

# Association Rules Algorithm Based on the Intersection

Xuegang Chen\* and Jie Xiao

College of Software and Communication Engineering, Xiangnan University, Chenzhou, 423000, China

**Abstract:** Mining association rules in the database is one of important study in data mining research. Traditional association rules consist of some redundant information, and need scan database many times and generate lots of candidate item sets. Aiming at low efficiency in association rules mining using traditional methods, this paper proposes the algorithm (ISMFP), which is based on intersection for mining the maximum frequent patterns. Firstly, applying the intersection theory of mathematics, put forwards a number of concepts and definitions. Then gives the process of association rules mining, and analyzes its performance. After that, the example describes the process of implementation of the algorithm. Finally, the experimental results show that the algorithm ISMFP is efficient on mining frequent patterns, especially there exists low threshold of support degree or long patterns.

**Keywords:** Association rules, data mining, intersection, maximum frequent pattern.

## 1. INTRODUCTION

T Agrawal firstly proposes the apriori algorithm for mining association rules from the customer transactional databases in 1993 [1], and the algorithm uses an iterative method of layer by layer search, and the connection-pruning method is used from the bottom to up. But it takes lots of time to connect and scan the database, then filter the candidate terms generated after the connection. At the same time, it is only suitable for mining the shorter patterns. In view of this situation, many researchers have done a lot of study [2-6], and the apriori algorithm is optimized, and the typical improved algorithms are as follows: a parallel mining algorithm, and it does not need pruning, and not generate candidate itemsets. Han et al propose the FP-Growth algorithm [7], and it does not generate candidate itemsets. Only need to scan the database for two times. Therefore, the mining efficiency is obviously improved. However, the FP-Growth algorithm needs to create recursively a large number of conditional pattern bases in the mining process. When a database is very large, the construction of FP-Tree based memory is not realistic. In order to overcome high cost calculation of the frequent pattern and avoid generating a lot of frequent items, Hu et al propose the LFIMiner algorithm based on FP-tree [8], this algorithm is used to deal with the pruning and optimization to shrink the search space, thus improve the performance of the algorithm effectively. VIPER puts forward a different algorithm [9], the data in the database is used by the representation of the longitudinal. Although the algorithm is the better efficiency, but still looks for from frequent pattern. Then an improved algorithm for mining the maximum frequent itemsets based of frequent pattern tree for mining the maximum frequent itemsets based on FP-tree was proposed, and this algorithm uses bottom-up search to mine the

maximum frequent itemsets, thus accelerated the count of candidates. If producing infrequent itemsets with lower dimension according to conditional pattern base on every layer when mining, cutting and reducing dimensions of candidate itemsets can largely reduce the amount of candidate itemsets [10]. An efficient CAR mining algorithm which is based on equivalence class-rules tree was proposed, and designs a frequent item sets for the storage datasets with tree structure, and it is an efficient algorithms based on equivalence class-rules tree [11]. Song proposed an improvement of apriori algorithm based on the KAF factor and the CHF factor to mine multi-valued attribute association rules and established a complete mining parameters adjustment mechanism acting very well in improving the speed and efficiency of mining [12]. A lattice-based approach for fast mining most generalization association rules is proposed, and a new algorithm for building a frequent-closed-itemset lattice is introduced, a theorem on pruning nodes in the lattice for rule generation is derived, and an algorithm for fast mining MGARs from the lattice constructed is developed [13]. Loan T et al propose an incremental method for mining class association rules when records are inserted into the dataset. a modified equivalence class rules tree is created from the original dataset, and it has some advantages, for example, The MECR-tree structure is used to generate rules quickly. The concept of pre-large itemsets is applied to CAR mining to reduce the number of re-scans on the original dataset. A theorem for quickly pruning infrequent nodes in the tree is developed to improve the process of updating the tree [14]. An efficient approach for mining cross-level closed itemsets and minimal association rules using closed itemset lattices is proposed, and design an efficient algorithm using a closed itemset lattice-based approach, which can mine the most relevant minimal cross-level association rules, and the parent-child relationship of the lattices has been exploited while mining cross-level closed itemset lattices [15]. A new approach based FEM and DFEM for frequent pattern mining (FPM) that runs fast for both sparse and dense databases is presented, and optimiza-

tion techniques for the proposed algorithms to speed the mining process, reduce the memory usage, and optimize the I/O cost [16]. The redefinition and classification of multi-valued attribute data by using conceptual lattice is presented, and proposes an improvement of Apriori algorithm based on the KAF factor and the CHF factor to mine multi-valued attribute association rules, and establishes a complete mining course parameters adjustment strategy acting very well in improving the speed and efficiency of mining algorithm [17].

