A Method to Optimize Apriori Algorithm for Frequent Items Mining

Ke Zhang[1,2], Jianhuan Liu[1], Yi Chai[1,2], Jiayi Zhou[1], Yi Li[1]
[1]Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education,
College of Automation, Chongqing University, Chongqing 400030, P.R.China
[2]State Key Laboratory of Power Transmission Equipment & System Security and New Technology,
College of Automation, Chongqing University, Chongqing 400030, P.R.China
Email:smeta@163.com, jianhuan0622@csu.edu.cn, chaiyi@cqu.edu.cn, zhoujiayi0619@gmail.com,chuiguodewind@qq.com

***Abstract*—This paper studies the fundamental problems of mining association rules. Based on the summary of classical mining algorithm and the inherent defects of Apriori algorithm, some related improvements are researched. In order to avoid scanning the database multiple times, the database mapping method is changed in this research. Meanwhile, after the support of candidate item sets is get, each candidate item set should be determined whether it is a frequent item set or not based on the prior knowledge of Apriori algorithm. If the candidate item sets generated by the element of the existing frequent item sets are certainly not frequent item sets, the element is not necessary to connect with others, which leads to an optimized connecting step. Lastly, for Apriori algorithm, the intersection operation is introduced to address the disadvantages that it takes many time costs to match with candidate item sets and transaction pattern. Through these improvement strategies, the optimized algorithm is presented and its advantages are explained in theory. And furthermore, to verify the effectiveness, the optimized algorithm has been applied to the floating car data. The experiments results show a shorter execution time and a higher efficiency under different supports and confident levels.**

Keywords-Data Mining, Association Rules, Apriori Algorithm, Optimization, Floating Car Data

I. INTRODUCTION

Data Mining focuses on a process of extracting knowledge from the original data which has some particular form[1]. From the definition, data is the source of forming knowledge. The process of extracting knowledge is neither universal criteria nor new scientific theories. All potential knowledge is discovered based on prior information and specific field. Furthermore, it is necessary to use natural language to express knowledge discovery with a particular precondition and constraints. In the commercial trade, staff rearranged the shelves after data analysis. Generally, the above-mentioned data analysis mainly includes previous sales record analysis and customer behavior and habits analysis. Then, a success is acquired through transforming the marketing strategy, such as the famous "beer and diapers".

Up to now, a lot of new concepts and methods have been generated in data mining field. A higher level characteristic of knowledge is found based on detail features of data. The current studies focus on classification, clustering, association rules mining, sequential pattern discovery and prediction and abnormal findings[1], etc. Especially, association rules mining refers to finding the correlation between some transactions. If there is some correlation between two transactions, one can be predicted by another transaction.

Association rules mining is used to find frequent pattern and correlation existed in item sets through data processing, analysis, synthesis and inference[2]. Association rules mining has achieved a very great application effect in business and other fields and it has become a research hotpot. Through generating frequent appearance of data, new patterns and hidden knowledge can be found. In the field of association rules mining, Apriori algorithm is most popular. It is based on frequent item sets generation algorithm. However, there are also many different aspects, such as searching strategies, scanning databases, data structure and so on[3,4]. Apriori algorithm is based on breadth-first search and its data structure is simple, clear and easy to understand. But, the application of Apriori algorithm needs to scan database many times which leads to a great overhead. Although Apriori algorithm itself has made some optimization, the time and space consumption is big and efficiency is not high. Overall speaking, it is difficult to avoid following defects[4,5].

(1) Produce a large number of candidate items. When generating all the candidate k-item sets C_k using frequent (k-1)-item sets, multiple connections are required and the number of comparison is large. If the set of a frequent item set generated by a candidate item set is L_k , the time complexity of connection constraints comparison is $O(k*x^2)$. From the previous analysis, the candidate 2-item sets are often larger, namely the value of x is greater. Assuming that each transaction mold is 1, the mold of generated candidate 1-item sets is 10^4 and the mold of generated candidate 2-item sets will reach to 10^7 orders of magnitude.

