Research of an Improved Apriori Algorithm in Data Mining Association Rules

Jiao Yabing

***Abstract*—Apriori algorithm is the classic algorithm of association rules, which enumerate all of the frequent item sets. When this algorithm encountered dense data due to the large number of long patterns emerge, this algorithm's performance declined dramatically. In order to find more valuable rules, this paper proposes an improved algorithm of association rules, the classical Apriori algorithm. Finally, the improved algorithm is verified, the results show that the improved algorithm is reasonable and effective, can extract more value information.**

***Index Terms*—Association rules apriori algorithm frequent item.**

I. Introduction

Association rules problems were first brought out by Agrawal and others in 1993, which were researched by many other researchers after that. They optimized the original algorithm, such as bringing in random sampling, parallel thoughts, adding reference point, declining rules, changing storing framework, etc. Those works were aimed at improving the efficiency of algorithm rules, spreading the applications of association rules from initial business direction to other fields, such as education, scientific research, medicine, etc. [1]

Association rules mining is to discover the associations and relations among item sets of large data. Association rules mining is an important branch of data mining research, and association rules is the most typical style of data mining.

Presently, association rules mining problems are highly valued by the researchers in database, artificial intelligence, statistic, information retrieval, visible, information science, and many other fields. Many incredible results have been found out. What can efficiently catch the important relationships among data are simple forms of association rules and easily to explanation and understanding. Mining association rules problems from large database has become the most mature, important, and active research contents. [2]

II. APRIORI Algorithm Statement

Apriori algorithm is the originality algorithm of Boolean association rules of mining frequent item sets, raised by R. Agrawa and R. Srikan in 1994. The core principles of this theory are the subsets of frequent item sets are frequent item sets and the supersets of infrequent item sets are infrequent item sets. This theory is regarded as the most typical data mining theory all the time. [3]

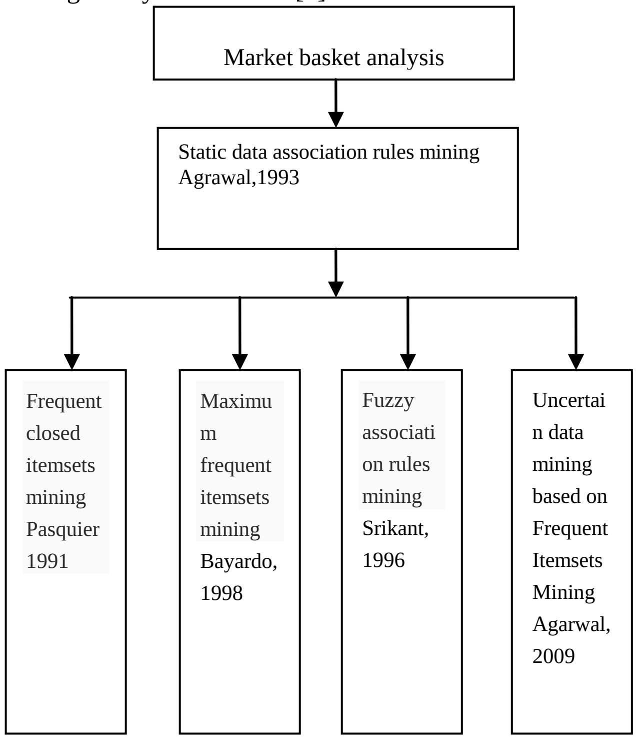
Market basket analysis
Static data association rules mining Agrawal,1993
Frequent closed itemsets mining Pasquier 1991
Maximum frequent itemsets mining Bayardo, 1998
Fuzzy association rules mining Srikant, 1996
Uncertain data mining based on Frequent Itemsets Mining Agarwal, 2009

Fig. 1. Association rules data mining classification.

The algorithm is used to find out all the frequent item sets. In the first iteration, item set A directly constitutes the first candidate item set C_1. Assume that $A=\{a_1,a_2,\ldots,a_m\}$, then $C_1=\{\{a_1\},\{a_2\},\ldots,\{a_m\}\}$. In the Kth iteration, firstly, the candidate item set C_k of this iteration emerges according to the frequent item set L_{k-1} found in the last iteration. (The candidate item set is the potential frequent item set and is the superset of the K-1th frequent item set. Item set with k candidate item sets is expressed as Ck, which was consisted by k frequent item sets L_k.) Then distribute a counter which has a initial value equals to zero to ever item set and scan affairs in database D in proper order. Make sure every affairs belongs to each item sets and the counter of these item sets will increase. When all the affairs have been scan, the support level can be gotten according to the actual value of $|D|$ and the minimum support level of the certain C_k of the frequent item set. Repeat the process until no mew item occurs. [4]

The algorithm includes two key processes: connecting step and pruning step.

