Design and Implementation of Improved Algorithm for Association Rules Mining

ZHANG Lin
Dept. of Computer Science and Technology
AnHui SanLian University
HeFei, AnHui, P.R.C., 230000
sirenrabbit@sina.com

ZHANG Jian-li
New Star Institute of Applied Technology
NO.451, Huangshan Road
Hefei, Ahhui, P.R.C., 230031
Wjxc1982@126.com

***Abstract*—For traditional algorithm of association rules mining based on frequent item sets and ignored the exception rules, as well as the traditional issues such as frequent item sets algorithm can be costly, a new association rules mining algorithm which isn't based on frequent item sets is given in this paper. This method scanned the database once firstly, and then uses the logical and set theory operations to generate association rules, including frequent association rules and exception rules. Through case analysis, results showed that the algorithm with high accuracy, low cost, and the ability to generate exception rules could provide some reference data for exception knowledge mining.**

Keywords-association rules; data mining; frequent item sets; algorithm; exception rules;

I. INTRODUCTION

Traditional association rules mining algorithm is based on frequent item sets, but the calculate process of frequent item sets often exist problems such as database scanned too many times or database traversal needs too much space. These problems will inhibit the efficiency of association rules mining algorithm. In addition, the rules which are discovered by association rules mining algorithm that based on frequent item sets often followed the most data in database, and the rules called exception rules which supported by few data often be ignored just because their data support less than the minsupport, and these ignored rules often have a great idea of value. In this paper, based on recent years' association rules mining algorithms[1-9], we advance a new association rules mining algorithm which is base on logic and sets operation. This algorithm is no longer based on frequent item sets, and can mining out more rules including exception rules, and its efficiency is high than traditional association rules mining algorithm which is based on frequent item sets.

II. ALORITHM DESCRIPTION[10]

A. Some concepts

Suppose I={i1,i2,...,im} is a collection of items, T={t1,t2,...,tn} is a transaction database, which is made up of a series of transactions with a unique identification TID, and each transaction ti (i=1,2,...,n) is a subset of I.

In the traditional association rule mining algorithm, strong association rule is defined as: For the T and I each association rule which meet the minsupport and minconfidence is the strong association rule.

Due to the support degree has nothing to the algorithm advanced in this paper, the strong association rule is defined as: If the number of transactions that contained I1∪I2 and the number of transactions that contained I1's ratio that is Count (I1∪I2)/Count (I1) is greater than or equal to minconfidence which is set by user, I1⇒I2 will be an association rule for T and I.

B. Design of improved algorithm for association rules mining

1) Produce initial linked list

If the collection of items is I={i1,i2,...,im}, transaction database is T={t1,t2,...,tn}, then after scan database once we can establish a linked list L with length m. Each node in L is composed of 4 fields: exp, data, count and next. In the 4 fields, exp field is used for store current node's item sets Ii(Ii⊆I); data field is used for store the TID of transaction which contains item sets Ii by bits whit length n (If transaction j contains item sets Ii, the corresponding bit will be set for 1, otherwise will be set for 0); count field is used for store the number of 1 in data field(that is the number of times that Ii appears in T); and next field is used for store the point which is point to next node. In the initial linked list L, the i-th node's exp field is ii, its data field is stored the TID of transactions which contain ii, its count field is stored the number of times that ii appears in T. For example, there is a sample database shown as following table Ⅰ:

TID	Item Sequence
1	B,C,D
2	A,B,C
3	A,C,E
4	B,C,D,E
5	A,C,D
6	A,D
7	B,C
8	A,B,C,D
9	A,B,C,E

TABLE I. sample database Ⅱ

By table I we know that I={A,B,C,D,E}, T={1,2,3,4,5,6,7,8,9}, item A appears in transaction 2, 3, 5, 6, 8 and 9, account 6 times. So the first node's value of exp field in L is A, its value of data field is 011011011, and its

value of count field is 6. Thereby, we can get the initial linked list L after scan table Ⅰ. The result shown as Figure 1:

