A DEVELOPED ALGORITHM OF APRIORI BASED ON ASSOCIATION ANALYSIS

Li Pingxiang[1] Chen Jiangping[2] Bian Fuling [2]

1. State Key Laboratory of information Engineering in Surveying Mapping &Remote Sensing WuHan University

2. Center of Spatial Information and digital Engineering WuHan University, 430079

Abstract

Based on association analysis , an improved algorithm of Apriori is presented in the paper. The main idea of the algorithm are:

(1) Count the probability of each attribute item(A_1 , A_2,…A_m) of a DB by scanning the DB first time;

(2)The probability of any two items A_k and A_m appeared synchronously in one record is P_{km}.

$\min(P_k , P_m) \le P_{km} \le P_k * P_m$,

if A_k and A_m is total correlation, then the P_{km} is the minimum of the P_k and P_m,;

if A_k and A_m is total independent, then the P_{km} is $P_k * P_m$;

So we can estimate :

$P_{km} = (a*\min(P_k, P_m)+b*P_k*P_m)/(a+b)$; $a+b=1$

Parameter “a” is the probability while A_k and A_m are total correlation,

Parameter “b” is the probability while A_k and A_m are total independent,

Parameter “a” and “b” can use other method such as association analysis to count. In this paper a method for calculate the parameter “a” and ”b” with association analysis is provided.

if P_{km} is more than the threshold value which the user set, then A_k , A_m are the frequent itemsets.

You can use the method which described above to find out all the frequent itemsets without scanning DB so many times.

(3)Count the support of the frequent itemsets by scanning the DB another time;

(4)Output the association rules from the frequent itemsets.

The detailed algorithm and it's sample are described in the paper . At last we compared it with algorithm apriori. The best quality is that the algorithm in our paper reduce the times of scanning DB.

KEY WORDS: Association Rule Algorithm Apriori Frequent Itemset Association Analysis

1 Introduction

With the development of the information technology and application of database. The collected data far exceed people's ability to analyze it. Thus new and efficient methods are needed to discover knowledge from large databases. And data mining appeared.

Data mining is the core of knowledge discovery in databases. It’s a procedure to find the useful and potentially knowledge in database. Association rules are one of the most important knowledge of data mining’s result which can be defined as the relation and dependency between the itemsets by given support and confidence in database.

In the algorithms of the association rules mining, apriori is the ancestor which offered by Agrawal R in 1993. The main idea of the apriori is scanning the database repeatedly. With the theory that the subset of the frequent itemset are frequent itemsets too, you can gain the length of frequent (k+1)-itemsets L_{k+1} from the frequent k-itemsets L_k. At the k time it scanned the database only the candidate itemsets C_{k+1} which generate from the L_k was concerned. Later the

appear times of the C_{k+1} can be verified by another scanning database. There was a drawback that cost much time and memory to generate candidate C_{k+1} . It would be more worst when the length of the frequent itemsets were very long and their support were very small.

Example: the number of candidate 2- itemsets will be more than 10^7 if the number of frequent 1-itemsets are 10^4 in algorithm apriori. Moreover, In apriori it would generate $2^{100} \approx 10^{30}$ candidate itemsets to find a frequent itesets with the length of 100,also, it needs to scan the database repeatedly for 100 times if the length of the frequent itemsets is 100.

From the above it can be concluded that the key problem of the apriori is it take too much time to mining the frequent itemsets. The time mainly costs in two areas, one is the time for scanning the large database repeatedly the other is for generating frequent itemsets with JOIN. There are a lot of improved algorithm for apriori such as AprioriTID, Apriori Hybri, Multiple oins, Reorder and Direct etc. The main idea of all these algorithm is according the theory that the subset of a frequent itemset is a frequent itemset and the superset of a infrequent itemset is a infrequent itemset. They scan the database repeatdly to mining the association rules.

There is another feature for algorithm AprioriTID, the support of the candidate frequent itemsets are calculated only at the first time it scanned the database D and also generated candidate transaction database D' which only includes the candidate frequent itemsets. Then the latter mining are based on the database D', It reduce the time of I/O operation because D' is smaller than D, so, it enhance the efficiency of the algorithm.