Aiming at the problem of low efficiency about association rules mining, this paper studies and analyzes the method of mining association rule, and puts forward a kind of association rule mining algorithm based on the intersection, Finally, the effectiveness of the algorithm is verified by the examples and experiments, to a certain extent, improves the algorithm efficiency.

Next, we describe the algorithm used in more detail.

## 2. CORRELATION ALGORITHM FOR ASSOCIATION RULE MINING

The most classical algorithm of association rules is Apriori algorithm. Since it has many inherent defects, and researchers have proposed various improved algorithms based on Apriori [1].

### 2.1. Apriori Algorithm

The apriori algorithm is the most simple and basic algorithm when searching for frequent itemsets, and R. Agrawal and R. Srikant proposed the algorithm in 1994, and it is an original algorithm for mining frequent item sets of Boolean association rules, Apriori is used an iterative method of layer by layer search, the K item set for searching (k+1) item. First, the count of each item is accumulated by scanning the database, and the item that satisfies the minimum support degree is collected, then find out the collection of frequent 1 sets, the collection is recorded as L1. Then, L1 is used to find a collection of frequent 2 sets, recorded as L2, so go on, until you can no longer find frequent K itemsets.

The mining of association rules is divided into two steps: (1) find all frequent itemsets. (2) produce strong association rules from frequent itemsets. And the first step decides its overall performance. The first step is introduced as follow:

Algorithm: apriori algorithm

Input: the transactions database for D and min-sup

Output: frequent itemsets L of D

L1 = find\_frequent\_1-itemsets (D);

For (k = 2;  $L_{k-1} \neq \emptyset$ ; k++) {

$C_k$  = apriori\_gen ( $L_{k-1}$ , min\_sup);

for each transaction t D { //scan D for counts

$C_t$  = subset ( $C_k$ , t); //get the subsets of t that are candidates

for each candidate  $c \in C_t$

c.count++;

}

$L_k = \{c \in C_k \mid c.\text{count} \geq \text{min\_sup}\}$

}

return  $L = \bigcup_k L_k$  ;

Procedure apriori\_gen ( $L_{k-1}$ : frequent (k-1)-itemsets, min\_sup)

for each itemsets  $l_1 \in L_{k-1}$

for each itemsets  $l_2 \in L_{k-1}$

if ( $l_1[1]=l_2[1] \wedge l_1[2]=l_2[2] \wedge \dots \wedge l_1[k-2]=l_2[k-2] \wedge l_1[k-1] < l_2[k-1]$ ) then {

$c = l_1 \triangleright \triangleleft l_2$ ; //join step: generate canddates

if has\_infrequent\_subset (c,  $L_{k-1}$ ) then

delete c; // prune step: remove unfruitful candidate

else add c to  $C_k$ ;

}

return  $C_k$  ;

Procedure has\_infrequent\_subset (c: candidate k-itemset;  $L_{k-1}$ : frequent(k-1)-itemsets) //use prior knowledge

for each(k-1)-subset s of c

If  $s \in L_{k-1}$  then

return TRUE;

return FALSE;

The core idea of apriori algorithm is divided into two steps, and they are join and prune.

Step1: join

In order to find out the  $L_k$  (frequent K itemsets), the K itemsets is generated by  $L_{k-1}$  and its own connection, recorded as  $C_k$ , and the elements of  $L_{k-1}$  are connected.

Step2: prune

$C_k$  is a superset of  $L_k$ , i.e. its members may or may not be frequent, but all frequent itemsets are contained in the  $C_k$ . Once scan database, determine the count of each candidate in  $C_k$ , so as to determine the  $L_k$ . However,  $C_k$  may be very large, so the amount of computation involved is very large. For compressed  $C_k$ , using the Apriori property, and any non frequent (k-1) terms are not likely to be subset of frequent K itemsets. Thus, if (k-1) item of a candidate K itemset is not in  $L_k$ , the candidate item is not likely to be frequent, so it can be deleted from the  $C_k$ .