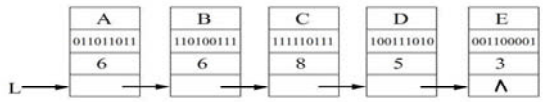

Fig.1 Initial linked list L for sample database Ⅰ

2) The idea of algorithm

Compare each 2 node in the initial linked list L. If node P and node Q's exp field have nothing in common, that is p->exp∩Q->exp=∅, then establish a new node New, New->exp=P->exp∪Q->exp， New->data=P->data AND Q->data (namely P->data and Q->data do a logical-and operation), New->count=Count(New->data) (namely New->count is the number of 1 that appears in New->data). After the establishment of the new node, calculate values of New->count/P->count and New->count/Q->count respectively. If New->count/P->count≥minconfidence, then P⇒Q is a rule for T and I, if New->count/Q->count≥minconfidence, then Q⇒P is a rule for T and I. Finally, look over the linked list L, judge if the node New is belong to L. If New∉L, then insert New to L. When all node in L had compared, look over L to find if there are new nodes inserted. If there are new nodes inserted,that means there may also rules are not found, therefor compare all node with new nodes that just inserted until L is no longer has new nodes insert.

Establish 5 points: First, Last, End, Start and P. In the 5 points, First is used for point to original comparision points; Last is used for point to L's current last node; End is used for point to L's original last node, its initial value is L; Start is used for point to the previous node of new nodes, that is the original last node, its initial value is also L; and P is used for point to current comparision points. After comparison in initial linked list L, if End=Last, that means no new nodes inserted, program ends. Otherwise, Start=End, End=Last, then into the comparision loop again.

Specific steps of the algorithm is as follows:

step 1: Establish the initial linked list L by scan the database once.

Step 2: Initialize the Last and End points, Last points to the last node, End=L.

Step 3: If End=Last, the program ends, otherwise, go to step 4.

Step 4: Initialize First and Start points, First=L, Start=End, End=Last.

Step 5:Compare the node between First and End to the node between Start and End, then insert new nodes that meet the criteria to L.

Step 6: Compare End and Last again, if End=Last, program ends, otherwise (there are new nodes inserted), go to step 4.

3) Description of algorithm

Main programs of this algorithm's pseudocode is as follows:

```
#define MC minconfidence
main( )
{   Boolean Flag=false;
    Boolean Ifadd=false; L
while(End!=Last)
{ First=L;
    Start=End;
    End=Last;
    while(First!=End)
    { if(Flag)
        { P=First; }
        if(First==Strart)
        { Flag=true; }
    else
    { P=Start; }
        while(P!=End)
        { P=P->next;
```

```
            if(First->exp∩P->exp==∅)
            {
            New=Node new(); //establish a new node
            New->exp=First->exp∪P->exp;
            New->data=First->data AND P->data;
            New->count=Count(New->data);
            New->next=NULL;
            if(New->count/First->count>=MC)
            { Ifadd=true;
              printf("P->exp⇒First->exp"); }
            if(New->count/P->count>=MC)
            { Ifadd=true;
              Printf("First->exp⇒P->exp"); }
            if(Ifadd==true&&New∉L)
            { Last->next=New;
              Last=New; } } }
First=First->next; }
Flag=false; } }
```

III. EXAMPLES ANALYSIS

Suppose there is a sample database shown as follows in table Ⅱ:

TABLE II. sample database Ⅱ

TID	Item Sequence
1	A, B, C, D
2	B, C, E
3	A, B, C, E
4	B, D, E
5	A, B, C, D

In this transaction database, I={A,B,C,D,E}, T={1,2,3,4,5}.

Step 1: Establish the initial linked list L by scan the database once.