For algorithm Apriori Hybri, it is the combination of algorithm Apriori and AprioriTID. When the candidate transaction database D' couldn't be contained in the RAM, it mines with algorithm Apriori otherwise with algorithm AprioriTID.

The most important step in mining association is generation frequent itemsets. In algorithm apriori the most time is consumed by scanning the database repeatedly. It would reduce the running time of the algorithm by reducing the times it scans the database far and away. In this paper a method of mining frequent itemsets by evaluating their probability of supports based on association analyzing were mentioned. First , it gained the probability of every 1-itemset by scanning the database, the 1-itemset with the more larger support than the probability the user sets would be frequent 1-itemsets. Second, it evaluates the probability of every 2-itemset ,every 3-itemset , every k- itemset from the frequent 1-itemsets. Third, it gains all the candidate frequent itemsets. Fourth, it scans the database for verifying the support of the candidate frequent itemsets, Last the frequent itemsets are mined and association rules also do. In the method it reduces a lot of times of scanning database and shortened the calculate time of the algorithm.

2 Related concepts

In this section, we present the definitions of the concepts that are used to describe the improved algorithm. Let us start from the following definitions for association rules.

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of all *items*, where an item is an object with some predefined attributes (e.g., price, weight, etc.). A *transaction* $T = \langle tid, I_t \rangle$ is a tuple, where ***tid*** is the *identifier* of the transaction and $I_t \subseteq I$. A transaction database ***T*** consists of a set of transactions. An itemset is a subset of the set of items. A k-itemset is an itemset of size k . We write itemsets as $S = i_{j1} i_{j2} \cdots i_{jk}$, $S \subseteq I$ omitting set brackets.

Definition 1. An association rule takes the form X⇒Y where X⊂*I*, Y⊂*I*, and X∩Y=Φ. the support of the rule X⇒Y in transaction database is :

$$\text{support}(X \Rightarrow Y) = |\{T : X \cup Y \subseteq T, T \in D\}| / |D|$$

marked by support(X⇒Y).

Definition 2. the confidence of the rule X⇒Y in transaction database is :

$$\text{confidence}(X \Rightarrow Y) = |\{T : X \cup Y \subseteq T, T \in D\}| / |\{T : X \subseteq T, T \in D\}|$$

marked by confidence(X⇒Y).

The mining association rules problem is generating association rules with itemsets which have more larger or equal support and confidence than the user set the minsupp and minconf.

Definition 3. Let $A = A_1, A_2, \ldots, A_n$ be a set of all events. If

$P(A_{i1}Ai_2 \ldots Ai_k) = P(Ai_1)P(Ai_2)\ldots P(Ai_k)$, (1)

For any $k(1<k\leq n)$,any $1 \leq i_1 < i_2 < \ldots < i_k \leq n$, the formula (1) is rignt, then

$A_1, A_2, \ldots, A_n$ are inter-independent events.

Candidate frequent itemsets refer to the itemsets whose probability is larger than the user sets, they may be the last frequent itemset or not.

3 The Improved algorithm for apriori

3.1 the main idea of the improved algorithm

Let $P_1, P_2 \ldots P_n$ are the independent probability of every item $A_1, A_2 \ldots A_n$,the probability for any two item A_k, $A_m(P_k<P_m)$both appeared in one transaction is P_{km} .

If A_k and A_m are total non-correlation, from definition 3 it can be concluded that $P_{km} = P_k * P_m$, if A_k and A_m are total correlation, then P_{km} is the minimum of the P_k and P_m that is P_k , so , $P_k * P_m \leq P_{km} \leq P_k$

Now the problem is: Given P_k and P_m, also $P_k * P_m \leq P_{km} \leq P_k$, Please evaluate P_{km} . The problem couldn't be solved with the conditions in mathematics. But in fact, there is a lot of information without accurate mathematic formula which be omitted. In this paper it offered a method by association analysis to confirm the formula.