Apriori algorithm based on frequent item set uses the iterative method of layer by layer search. The algorithm is

simple and easy to implement. But there are some difficult to overcome. The number of scanning the database is too much, In the Apriori algorithm, when each candidate set is generated, we must carry out a comprehensive search in the database. If you want to generate frequent itemsets of a maximum length for N, we scan for N in the database. When a large number of transaction data is stored in the database, the load of the system I/O is very large, and the time of each scan database will be very long, so the efficiency is very low.

Apriori algorithm generates a large number of intermediate itemsets. Only using support, important degree of the various attributes are not considered. In real life, some affairs occur very frequently, and some affairs are very sparse, so there exists a problem for mining, if the minimum support threshold is set too high, meaningful rules may not be found. if the minimum support threshold is set too low, then a lot of no practical rules will fill in the whole data mining process, greatly reducing the availability and efficiency of mining rules

The algorithm only considers the single dimensional Boolean association rules mining, but in practical application, it may appear the multi-dimensional, the number, the multi-level association rules and inaccurate. At this time, the algorithm is no longer applicable, need to improve, and even need to redesign algorithm, but this algorithm is the basis of the improved algorithm later.

**3. CONCEPTS**

Suppose that  $I = \{i_1, i_2, i_3, \dots, i_m\}$  is a set of m different projects, Given a transaction databases for DB, thereinto,  $T \subseteq I$ , T has a unique identifier of TID, If X is an item of the I, and  $X \subseteq T$ , says that the transaction T support itemsets X. Association rules is implicative expression of  $X \Rightarrow Y$ , thereinto,  $X \subseteq I, Y \subseteq I$ , moreover  $X \cap Y = \emptyset$ .

Under normal circumstances, the support degree and credibility are two very important parameters in mining association rules, and they are looked as a standard of measuring the quality of mining association rules. The support degree (Support (  $X \Rightarrow Y$  ), that is, s % the transactions support both item sets A and B in the set D, s % is called the support degree of association rules  $X \Rightarrow Y$ , The support degree describes the probability of the union for two items in all transactions, and been expressed as:

$$\text{Support} ( X \Rightarrow Y ) = \text{Support} ( X \cup Y ) \tag{1}$$

The credibility degree (Credibility (  $X \Rightarrow Y$  ), that is, the transactions of in the set D support the transactions of the set X, at the same time c% support the set Y, c% is called the credibility degree of the association rules  $X \Rightarrow Y$ , that is, how much the probability of the item set Y does appear, at the same time the item set X appear in the transaction T, and is expressed as follows:

$$\text{Credibility}( X \Rightarrow Y ) = \text{Support} X \cup Y / \text{Support}(X) \times 100\% \tag{2}$$

The user specifies the minimum support degree and credibility according to the mining demand, and were re-

corded respectively as min\_con and min\_sup, The degree of support and credibility are a measure of the user's interest in knowledge, Only when  $\text{Support}( X \Rightarrow Y ) \geq \text{min\_sup}$  and  $\text{Confidence}( X \Rightarrow Y ) \geq \text{min\_con}$ , are called strong rules, and they are very important for the user, and are called the useful association rules. So we just need to find strong association rules, and calculate all the frequent itemsets according to the minimum support degree, then we may mine some long patterns of frequent item sets from all the frequent item sets, and finally calculate the maximum frequent sets.

Definition1. Suppose that T1 and T2 are item sets, S is composed of common item sets of T1 and T2, and is called the intersection of T1 and T2, and is recorded  $T1 \cap T2$ , and as follows:

$$S = T1 \cap T2 = \{x \mid (x \in T1) \wedge (x \in T2)\} \tag{3}$$

For example,  $T1 = \{a, b, c, d\}, T2 = \{a, b, d\}$ , then  $T1 \cap T2 = \{a, b, d\}$ . T1 and T2, for any two itemsets, definition1 can be generalized, if the minimum support degree is m, the frequent items is  $FI = T1 \cap T2 \cap \dots \cap Tm$ .

Definition2. Set any two limited itemsets T1 and T2, the number of its transactions are recorded respectively  $| T1 |, | T2 |$ , then  $| T1 \cap T2 | \leq \min ( | T1 |, | T2 | )$ .

Definition3. Suppose that X is a set of projects in the I, if X contains k items, X is called k - itemsets.

Definition4. If the itemsets  $X \subseteq T$ , is called transaction T satisfies itemsets X; the support degree of the itemsets X in the transaction database DB is recorded Sup (X), that is, the transaction databases DB contain the number of the transactions of X.