(2) Scan the database repeatedly. To obtain the support of every candidate k-item set, the database should be scanned multiple times. When the value of k is 1, 2, 3 ... , n, the database is scanned at least n times. Actually, the mining mold is usually larger in practical application, and scanning the database repeatedly will bring time overhead.

(3) Spend too much time when matching candidate item sets and transaction pattern. For $c \in C_k$, every (k-1)-item sets in c should be tested whether it is in frequent (k-1)-item sets L_{k-1} or not. Under the best circumstances, only scanning the L_{k-1} once can reach the requirement. But under the worst circumstance, the requirement will be reached until scanning L_{k-1} k times. And under the average conditions, for $c \in C_k$, the times of scanning L_{k-1} is $k*|L_{k-1}|/2$, where $|L_{k-1}|$ represents the mold of L_{k-1}. So, for all candidate k-item sets C_k, the needed

times of scanning is $C_k*k*|L_{k-1}|/2$. Obviously, this operation brings a lot of time consumption.

Based on the above deficiencies of Apriori Algorithm, this paper concerns that algorithm may generate a large number of candidate item sets and multiple scanning times are needed to match candidate item sets and transaction pattern. For these deficiencies, this paper presents three optical solutions. The first one is used to change the mapping method of database. The second one is used to optimize the connection steps. The set intersection operator is introduced in the last solution. The combination of these strategies gives a improved algorithm.

II. Related Researches

Domestic and foreign scholars in the association rule mining has invested a lot of energy to do researches. They mostly focus on the traditional algorithm improvement. In the aspects of improving the efficiency of mining rule algorithm, many researchers have made unremitting efforts.

Agrawa and other scholars analyzed the basket firstly in 1993 and given the association rules mining in data mining. Furthermore, the classical Apriori Algorithm was been proposed in the following year[6]. Subsequently, many researchers studied the problem of mining association rules in algorithm optimization. Savasere *et al* proposed a method based on Partition[7]. Park *et al* proposed a method based on Hash[8]. Mannila and Toivonen proposed a method based on Sampling[9,10]. After the introduction of these methods and techniques, J. Han proposed an inventive solution that is named FP-Growth Algorithm[11]. It introduced a new data structure. More importantly, this method just scans the database twice and does not generate candidate item sets.

The foreign association rule mining concepts and ideas were introduced by Chinese Academy of Science[12]. A new temporal association rules mining algorithm based on Apriori Algorithm was proposed and the application of association rules was promoted. Liu *et al* proposed OpportuneProject Algorithm[13]. Depending on the characteristics of local data sets, it determined using virtual projection based on tree or using non-filtered projection based on array dynamically. This algorithm mainly used a depth-first search method, but it also used the breadth-first to establish the top of tree when necessary. C. Li *et al* introduced matrix to improve Apriori algorithm, which reduces the computing costs and improves operational efficiency significantly[14]. For the changing support, how to maintain the maximum frequent item sets is a problem. To address this problem, Y. Shao *et al* proposed a fast update algorithm to adjust the support in the mining process[15]. Based on this research, L. Tang proposed an improved incremental update algorithm of association rules. It can simultaneously solves the problem of minimum support and update the database[16]. For inefficient Apriori algorithm, G. X. Cui *et al* improved the vertical layout of database and connections and pruning strategies in the mining process[17]. B. Shen *et al* put forward a new dynamical association rule and its support vector and confidence vector coincide with the classical definition, reflecting a better dynamical information over time[18]. On this basis, they further proposed two dynamical association rules mining algorithm, pointing out that the algorithm is fit for high-density mass data mining. In data mining filed, redundancy of the original rule sets is difficulties for users to understand. For this problem, Y. Chen focused on the relationship between the original rule sets and regulation set representation[19].

III. A Method to Improve Apriori

A. Improved Algorithm Ideas

In Apriori algorithm, the support of item sets is determined based on transaction identifier when scanning the transaction database. With increments of k , the database will be scanned for multiple times. In this paper, a method to improve the handing of database has been presented. Through using the new data mapping, the database is scanned only once. And following frequent item sets mining simply are obtained by scanning C_k and L_k. This process addressed the problem of multiple scannings.