Let parameter a be the probability which A_k and A_m are total correlation, and parameter b for total non-correlation. a+b=1, 0 < a, b < 1, then P_{km} can be defined as formula (2) below:

$$P_{km} = a * P_k + b * P_k * P_m \quad (2)$$

There are a lot methods to confirm the value of parameter a and b, in this paper it provide a way to define "a" and " b"by association analysis.

3.2 Confirm the Parameter "a", "b" by association analysis

There are a series of criterion about environment. The most ingredients of the pollution can be confirmed based on the source. So we can consider the criterion as a referenced list and the list needed to find the correlation as a comparison list. Then we'll get the correlation coefficient which is the parameter "a"in our formula (2), and b=1-a. The details as below:

Let $S=\{S_1, S_2, \ldots S_m\}$ be the value list of item A_m, $S_1, S_2, \ldots S_m$ are sample extracted from the DB and $X=\{ X_1, X_2, \ldots X_m \}$be the value list for item A_k , $X_1, X_2, \ldots X_m$ are sample extracted from the DB.

$$a_{km} = \frac{\min_i \min_k \Delta_i(k) + \rho \max_i \max_k \Delta_i(k)}{\Delta_i(k) + \rho \max_i \max_k \Delta_i(k)} \quad (3)$$

in the formula (3), a_{km} is the correlation coefficient of item A_m and A_k , $\Delta_i(k) = |S_i - X_i|$, p is the distinguished coefficient which set by users, usually , $p \in (0, 1)$.

We can use the formula (3) to calculate the correlation coefficient of any items.

3.3 Description of algorithm

1. Create a new array PFA[n], the original value for each element is 0;

scanning the database, calculating the probability of each itemset $A_1, A_2, \ldots, A_n$ respectively and marked by $P_1, P_2, \ldots, P_n$.

Let each element of the array PFA[1], PFA[2], …PFA[n] be the $P_1, P_2, \ldots, P_n$ .which refer to the probability of each itemset $A_1, A_2, \ldots, A_n$.

The process for calculating the probability of A_i appearing .

(a)If it is not the end of database ,then read and get the recorder;

(b)If there is an item A_i in the recorder then PFA[i] = PFA[i]+1;

(c)Repeat the above procedure until the end of the database then PFA[i]=PFA[i]/(number of records in the database).

Repeat step (a)(b)(c)to calculate the probability of itemset A_1,A_2, …, A_n appearing.

2.Set a minimum value V_1 for the probability of A_i appearing, if the probability of A_i appearing PFA[i] is larger than V_1 then itemset A_i is a frequent 1-itemset. so, you get some frequent 1-itemset, let "m" be the number, and PFA[1],PFA[j],…PFA[m] be the probability of 1-itemset appearing respectively.

3 Due to the probability of 1-itemset appearing PFA[1], PFA[j], …PFA[m] ,base on the formula (2),then the probability of any two itemset appeared in one recorder can be evaluated. Set a minimum value V_2 for the probability of A_i and A_j appeared synchronously in one record, if the probability is larger than V_2 then itemset A_iA_j is a candidate frequent 2-itemset.otherwise set the value of the probability is zero to predigest the later calculation. Let the element of array PUA_2 [i] record the value of candidate frequent 2-itemsets.

Let V_2 be the minimum probability for candidate frequent 2-itemset, and V_3 for candidate frequent 3-itemset, V_{k-1} for candidate frequent (k-1)-itemset

Set minimum probability V_{k-1} :

V_{k-1}=a*min(PFA_{k-1}[1],PFA_{k-1}[2],…PFA_{k-1}[m])+b*min(PFA_{k-1}[1],PFA_{k-1}[2],…PFA_{k-1}[m])*

max(PFA_{k-1}[1],PFA_{k-1}[2],…PFA_{k-1}[m])

4. Recur the above step 1.2.3., from k=2 to n to calculate the probability of k-itemsets A_1,A_2,… ,A_k appearing in one recorder;

5.Scan the database another time to calculate the support of the candidate frequent itemsets which is the result of step 4.