Definition5. If the support degree Sup (X) of the item set X in the transaction databases DB is greater than given minimum support degree threshold min-Sup for the user or the experts, the item set X is called X large itemsets or frequent itemsets.

Definition6. If all superset of the frequent itemsets L are non frequent itemsets, then L is called the maximum frequent itemsets (or is called the maximum frequent patterns), all collections of L is called the maximum frequent itemsets, and is recorded MFI (Maximal Frequent Itemsets).

Definition7. if the itemsets L is the maximum frequent itemsets, all subsets in the itemsets L are frequent itemsets FI (the frequent itemsets).

**4. MAXIMUM FREQUENT PATTERN ALGORITHM BASED ON THE INTERSECTION**

**4.1. The Algorithm**

The algorithm ISMFP is used to set theory and the idea of top-down search, and the implementation process as shown in Fig. (1).

Algorithm: ISMFP algorithm

Input: the transactions database for DB and min-sup

Output: MFI

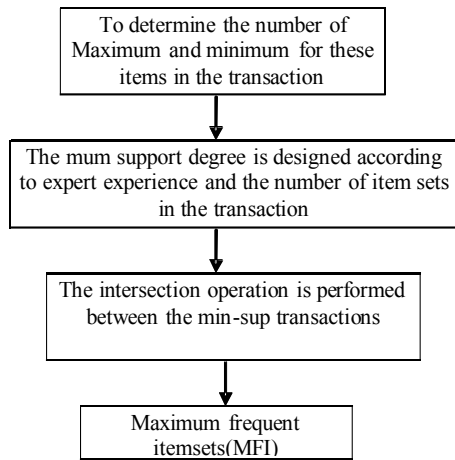


Fig. (1). The algorithm implementation process.

```

(1)  $L_k = \text{NULL}$ 
(2) The number of the transactions is determine in the DB, and  $|DB|$  is the number of the transactions, count up the number of each transaction, Therefore, T. count is the number of the transactions, Min_count is the minimum value of items, Max_count is the maximum value of items;
(3) For (T. count=Max_count; T. count >= Min_count; T. count--)
    // Taking out T. count itemsets from the transaction database DB, T. count is the number of itemsets}
(4) Saving the number of the transactions for the T. count items to an array of memory.
(5) Invoking function Intersection (T. count, min-sup)
Intersection (T. count, min-sup) {
    FI=NULL // FI is a frequent itemsets
    For (i=Max_count; i >= min-sup; i--)
        FI=Ti∩Ti-1∩ ... ∩Tmin-sup //Max_count itemsets and each itemsets of low order perform intersection operation. Therefore, a superset of FI is the non frequent itemsets)
     $L_k = FI$ 
    IF ( $FI \subseteq L_k$ )
        Delete FI from the frequent itemsets}
    
```

The algorithm can be divided into three parts: The first part of the statistical transactions, and its time complexity is  $O(\text{Max\_count} * |DB|)$ , among them,  $|DB|$  is the number of the transactions, Max\_count is the maximum length of the transactions; The second part time complexity of intersection operation is  $O(\text{Max\_count} * \text{Max\_count} * |DB|)$ ; And the third part, time of judgment superset and statistics transaction are similar, therefore, and their time complexity of the algorithm are  $O(\text{Max\_count} * \text{Max\_count} * |DB|)$ .

The algorithm of the maximum frequent patterns mining based on the intersection is a new method, Avoiding the

problem of the traditional Apriori algorithm to generate a large number of candidate itemsets and repeat scanning a database, and proposed algorithm improves the efficiency for mining association rules, the example analysis and the experiments results show that the algorithm is effective in this paper.

4.2. Example Analysis

The transaction data of All Electronics [1], As shown in Table 1, there is nine transactions in the database, That is  $|DB| = 9$ . Set threshold value about the minimum support degree, and min-sup=2.