With the support of each candidate item sets to determine frequent item sets, if a candidate item set generated by elements of L_{k-1} is definitely not a frequent item set, this element is not considered.

The transaction pattern matching requires a lot of time overhead in Apriori algorithm, so some knowledge about set is introduced. The new transaction identifier is generated through carrying intersection operator to transaction identifier.

B. Improved Algorithm Descriptions

- Optimize mapping method of database

In Apriori algorithm, the database is handled by using class that contains two linear tables. All of the items in the transaction are mapped by transaction identifier. In improved transaction database, a new data structure is used to store mapping approaches[20]. In this method, a linear table is used to store items and another linear table is used to store corresponding TID list which consists of all supported transaction identifier.

After this mapping process, it is easy to calculate the amount of transaction in candidate k-item sets because the support of item sets is equivalent to the mode of this TID list. Through mapping transformation, all of transaction identifier of candidate 1-item sets can be obtained after scanning database firstly. The following candidate item sets can be obtained by scanning C_k and L_k, resulting needed to scan database only once.

- Optimize connection steps

Assuming the transaction database is $\{\{a, b\}, \{a, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}\}$ and minimum support is 2, frequent 2-item sets is $\{\{a, b\}, \{a, c\}, \{a, d\}, \{b, d\}, \{c, d\}\}$. When generating candidate 3-item sets, $\{a, b, c\}$ will be generated. But obviously $\{a, b, c\}$ is not a item set of transaction database. In Apriori algorithm, candidate item sets are generated first and then are pruned, but some of

these candidate item sets are not items in the transaction database.

After support of each candidate item set is obtained, it is necessary to determine whether the support met to the minimum support. Since the candidate item sets generated by element of L_{k-1} are surely not frequent item sets, this element does not need to participate in connections. For example, if known L_3 = {{1, 2, 3}, {1, 2, 4}, {1, 3, 4}, {2, 3, 4}, {2, 3, 5}}, ask L_4. Here the minimum support is set as 3. Because $L_3(1) = 3$, $L_3(2) = 4$, $L_3(3) = 4$, $L_3(4) = 3$ and $L_3(5) = 1 < 3$, an optimized L_3' = {{1, 2, 3}, {1, 2, 4}, {1, 3, 4}, {2, 3, 4}} can be obtained. Furthermore, C_4 = {1, 2, 3, 4} and L_4 = {1, 2, 3, 4} can also be obtained easily. Conversely, candidate 4-item sets is C_4 = {{1, 2, 3, 4}, {2, 3, 4, 5}} based on classical Apriori algorithm. From the results point of view, it is clear that classical Apriori algorithm can generate more redundant candidate item.

- Introduce set intersection operation

When candidate *k*-item sets are generated by frequent (*k*-1)-item sets, the intersection operation can be carried on corresponding transaction number sets to generate new transaction number sets. And then, support of candidate *k*-item sets can be acquired through adding up transaction mold of new transaction number sets.

C. Improved Algorithm Implementations

- OApriori Algorithm

Input：transaction database D，minimum support *min_sup*
Output: frequent k-item set L_k
(1) L_1 = *find_frequent_1_itemset*(D, *min_sup*);
(2) *for*($k = 2$; $L_{k-1} \neq \Phi$; $k++$){
(3) $L_{k-1}' = L_{k-1}$;
(4) *for each item* $x \in L_{k-1}$
(5) *for each field* $f \in x$
(6) { $f.count$ ++;
(7) $L_f[f.count] = x$; }
(8) *delete*(L_{k-1}', $L_f[f.count] < k$-1);
(9) *for each item* $x, y \in L_{k-1}'$
(10) *if* ($x_1 = y_1$ && $x_2 = y_2$ && ... $x_{k-2} = y_{k-2}$ && $x_{k-1} < y_{k-1}$)
(11) $n = 0$;
(12) *for*($i = 1, j = 1$; $i \leq |T_x|, j \leq |T_y|$)
(13) *if* ($T_x[i] < T_y[j]$) i ++;
(14) *else if* ($T_x[i] > T_y[j]$) j++;
(15) *else* {
(16) n ++;
(17) $T_{x \otimes y} = T_{x \otimes y}\{T_x[i]\}$
(18) i++; j++;}
(19) *if* ($n \geq N$**min_sup*)
(20) { $T'_{x \square y} = T_{x \square y}$; $L_k = L_k \{x \square y\}$ }
(21) r*eturn* $L = \cup L_k$;