(a)Create a new array DMA[m] with each element's original value is zero.(m =number of candidate frequent itemsets);

(b)Read and get the recorder of the database until the end of the database.

(c)If there are itemsets A_i,A_j, … ,A_k in any recorder synchronously and A_i≠0,A_j≠0…A_k≠0; then the support for A_iA_j … A_k DMA[k]= DMA[k]+1.

recur the above step(b)(c)to calculate the actual support of every candidate frequent itemsets until the end of the database.

6. Find out the frequent itemsets from the candidate frequent itemsets. If DMA[k] is larger than the minimum support which the user set, then output the frequent itemsets.

Step 5.,6. is used to confirm the probability and support of the candidate frequent itemsets which come out by the method of probability evaluation whether satisfy the request of the user.

7. Output the association rule from the result of the step 6.

3.4 Explanation and simulation of the algorithm

The transaction database of some suboffice in AllElectronics is choosed to compare the efficiency of our algorithm and Apriori. The detailed data is presented in fig1(A). In the algorithm the paper providing the process of finding frequent itemsets include 3 steps:

Firstly it scans the database to come out the probability of frequent 1-itemsets;

Then evaluates the probability of candidate frequent 2-itemsets,3-itemsets …m-itemsets, based on the probability of frequent 1-itemsets;

Last it scans the database another time to confirm the frequent itemsets from the candidate frequent itemsets.

There is a database with 9 transaction and 5 itemsets. The parameter "a","b"is 5/9 and 4/9 respectively in the simulation of the algorithm.

TID	ITEMSETS

Calculated the probability of each itemset to find out frequent 1-itemsets at the support was 2/9

T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I5
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

(1A)

Itemsets	Probability
I1	6/9
I2	7/9
I3	5/9
I4	2/9
I5	3/9

(1B)

From frequent C1(fig 1B) to evaluate the probability of candidate frequent 2-itemstes(fig 1C)

Itemsets	Probability
{I1,I2}	438/729
{I1,I3}	345/729
{I1,I4}	138/729
{I1,I5}	207/729
{I2,I3}	365/729
{I2,I4}	146/729
{I2,I5}	219/729
{I3,I4}	130/729
{I3,I5}	195/729
{I4,I5}	114/729

(1C)

Set a=5/9,b=4/9;

Threshold V_2=(5*(2/9)+4*(2/9)*(7/9))/9

=146/729

Get the candidate frequent 2-itemsets below(1D):

2-itemsets	Probability
{I1,I2}	438/729
{I1,I3}	345/729
{I1,I5}	207/729
{I2,I3}	365/729
{I2,I4}	146/729
{I2,I5}	219/729
{I3,I5}	195/729

(1D)

From frequent C2 to evaluate the probability of candidate frequent 3-itemstes(fig 1E)

Itemset	Probabilty
{I1,I2,I3}	69*345/59049
{I1,I2,I4}	73*138/59049
{I1,I2,I5}	73*207/59049
{I1,I3,I4}	69*130/59049
{I1,I3,I5}	69*195/59049
{I1,I4,I5}	69*114/59049
{I4,I2,I3}	73*130/59049
{I3,I4,I5}	65*114/59049
{I2,I4,I5}	73*114/59049
{I2,I3,I5}	73*195/59049

(1E)

Set a=5/9,b=4/9;

Threshold V_3=(5*(146/729)+4*(146/729)*(7/9))/9

=73*146/59049

Get the candidate frequent 3-itemsets below(1F):

3-itemsets	Probability
{I1,I2,I3}	69*345/59049
{I1,I2,I5}	73*138/59049
{I2,I3,I5}	73*195/59049

Fig1 The process of producing candidate itemsets

4 Comparison for the algorithms

It was compared between our algorithm and apriori in this section ,the number of the candidate frequent itemsets and times of scanning the database. Which also were the linchpin of the efficiency in a algorithm. The algorithm in this paper ahead apriori in reducing the times of scanning the database. It would scan the database k times to find out frequent k-itemsets in apriori while only 2 times in our algorithm by putting forward a concept of candidate frequent itemset..