According to the ISMFP algorithm can, we obtain the maximum frequent itemsets  $MFI = \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I2, I4\}\}$ , and the implementation process of ISMFP algorithm is as follows:

**Step 1:** Counting the number of each transaction in the transaction databases DB, as shown in Table 2.

**Step 2:** Starting from T8 of the largest number the items, but the number is less than min-sup, 4 itemsets of T8 and any of 3 itemsets are performed intersection operation,  $FI1 = T8 \cap T9 = \{I1, I2, I3\}$ ,  $FI2 = T8 \cap T4 = \{I1, I2\}$ ,  $FI3 = T8 \cap T1 = \{I1, I2, I5\}$ , after judging for the maximum frequent patterns, and FI2 is a subset of the intersection for subsequent itemsets, then get rid of, as follows:

$$MFI = \{\{I1, I2, I3\}, \{I1, I2, I5\}\}.$$

**Step 3:** After the intersection of 4 itemsets and 3 itemsets, then intersection operation about 3 sets and 2 sets are performed, Get the  $FI = \{I2, I4\}$ , after the judgment of the maximal frequent patterns, added to the MFI, and as shown in Table 3, and continue to execute the operation repeatedly by this method.

**Step 4:** Finally use ISMFP algorithm to get the maximum frequent item sets, as follows  $\{\{I2, I2\}, \{I1, I2, I3\}, \{I1, I2, I5\}\}$ , and as shown in Table 4.

**Step 5:** End.

Table 1. Transaction data.

TID	Item
T1	I1, I2, I5
T2	I2, I4
T3	I2, I3
T4	I1, I2, I4
T5	I1, I3
T6	I2, I3
T7	I1, I3
T8	I1, I2, I3, I1, I5
T9	I1, I2, I3

Table 2. Transaction statistics.

Items	
TID	Count
T1	3
T2	2
T3	2
T4	3
T5	2
T6	2
T7	2
T8	4
T9	3

Table 3. FI.

FI
11, 12, 13
11, 12, 15
11, 12
12, 14

Table 4. MFI.

MFI
12, 14
11, 12, 15
11, 12, 13

5. EXPERIMENT AND DISCUSSION

Based on CPU Pentium 4 3.06 GHz, 256 MB of memory, operating system for Windows server 2000 PC, 160G hard drive, using Microsoft Visual C++ edit program, The database system is the experimental environment of Microsoft Server SQL 2000, The test database is the same as the paper [18] in the test. The database has 8124 records, which record 23 the attributes of a mushroom has. Under the same conditions, we regard the DMFIA algorithm as the reference of the experiment, The Minimum support are respectively set: 30%, 20%, 10%, 1%, the experimental results are shown in Fig. (2) and effectiveness of the algorithm is verified in this paper.

According to the idea of algorithm, statistics the number of any item set at first, starting with the maximum terms, with the low one set min-sup transaction intersection operation, due to reduced cycle times and search space, Obviously reduce the cost of the need time, If the frequent item sets are the subset of the maximal frequent item sets, they are not added to the maximum frequent item sets, or there are such subsets, we will delete them. Certainly, the detection of the subset should be further optimized, this intersection is particularly suitable for the transaction support threshold is smaller or long patterns, and the efficiency is more obvious.

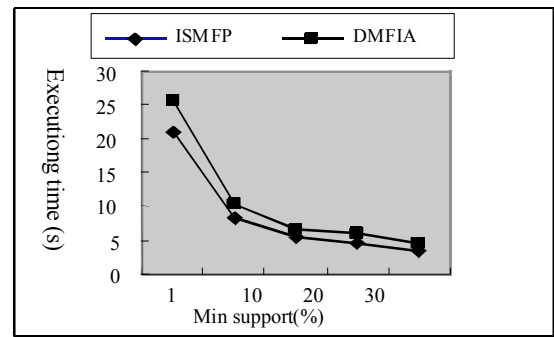


Fig. (2). Compare of algorithm performance.

It can be found from the experimental results, the ISMFP algorithm and the DMFIA algorithm with the increase of setting the support degree, time of the execution reduces constantly, When the support degree is 1%, time of Executed algorithm is 21.9 s in the ISMFP, but the DMFIA algorithm Executing time is 27.6 s, when support degree is 30%, ISMFP algorithm Executing time is 1.5 s, but the DMFIA algorithm executing time is 7.8 s, which shows that whether the preset minimum support degree is high or low, executing time of this algorithm is shorter, and faster and more efficient than the DMFIA algorithm. Therefore, the algorithm of this paper could improve the performance of mining maximum frequent patterns effectively.