- Frequent 1-item sets mining algorithm

Input: transaction database D, minimum support *min_sup*;
Output: L_1;
procedure find_frequent_1_itemset (D, *min_sup*);
(1) C_1 = *get_item*(D);
(2) *for each transaction* $t \in D$
(3) { N++;
(4) *for each item* $x \square t$
(5) { $x.count$ ++; $T_x[x.count] = t_{id}$;}
(6) }
(7) $L_1 = \{x \mid x \square C_1$ && $\{x.count \geq N$**min_sup*$\}\}$
(8) return L_1;

IV. Experiments and Results

This section provides an example of a transactional database application (shown in Tab. 1, *min_sup* = 60%) and floating traffic data. And OApriori algorithm is tested by these two examples, validating the theoretical and practical effectiveness.

TABLE I. Original Transaction Database

TID	Item sets	TID	Item sets
1	*a c d f g i m p*	2	*a b c f l m o*
3	*b f h j o*	4	*b c k s p*
5	*a c e f l m n p*		

A. Implementation Process of OApriori

Firstly, class consisted by two linear tables is handled to form a new transaction table (shown in Tab. 2). After this operation, original transaction database will not be scanned any more.

TABLE II. Transaction Table after Remapping

Items	TID	Items	TID
a	{1,2,5}	*j*	{3}
b	{2,3,4}	*k*	{4}
c	{1,2,4,5}	*l*	{2,5}
d	{1}	*m*	{1,2,5}
e	{5}	*n*	{5}
f	{1,2,3 ,5}	*o*	{2,3}
g	{1}	*p*	{1,4,5}
h	{3}	*s*	{4}
i	{1}		

Through remapping, an improved transaction table is formed. This process only requires scanning the original sample transaction database once and the transaction identifier of each item has been recorded into the new table. Because support of some a item is equivalent to the mold of transaction identifier list of this item, it is easy to get transaction number.

Adding up the transaction number of each item easily get the candidate 1-item sets (Shown in Tab. 3). Through pruning the candidate item sets, frequent 1-item sets is acquired (Shown in Tab. 4).

Through a similar operation, candidate 2-item sets and frequent 2-item sets can be obtained (Shown in Tab. 5 and Tab. 6).

TABLE III. CANDIDATE 1-ITEM SETS

Items	TID	Support	Items	TID	Support
a	{1,2,5}	3	*j*	{3}	1
b	{2,3,4}	3	*k*	{4}	1
c	{1,2,4,5}	4	*l*	{2,5}	2
d	{1}	1	*m*	{1,2,5}	3
e	{5}	1	*n*	{5}	1
f	{1,2,3,5}	4	*o*	{2,3}	2
g	{1}	1	*p*	{1,4,5}	3
h	{3}	1	*s*	{4}	1
i	{1}	1			

TABLE IV. FREQUENT 1-ITEM SETS

Items	TID	Support	Items	TID	Support
a	{1,2,5}	3	*f*	{1,2,3,5}	4
b	{2,3,4}	3	*m*	{1,2,5}	3
c	{1,2,4,5}	4	*p*	{1,4,5}	3

TABLE V. CANDIDATE 2-ITEM SETS

Items	TID	Support	Items	TID	Support
{*a,b*}	{2}	1	{*b,p*}	{4}	1
{*a,c*}	{1,2,5}	3	{*c,f*}	{1,2,5}	3
{*a,f*}	{1,2,5}	3	{*c,m*}	{1,2,5}	3
{*a,m*}	{1,2,5}	3	{*c,p*}	{1,4,5}	3
{*a,p*}	{1,5}	2	{*f,m*}	{1,2,5}	3
{*b,c*}	{2,4}	2	{*f,p*}	{1,5}	2
{*b,f*}	{2,3}	2	{*m,p*}	{1,5}	2
{*b,m*}	{2}	1			