Connecting step: in order to get L_k, connect L_{k-1} with itself. Set this candidate as C_k and assume L_1 and L_2 are the item sets of L_{k-1}. $L_{i[j]}$ is the jth item of L_i. Assume the affairs and items of the item set are in the dictionary order. Execute the connection $L_{k-1} \triangleright\triangleleft L_{k-1}$, in which the elements of L_{k-1}, L_1 and L_1, are connectable, if they have the same first $(k\text{-}2)^{th}$ items.

Manuscript received August 4, 2012; revised September 10, 2012.
The author is with the Shandong Yingcai University Jinan China (e-mail: Jiaoyabing@163.com).

That is, the elements of L_{k-1}, L_1 and L_1, are connectable, if $(L_1[1]=L_2[1]) \wedge (L_1[2]=L_2[2]) \wedge \ldots\ldots(L_1[k\text{-}2]=L_2[k\text{-}2]) \wedge (L_1[k\text{-}1]=L_2[k\text{-}1])$. The requirement of $(L_1[k\text{-}1]<L_2[k\text{-}1])$ simply assure no repetition. The result item set of connecting L_1 and L_2 is $L_1[1]\ L_1[2]\ldots L_1[k\text{-}1]\ L_2[k\text{-}1]$.

Pruning step: C_k is the superset of L_k, that is that the members of it could be frequent or not, but all the k frequent item sets are all include in C_k. Scan the database, clear the counters of every candidate item sets of C_k to assure L_k. However, C_k might be very large, and then the amount of calculation will be huge. In order to decrease C_k, there are following method using the prosperities of Apriori: any infrequent item sets with k-1 items are not the subset of frequent item sets with k items. Consequently, if the (k-1) subset of a candidate item set with k items are not in L_{k-1}, the candidate item set is not frequent and can be deleted in C_k.

Apriori algorithm employs the bottom up, width search method, it include all the frequent item sets. When the database of affairs is sparse (such as market basket database), the form of frequent item set of this database is usually short. Apriori algorithm and similar algorithm can get favorable properties under this condition. However, once this kind of algorithms meet dense database (such as, telecom, population census, etc.), as large amounts of long forms occur, the properties sharply drop.

III. Improve APRIORI Algorithm

A. The Theory of Algorithm

Apriori algorithm has sound theory base, but the focus is its efficient issue. Agrawal raised several improving methods in person, like Apriori Tid, Apriori All, etc. After that many optimized methods were raised based on the framework of Aprioir algorithm. This study introduced an algorithm that decrease the number of candidate items in the candidate item set C_k.

In the Apriori algorithm, C_{k-1} is compared with support level once it was found. Item sets less than the support level will be pruned and L_{k-1} will come out which will connect with itself and lead to C_k. The optimized algorithm is that, before the candidate item sets C_k come out, further prune L_{k-1}, count the times of all items occurred in L_{k-1}, delete item sets with this number less than k-1 in L_{k-1}. In this way, the number of connecting items sets will decrease, so that the number of candidate items will decline.

B. The Realization of Algorithm

According to the properties of frequent item sets, this algorithm declines the number of candidate item sets further. In other words, prune L_{k-1} before C_k occur using L_{k-1}. This algorithm can also be described as following:

Count the number of the times of items occur in L_{k-1} (this process can be done while scan data D);

Delete item sets with this number less than k-1 in L_{k-1} to get L_{k-1}. To distinguish, this process is called Prune 1 in this study, which is the prune before candidate item sets occur; the process in Apriori algorithm is called Prune 2, which is the prune after candidate item sets occur. Thus, to find out the k candidate item sets, the following algorithm can be taken:

Prune $l(L_{k-1})$, that is executing Prune 1 to L_{k-1};

Use L_{k-1} to connect with its elements and get the k candidate item sets C_k;

Prune $2(C_k)$, that is executing Prune 2 to and finally get the k items candidate set which should calculate its support level (the superset of k items frequent set)

The following is the description of the optimized algorithm:

Input: affairs database D: minimum support level threshold is minsup

Output: frequent item sets L in D

```
1)  $L_1$=frequent_1-itemsets($D$);
2)  For ($k$=2;$L_{k-1}\neq\varphi$;$k$++);
3)  Prune$1(L_{k-1})$;
4)  $C_k$=apriori_gen($L_{k-1}$;minsup);
5)  for all transactions $t\in D$
{
6)  $C$= sumset ($C_k$,$t$); find out the subset including $C_k$
7)  for all candidates $c\in C_t$
8)  {  $c$.count ++; }
9)  $L_k$ =$\{c\in C_k|c.\text{count}\geq\text{minsup}\}$   //result of Prune
    $2(C_k)$   }        }
10) Return Answer $\cup_k L_k$
```