In the table1 below, it listed the frequent 1-itemsets, 2-itemsets, 3-itemsets for both of the two algorithm respectively.

Table 1 The Comparison of the frequent itemsets

(candidate)frequent itemsets	Algorithm Apriori	Algorithm in this paper
1-itemsets	{I1,I2,I3,I4,I5}	{I1,I2,I3,I4,I5}
2-itemsets	{I1,I2},{I1,I3},{I1,I5},{I2,I3},{I2,I4}, {I2,I5}	{I1,I2},{I1,I3},{I1,I5},{I2,I3},{I2,I4}, {I2,I5},{I3,I5}
3-itemsets	{I1,I2,I3},{I1,I2,I5}	{I1,I2,I3},{I1,I2,I5},{I2,I3,I5}

From the table 1,we knew that there were more candidate frequent itemsets in the algorithm this paper provided than that in the algorithm apriori. It make sure that it wouldn't miss any frequent itemsets.

5 Experiment and the analysis of the results

In this section, we report our experimental results. To validate the algorithm presented in the paper ,A database for recording the update of the spatial database are used in the experiment. There are 10 items in the database, such as I1for mender , I2 for DEPT of the mender, I3 for layer of the tower, I4 for layer of the road , I5 for layer of the polluter area, I6 for layer of line, I7 for layer of water, I8 for layer of building, I9 for layer of thunder and I10 for layer plant. The purpose of the experiment was mining the association rules between the mender and other spatial layer. All experiments were conducted on a PC with an Intel Pentium III 800MHz CPU and 128M main memory, running Microsoft Windows2000. All programs were coded in Microsoft Visual C++ 6.0. The experiments were conducted on real data sets by the algorithm described in this paper and apriori. Table 2 below listed the candidate frequent 1-itemsets,2-itemsets,3-itemsets of our algorithm and frequent 1-itemsets , 2-itemsets, 3-itemsets of algorithm apriori. We report here only results on some typical data sets,with 10 itesmsets and between 5000 tuples.

Table 2 The comparison of the frequent itemset in the experiments

Frequent itemsets for association rules	Algorithm Apriori	Algorithm in this paper
1-itemsets	{I1,I2,I4,I5,I7,I8,I9,I10}	{I1,I2,I4,I5,I7,I8,I9,I10}
2-itemsets	{I1,I2},{I1,I5},{I1,I8},{I1,I9}, {I1,I10},{I2,I5},{I2,I8},{I2,I9}, {I1,I10},{I5,I8},{I5,I10}	{I1,I2},{I1,I5},{I1,I7},{I1,I8},{I1,I9},{I1,I10},{I2,I5},{I1,I7},{I2,I8}, {I2,I9},{I1,I10},{I5,I8},{I5,I10}, {I8,I10}

3-itemsets	{I1,I2,I5},{I1,I2,I8},{I1,I2,I10}, {I1,I5,I8},{I1,I5,I10},{I1,I5,I10}, {I2,I5,I8}	{I1,I2,I5},{I1,I2,I8},{I1,I2,I9},{I1, I2,I10},{I1,I5,I8},{I1,I5,I10},{I1,I5 ,I10},{I2,I5,I8},{I5,I8,I10}
4-itemsets	{I1, I2, I5,I8}	{I1, I2, I5, I8},{I1, I2, I5, I10}