TABLE VI. FREQUENT 2-ITEM SETS

Items	TID	Support	Items	TID	Support
{*a,c*}	{1,2,5}	3	{*c,m*}	{1,2,5}	3
{*a,f*}	{1,2,5}	3	{*c,p*}	{1,4,5}	3
{*a,m*}	{1,2,5}	3	{*f,m*}	{1,2,5}	3
{*c,f*}	{1,2,5}	3			

TABLE VII. FREQUENT 3-ITEM SETS

Items	TID	Support	Items	TID	Support
{*a,c,f*}	{1,2,5}	3	{*a,f,m*}	{1,2,5}	3
{*a,c,m*}	{1,2,5}	3	{*c,f,m*}	{1,2,5}	3

TABLE VIII. FREQUENT 4-ITEM SETS

Items	TID	Support
{*a,c,f,m*}	{1,2,5}	3

In the process of generating candidate 2-item sets, taking into account that any candidate item set generated by {*c*, *p*} is surely not a frequent item set, so {*c*, *p*} will be deleted directly. Other frequent 2-item sets will be carried on intersection operation to generate candidate 3-item sets and calculate the transaction support. Obviously, candidate 3-item sets is the same to the frequent 3-item sets(Shown in Tab. 7). As the same operation, frequent 4-item sets can be easily acquired (Shown in Tab. 8).

As can be seen from the analysis above, OApriori algorithm inherits the advantages of Apriori algorithm. Meanwhile, OApriori algorithm significantly improves the performance and efficiency of the original Apriori algorithm.

B. Urban Traffic Information Mining

In this paper, association rule mining algorithm is applied to mine urban traffic information. Based on floating car GPS data provided by monitoring center in Chongqing, urban traffic information is mined through Apriori algorithm and OApriori algorithm under the condition of different support and different confident level. Especially, the execution time and efficiency are recorded in above process, leading a better result compared with Apriori.

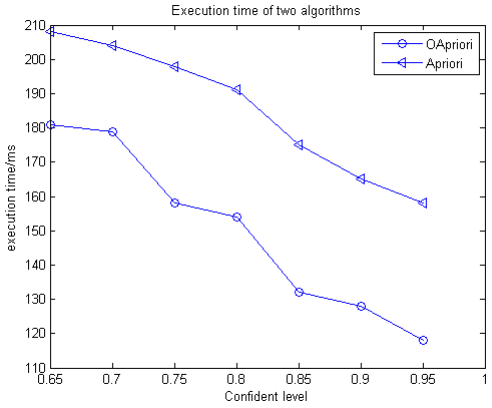

Figure 1. Execution Time of Two Algorithms under Different Confidence.

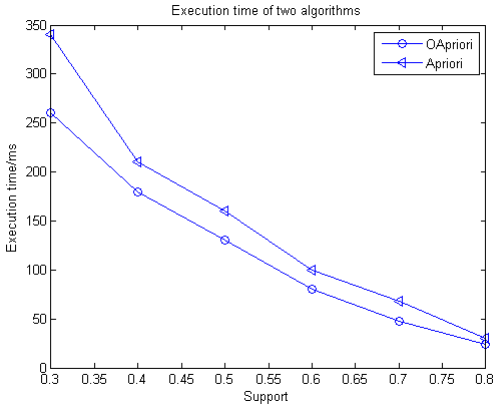

Figure 2. Execution Time of Two Algorithms under Different Support

If support is keeping as 40%, this research recorded the execution time varied with confident level. As clearly shown in Fig. 1, with increasing of confident level, the two algorithms tends to decrease the execution time. But execution time of OApriori algorithm is significantly less

than original Apriori algorithm, reflecting the improvement of time efficiency.

If confident level is keeping as 80%, this research also recorded the execution time varied with support. As obviously shown in Fig. 2, a same result is acquired. With increasing of support, the two algorithms always tends to decrease the execution time and OApriori algorithm outperforms Apriori algorithm.

