel Pentium IV with 512 MB memory. With consideration of database having handful 150 records result for most frequent combination and total time elapsed were obtained.
Approximate time required for executing the algorithm is:

Table 2: comparison of two algorithms with Time Elapsed for Execution

Algorithm	Time
Apriori Algorithm	5590 ms approx
DFR Algorithm	1800ms approx

Graphical output

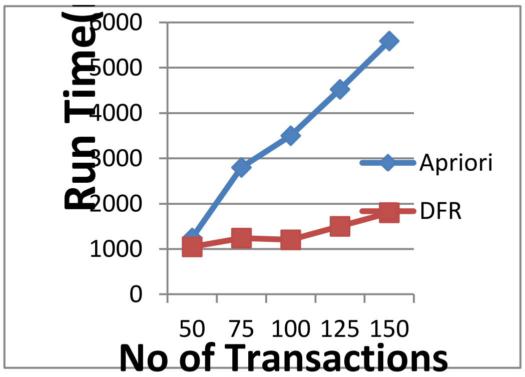

Fig 1: Comparision in different parameters

Table 3: comparison with different parameters

Sr. No	Properties	Apriori Algorithm	DFR(Direct Fined Remove)Algorithm
1	Dataset Used	Book Records	Book Records
2	Data Structure Used	Hash table	Hash Table, Hash Tree
3	No.of Database Scans/cycle Performed	15	3
4	Memory Consumed	135MB	147MB
5	Running Tmie	5590 ms approx	1800 ms approx

5. CONCLUSION

In this paper, The Apriori algorithm of mining association rule according to the property of frequent items, and optimized execution time requird for an improved algorithm for generating the k-frequent itemsets designed to reduce the number of database scanning and the redundancy. When pruning the candidates itemsets, the algorithm eliminate non-frequent subset of candidates, and then directly generate the frequent itemsets. According to the experiment's results, the efficiency of improved algorithm is better. Meanwhile, still need further research to find methods to estimate how much improvement this Apriori algorithm can implement.

6. REFERENCES

[1] Sheng Chai, Hai-Chun Wang, Ji-Fan Qiu, 2009 "DFR: A New Improved Algorithm for Mining Frequent Itemsets", *IEEE/ACM Fuzzy Systems and Knowledge Discovery.*

[2] Jen-PengHuang; Guo-ChengLan; Huang-Cheng Kuo Tzung Pei Hong, 2008"A Decomposition Approach for Mining Frequent Itemsets", *Intelligent Information Hiding and Multimedia Signal Processing*. IIHMSP 2008. *Third International Conference 2008* Volume: 2, On page(s): 605-608, Dec 2008.

[3] Jen-PengHuang; Guo-ChengLan; Huang-Cheng Kuo Xuezhi chi , "A New Matrix-Based Association Rules Mining Algorithm",2012 *Proc. 9th IEEE Int. Conf. on Fuzzy systems and Knowledge Discovery(FSKD2012).*

[4] D. Zhang, Y. Wang, and W. Kinsner (Eds.), "An Algorithm to Improve the Effectiveness of Apriori", *Proc. 6th IEEE Int. Conf. on Cognitive Informatics (ICCI'07),2007.*

[5] J. Han, M. Kamber 2006, *Data Mining,Concepts and Techniques*",Morgan Kaufmann Publishers, San Francisco, CA.

[6] Pimgxiang,ChenJiangping,Bian, Fuling,"*Adeveloped algorithm of apriori based on association anlaysis,Geo-special Information science, 2004.*

[7] Agrawa IR, Imielinski T, Swami A Miningassociation rules between sets of items in largedatabases (C). In:Buneman P, Jajodia S,eds. Proc.of the ACM SIGMOD Conf. on Management ofData.1993.