CONCLUSION

Since the maximum frequent patterns have already implied all frequent patterns, we convert the problem of discovering frequent patterns to find the maximum frequent patterns, avoiding the production of a large number of candidate sets. This paper puts forward a kind of association rule mining algorithm based on the intersection, which reduces the number of cycles and the search space by using the principle of intersection and the maximum frequent pattern. Experimental results show that the proposed algorithm can improve the performance effectively.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

The authors would like to thank for financial support by youth fund project of the humanities and social sciences of Education Ministry for research on online social network public opinion mining and risk management based on big data analysis, Social science fund project of Hunan Province No. 13YBA302 science and technology plan project of Hunan Province No. 2014FJ3010, 2013FJ3032, Education department scientific research key projects of Hunan Province No.2014A135, education science "twelfth five-year" plan project of Hunan province No. XJK014CGD081, Xiangnan University research fund No.2012YY45 and Key projects of teaching reform of Xiangnan University.

REFERENCES

[1] M. Fan and X. F. Meng, "Data mining: concepts and techniques", China Machine PRESS: BeiJing, 2001.

- [2] R. J. Bayardo, "Efficiently mining long patterns from databases". Proc. 1998 ACM-SIGMOD Int'l Conf. Management of Data(SIGMOD'98). Seattle Washington, pp. 85-93, 1998.
- [3] R. C. Agarwal, C. C. Aggarwal, and V. V. V. Prasad, "Depth first generation of long patterns". In Proceedings of the ACM SIGMOD Conference, pp. 108-118, 2000.
- [4] D. Burdick, M. Calimlim, J. Gehrke, "A maximal frequent itemset algorithm for transactional databases". Int'l Conf. on Data Engineering, pp. 443-452, 2001.
- [5] G. Grahne, J. Zhu, "High performance mining of maximal frequent Itemsets", Proc of the 6th SIAM Int'l Workshop on High Performance Data Mining. San Francisco, pp. 135-143, 2003.
- [6] Y. X. Zhu, X. P. Zhang, and J. Y. Wang, "Improved algorithm for mining frequent item sets", *Computer Engineering and Applications*, vol. 45, no. 4, pp. 143-145, 2009.
- [7] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation", In: Proc ACM-SIGMOD Dallas TX, pp. 1-12, 2000.
- [8] T. M. Hu and S. Yuan, "Discovery of maximum length frequent itemsets", *Information Sciences*, vol. 178, no. 1, pp. 69-87, 2008.
- [9] P. Shenoy, J. R. Haritsa, and S. Sudarshan, "TurboVertical mining of large databases". SIGMOD Conference, pp. 22-33, 2000.
- [10] P. K. Yang, H. Peng, and X. F. Zhou, "FP-M FIA: improved algorithm for mining maximum frequent itemsets based on frequent—pattern tree". *Journal of Computer Application* vol. 35, no. 3, pp. 775-778, 2015.
- [11] X. W. Liu, High efficiency association rules mining algorithm based on equivalence class-rules tree", *Computer Applications and Software*, vol. 32, no. 1, PP. 313-319, 2015.
- [12] X. B. Guo, S. L. Zhao, and C. B. Wang, "Multi-valued attribute association rules mining based on concept lattice". *Computer Science*, vol. 41, no. 3, pp. 267-271, 2014.
- [13] B. Vo, T. P. Hong, and B. Le, "A lattice-based approach for mining most generalization association rules", *Knowledge-Based Systems*, vol. 45, no. 6, pp. 20-30, 2013.
- [14] T. T. Loan, "Nguyen ngoc-thanh nguyen. updating mined class association rules for record insertion". *Appl Intell*, vol. 42, pp. 707-721, 2015.
- [15] T. Hashema, C. F. Ahmeda, and M. Samiullah, "An efficient approach for mining cross-level closed itemsets and minimal association rules using closed itemset lattices". *Computer Science*, vol. 41, no. 3, pp. 267-271, 2014.
- [16] L. Vu and G. Alaghband, "Efficient algorithms for mining frequent patterns from sparse and dense databases", *Journal of Intelligent Systems*, vol. 24, no. 2, pp. 181-197, 2015.
- [17] X. B. Guo, S. L. Zhao, and C. B. Wang, "A New Visualizing Mining Method of Multi-Valued Attribute Association Rules for Ordinary Users", *Chinese Journal of Electronics*, vol. 43, no. 2, pp. 344-352, 2015.
- [18] Y. Q. Song, Y. Q. Zhu, Z. H. Sun, and G. Chen, "An algorithm and its updating algorithm based on FP-tree for mining maximum frequent itemsets", *Journal of Software*, vol. 14, no. 9, pp. 1586-1592, 2003.

---

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© Chen and Xiao; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the (<https://creativecommons.org/licenses/by/4.0/legalcode>), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.