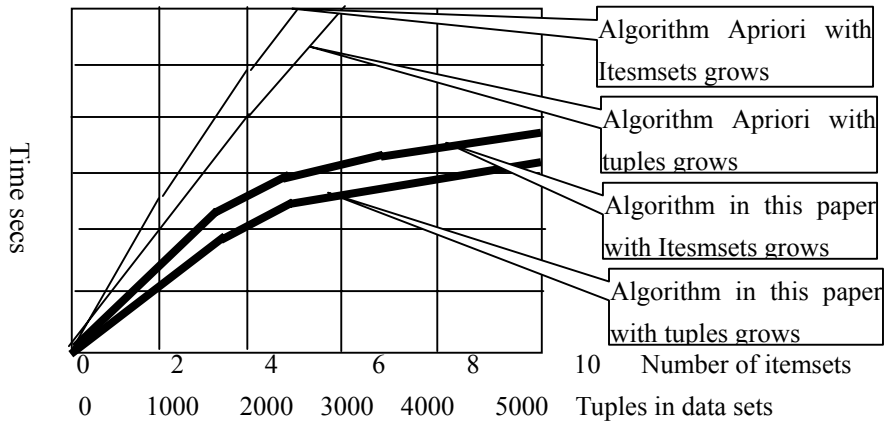

Fig2 Execution time

The first data set we use has 1000 tuples, then increase 1000 tuples every time. We test the execution time of the algorithms with respect to number of tuples and itemsets. Fig 2 shows that: With tuples from 0 to 5000, when the number of tuples is small, both algorithms have similar performance. However, as the number of tuples grows, the algorithm in this paper takes effect. It keeps the runtime low. In contrast, the algorithm apriori does not scale well under large number of tuples. Figure 2 also shows the execution time of both algorithms with respect to the itemsets increased. As the number of itemsets goes up , the runtime of both algorithms has increases and the algorithm in this paper grows slower than algorithm apriori.

Similar statements can be made about Fig 2, where the number of tuples and itemsets varies from 0 to 5000, 0 to 10 respectively. Algorithm in this paper present a smoother increasing of runtime than that of algorithm apriori.

6. Conclusions

Based on the previous studies on algorithm apriori, we proposed a method for mining association rules in large databases with association analysis and probability evaluating. The method developed in this paper explores efficient mining of association rules by probability evaluating . First it scans the database to filter frequent 1-itemsets and gets their probability respectively. Then it gets the candidate frequent 2-itemset,3-itemsets up to n-itemsets by evaluating their probability on formula (2) in the paper and the result of the first step. Last it scans the database for another time to refine the candidate frequent itemsets to the frequent itemsets.

Data mining in association rule is an important research topic and apriori is the core algorithm in mining association rule. We propose a method that enhances the efficient of algorithm by evaluating the probability of candidate frequent itemsets. It shortens the runtime of algorithm by reducing the times of scanning database. A formula is provided in this paper. We also present an efficient method for confirming the parameter “a”,”b” in formula (2) by association analysis. If the parameter “a”,”b”are unseemliness, it will omit some association rules which users are interested in. The solution for this problem is multi enactment for parameter “a” and “b”,then choose the best. The method of grey relational analysis is widely used in the association analysis of environment databases. There are a lot of other methods to confirm the parameter “a”,”b”. In this paper we implemented the algorithm and the performed experiments show significant benefits for different number of the itemsets and tuples in the database.

References

[1] S. Agarwal, R. Agrawal, P. M. Deshpande, A. Gupta, J. F.Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. *VLDB'96*.

[2] Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. *KDD'99*.

[3] K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cubes. *SIGMOD'99*.

[4] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26:65–74, 1997.

[5] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. *KDD'99*.

[6] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J. D. Ullman. Computing iceberg queries efficiently.*VLDB'98*.

[7] G. Grahne, L. Lakshmanan, and X. Wang. Efficient mining of constrained correlated sets. *ICDE'99*.

[8] J. Han, J. Pei, G. Dong, and K. Wang. Efficient computation of iceberg cubes with complex measures. *ACMSIGMOD'01*.

[9]R Agrawal,T Imielinski,A Swami. Mining associations between sets of items in massive databases. In:Proc of the1 993 ACM- SIGMOD Int'l Conf on Management of Data. NewYork: ACM Press,1993 .
2 0 7-2 1 6

[10]R Srikant,Q Vu,R Agrawal. Mining association rules with item constraints. In:Proc of the 1997Third Int'l Conf on Knowledge Discovery in Databases and Data Mining. New port Beach, California: AAAI Press,1997. 67- 73

[11] S Brin,R Motwani,JUllman etal. Dynamic itemset counting and implication rules for market basket data. In:Proc of the1 997ACM- SIGMOD Int'l Conf on the Management of Data. New York: ACM Press,1 997. 2 55- 2 64.