Algorithm: Prune Function:

Input: set k-1 frequent items of L_{k-1} as input parameter

Output: go back and delete item sets with this number less than k-1 in Lk-1

Procedure Prune $1(L_{k-1})$

```
1)  for all itemsets  $L_1\in L_{k-1}$
2)  if count($L_1$)≤$k$-1
3)  then delete all $L_j$ from $L_{k-1}$ ($L_1\in L_{k-1}$)
4)  reture $L'_{k-1}$ // go back and delete item sets with this
    number less than $k$-1 in $Lk$-1
```

Chart 3-1 shows the process of the algorithm finding out the frequent item sets, the minimum support level is 2.

TABLE I: Candidate Item Sets C_1 Frequent Item Sets L_1

TID	Item List
T1	A,B,D
T2	A,B,C,D
T3	A,B,D,E
T4	B,E,F
T5	A,B,D,F
T6	A,C,D,E

Occur Candidate Item Set C_2

Item Set	Support Level	Item Set	Support Level
A,B	4	B,F	2
A,C	2	C,D	2
A,D	5	C,E	1
A,E	2	C,F	0
A,F	1	D,E	2
B,C	1	D,F	1
B,D	4	E,F	1
B,E	2		

Occur Frequent Item Set L_2

Item Set	Support Level	Item Set	Support Level
A,B	4	B,E	2
A,C	2	B,F	2
A,D	5	C,D	2
A,E	2	D,E	2
B,D	4		

L'_2 after pruning

Item Set	Support Level	Item Set	Support Level
A,B	4	B,D	4
A,C	2	B,E	2
A,D	5	C,D	2
A,E	2	D,E	2

Occur Candidate Item Set C_3

Item Set	Support Level	Item Set	Support Level
A,B,C	1	A,C,E	1
A,B,E	4	A,D,E	2
A,D,E	1	B,D,E	1
A,C,D	2	C,D,E	1

after pruning is empty

Item Set	Support Level
A	5
B	5
C	2
D	5
E	3
F	2

The process of the algorithm finding out the frequent item sets

IV. Advantages of the Algorithm

The basic thought of this optimized algorithm is similar with the apirori algorithm, which is they all get the frequent item set L_1 which has support level larger or equal to the given level of the users via scan the database *D*. Then repeat that process and get L_2，L_3……L_k.

But they also have differences. The optimized algorithm prunes L_{k-1} before C_k is consisted. In other words, count the frequent item set L_{k-1} which is going to connect. According to the result delete item sets with this number less than k-1 in L_{k-1} to decrease the number of the connecting item set and remove some element that is not satisfied the conditions. This will decrease the possibility of combination, decline the number of candidate item sets in C_k, and reduce the times to repeat the process. For large database, this algorithm can obviously save time cost and increase the efficiency of data mining. This is what Apriori algorithm do not have.

Although this process can decline the number of candidate item sets in C_k and reduce time cost of data mining, the price of it is pruning frequent item set, which could cost certain time. For dense database (such as, telecom, population census, etc.), as large amounts of long forms occur, the efficiency of this algorithm is higher than Apriori obviously.

V. Summary

This study is focused on how to solve the efficient problems of Apriori algorithm and raise another association rules mining algorithm. This has certain reference value to research and solve the issues of data expiation and information lacking. It hopes to dig out more useful information.

Acknowledgment

Author Introduction: Jiao Yabing, female, (1978--), Master, Lecturer, research direction: decision support theory and system analysis, teacher of ShanDong Yingcai University. Shandong Province University Science and Technology Plan :J11LG71

References

[1] J. Han and M. Kamber, *Conception and Technology of Data Mining*, Beijing: China Machine Press, 2007.
[2] J. N. Wong, translated, *Tutorials of Data Mining*. Beijing. Tsinghua University Press, 2003.
[3] Y. Yuan, C. Yang, Y. Huang, and D. Mining, *And the Optimization Technology nd Its Application*. Beijing. Science Press, 2007.
[4] C. Wang, R. Li, and M. Fan, "Mining Positively Correlated Frequent Itemsets," *Computer Applications*, vol. 27, pp. 108-109, 2007
[5] Y. S. Kon and N. Rounteren, "Rare association rule mining and knowledge discovery:technologies for frequent and critical event detection. H ERSHEY," PA: *Information Science Reference*, 2010
[6] W. Sun, M. Pan, and Y. Qiang, "Inproved association rule mining method based on t statistical," *Application Research of Computers*. vol. 28, no. 6, pp. 2073-2076, Jun, 2011.