After scanned the database, we can get the initial linked list L shown as follows in Figure 2:

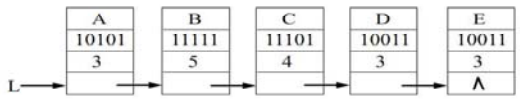

Fig.2 Initial linked list L for sample database II

Step 2: Initialize the Last and End points, Last points to the last node, End=L, shown as follows in Figure 3:

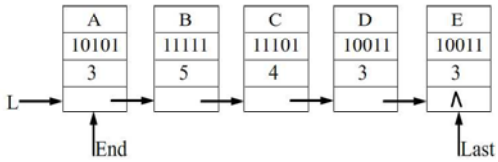

Fig.3 Initialize point Last and End for linked list L

Step 3: On account of End≠Last, last the program proceed to step 4.

Step 4: Initialize First and Start points, First=L, Start=End, End=Last, shown as follows in Figure 4:

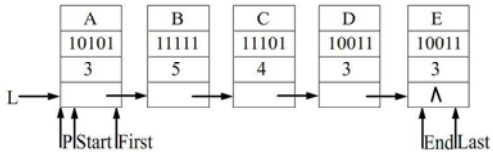

Fig.4 Initialize point First and Start for linked list L

Step 5:Compare the node between First and End to the node between Start and End, then insert new nodes that meet the criteria to L, and output the appropriate rules, shown as follows in Figure 5.

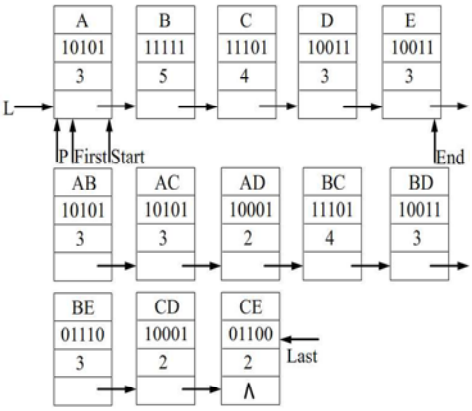

Fig.5 Linked list L after the first node compare

And gained the following rules: A⇒B, B⇒A, A⇒C, C⇒A, A⇒D, D⇒A, B⇒C, C⇒B, B⇒D, D⇒B, B⇒E, E⇒B, D⇒C, E⇒C

Step 6: Compare End and Last again, because End≠Last, so we repeat step 4 to step 6 until no new nodes insert to L. Its running process is shown as follows in Figure 6:

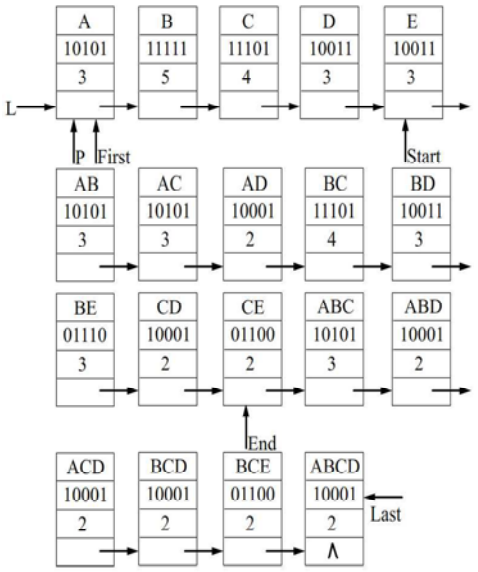

Fig.6 Linked list L after the second node compare

After comparison again we gained the following new rules: A⇒BC, BC⇒A, A⇒BD, BD⇒A, A⇒CD, CD⇒A, B⇒AC, AC⇒B, AD⇒B, CD⇒B, CE⇒B, C⇒AB, AB⇒C, AD⇒C, BD⇒C, BE⇒C, D⇒AB, AB⇒D, D⇒AC, AC⇒D, D⇒BC, E⇒BC, AB⇒CD, CD⇒AB, AC⇒BD, BD⇒AC, AD⇒BC

Just because End≠Last still come into existence，so we repeat step 4 to step 6 again. Its running process is shown as follows in Figure 7:

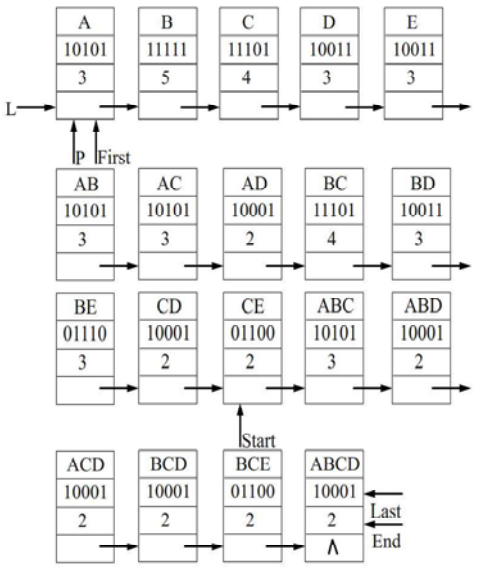

Fig.7 Linked list L after the third node compare

Then we gained the following new rules: A⇒BCD, BCD⇒A, ACD⇒B, ABD⇒C, ABC⇒D, D⇒ABC

Now End=Last, program ends. We get all of the association rules. These rules not only contains the rules based on frequent item sets, but also contains expection rules.

IV. ALGORITHM ANALYSIS

This algorithm just need to scan database once, and does not have to produced frequent item sets. Compare to traditional association rules mining algorithm based on frequent item sets, this algorithm's implementation efficiency is large improved, and produced comprehensive rules than association rules mining algorithm based on frequent item sets. It not only contains rules produced by frequent item sets, but also contains rules produced by the datas whose support is less than minsupport.

In performance tests, compare to traditional association rules mining algorithm based on Apriori algorithm to produce frequent item sets, and using the database that containing 18 items and 2714 transactions. The running time count from input to output. The result shown that compare to traditional algorithm the efficiency of algorithms in this paper is improved about 60%, and produced exception rules which traditional algorithms can not produced.

V. THE ENDING

This paper advanced a new association rules mining algorithm which does not based on frequent item sets. This algorithm's mining efficiency and its comprihensive of mining rules are all over traditional association rules mining algorithm. But if the number of item are too much, or the minconfidence's value is set too small, it'll produce a linked list L whose length will so large. At that time, we can consider the linked list for subparagraph processing, such as we can establish a new linked list for new node to replace insert the node to L.

REFERENCES

[1] Agrawalr, Imielinski, Swamia. "Mining association rules between sets of items in large databases." Proc of ACM SIGMOP Conference on Management of Data.New York:ACM Press, pp. 207-216, 1993.

[2] WuFan, Chiangsw, Linjr. "A new approach to mine frequent patterns using item-transformation methods," Information Systems, pp. 1056-1072, July, 2007.

[3] Zhang Zhongping, Li Yan, Lin Zhijie etc., "Frequent item sets mining algorithm based on index array," Application Research of Computers, pp. 44-46, January, 2009.

[4] Liu Yingdong, Leng Mingwei, Chen Xiaoyun, "Maximal Frequent Itemsets Mining Algorithm Based on Linked List Array," Computer Engineering, pp. 89-90,93, June, 2010.

[5] Wang Pingshui, "Research on association rules mining algorithm," Computer Engineering and Applications, pp. 115-116, 2010.

[6] Yang Ping, Yang Tianshe, Du Xiaonin etc., "A class-attribute interdependency maximization based algorithm for supervised discretization," Control and Decision, pp. 592-596, April, 2011.

[7] Wang Xiangrui, "Research and Application of Association Rules in Data Mining Technology," Coal Technology, pp. 205-206, August, 2011

[8] Zhang Guanglu, Lei Jingsheng, Wu Xinghui, "An Improved Apriori Algorithm for Mining Association Rules," Computer Technology and Development, pp. 84-88,92, June, 2010.

[9] Liu Xiaowei, Chen Junli, Qu Shifu etc., "Improved Apriori algorithm.Computer Engineering and Applications," Computer Engineering and Applications, pp. 149-151,159, November, 2011.

[10] Mao Guojun, Duan Lijuan, Wang Shi, Principle and algorithm of data mining, CA: Tsinghua University press, 2007.