V. CONCLUSIONS

This paper describes the association rule mining and carries out a research on the association rule mining algorithm. For the inherent defect of Apriori algorithm, an optimized algorithm is presented and realized. In order to avoid scanning the database multiple times, the database mapping method is changed in this research. Meanwhile, after the support of candidate item sets is get, each candidate item set should be determined whether it is a frequent item set or not based on the prior knowledge of Apriori algorithm. If the candidate item sets generated by the element of the existing frequent item sets are certainly not frequent item sets, the element is not necessary to connect with others, which leads to an optimized connecting step. Lastly, for Apriori algorithm, the intersection operation is introduced to address the disadvantages that it takes many time costs to match with candidate item sets and transaction pattern. Through these improvement strategies, the optimized algorithm is presented and its advantages are explained in theory.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China (Grant No.61203084 and 61374135) and the National Natural Science Foundation of Chongqing China (cstc2011jjA40013).

REFERENCES

[1] Mao, G. J., Data Mining Theory and Algorithm[M]. Beijing: *Tsinghua University Press*, 2007.

[2] Li, Q., Data Mining Association Analysis Algorithm[D]. Harbin: Harbin Engineering University, 2010. (in Chinese)

[3] Lin, Y. Q., A Review of Association Rules Mining Algorithm[J]. *Software Guide*, Vol. 11, pp. 27-29, 2012.

[4] Chi, X., & Fang, Z. W. Review of association rule mining algorithm in data mining. In *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on,* 2011, pp. 512-516.

[5] Bednar, P., Babic, F., Albert, F., Paralic, J., & Bartok, J. Design and implementation of local data mining model for short-term fog prediction at the airport. In *Applied Machine Intelligence and Informatics (SAMI), 2011 IEEE 9th International Symposium on,* 2011, pp. 349-353.

[6] Agrawal, R., Imieliński, T., & Swami, A. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record.* 1993, pp. 207-216.

[7] Savasere, A., Omiecinski, E. R., & Navathe, S. B. (1995). An efficient algorithm for mining association rules in large databases.

[8] Park, J. S., Chen, M. S., & Yu, P. S. An effective hash-based algorithm for mining association rules. *ACM,* Vol. 24, pp. 175-186. 1995.

[9] Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., & Verkamo, A. I. Finding interesting rules from large sets of discovered association rules. In *Proceedings of the third international conference on Information and knowledge management*, 1994, pp. 401-407.

[10] Toivonen, H.. Sampling large databases for association rules. In *VLDB*, Vol. 96, pp. 134-145, 1996.

[11] Han, J., Pei, J., & Yin, Y. Mining frequent patterns without candidate generation. In *ACM SIGMOD Record*, 2000, Vol. 29, pp. 1-12.

[12] Ouyang, W. M., Cai, Q. S., Discovery of Association Rules with Temporal Constraint in Databases. *Journal of Software*, Vol. 10, pp. 527-532, 1999.

[13] Liu, J. Q., Massive Data Mining Technology Research. Hangzhou: Zhejiang University PhD thesis, 2003. (in Chinese)

[14] Li, C., Yu, Z. P., Improved Apriori Algorithm Based on Matrix. *Computer Engineering*, Vol. 32, pp. 68-71, 2006.

[15] Shao, Y., Chen, B., Fast Update Algorithm for Association Rule. *Computer Engineering*, Vol. 35, pp. 62-65, 2009.

[16] Tang, L., Jiang, H., An Improved Incremental Updating Algorithm for Association Rules. *Computer Applications and Software*, Vol. 29, pp. 246-248, 2012.

[17] Cui, G. X., Li, L., Research and Improvement on Apriori Algorithm of Association Rule Mining. *Journal of Computer Application*, Vol. 30, pp. 2952-2955, 2010.

[18] Shen, B., Yao, M., A New Kind of Dynamic Association Rule and its Mining Algorithms. *Control and Decision*, Vol. 24, pp. 1310-1315, 2009.

[19] Chen, Y., Shan, S. Q., Minimum-redundant and Lossless Association Rule-set Representation. A*CTA AUTOMATIC SINICA*, Vol. 34, pp. 1490-1496, 2009.

[20] Liu, B. Z., Improved Apriori Mining Frequent Items Algorithm. *Application Research of Computers*, Vol. 29, pp. 475-477, 2012.