IWFPM: Interested Weighted Frequent Pattern Mining with Multiple Supports

Xuyang Wei[1*], Zhongliang Li[1], Tengfei Zhou[1], Haoran Zhang[1], Guocai Yang[1]
[1] School of Computer and Information Science, Southwest University, Chongqing, China.

* Corresponding author. Tel.: 15086897915; email: paul.g.yang@gmail.com
Manuscript submitted January 15, 2014; accepted September 12, 2014.

Abstract: Association rules mining has been under great attention and considered as one of momentous area in data mining. Classical association rules mining approaches make implicit assumption that items' importance is the same and set a single support for all items. This paper presents an efficient approach for mining users' interest weighted frequent patterns from a transactional database. Our paradigm is to assign appropriate minimum support (minsup) and weight for each item, which reduces the number of unnecessary patterns. Furthermore, we also extend the support-confidence framework and define an interest measure to the mining algorithm for excavating users' interested patterns effectively. In the end, experiments on both synthetic and real world datasets show that the proposed algorithm can generate more interested patterns.

Key words: Data mining, frequent pattern mining, association rules technology, interest.

1. Introduction

Data mining aims to discover knowledge patterns hidden in large database. Several data mining approaches are used to extract interesting knowledge [1]. Like, association rule [2], the area being researched most actively, is a technique to detect the relation in the data. Classification [3] is used for identifying the different class existing in categorical labeled data, clustering [4] is a technique to group the data into clusters based on similarities among them. All these techniques may be applied widely according to the needs of the enterprise.

In the paper, we mainly study the association rules technique, whose purpose is to search for the interesting correlations, associations between items in the data repositories. In the existing applications, the algorithms for mining frequent item sets can be classified into two types: Apriori-like methods and FP-growth-like methods. Among them, the Apriori-like methods are based on the well-known Apriori algorithm [2] and focus on reducing candidate item sets. The FP-growth-like methods are on the basis of FP-growth algorithm [5] and avoid generating candidate item sets by building a FP-tree, which overcome the bottlenecks of the Apriori-like algorithm. Later, references [6]–[9] have made corresponding improvement in time and space. However, these improved methods have discovered frequent patterns by setting fixed consistent minsup for all items and concerns are still unsolved.

The first concern is that all items are treated uniformly in previous approaches, but real items have different importance. For this reason, weighted frequent pattern mining algorithms have been suggested to filter interesting rules by attaching different weights to items [10]–[13]. High weights are attached to items of high importance and the unimportant rules are easily differentiated. Ahmed *et al*. [14] also introduce the dynamic weight for each item, and propose the algorithm DWFPM (dynamic weighted frequent pattern mining). Reference [15] extends prior research based on the Valency model and automates the process of weight assignment by formulating a linear model that capturing relationships between items. The second concern is that most traditional mining algorithms have used the single constant minsup, which bring "rare item problem" dilemma, and how to set the reasonable minsup becomes a serious problem. If the minsup is high, a lot of potential item sets are removed during the pruning process, whereas it may generate a huge number of unnecessary item sets. For these problems, Liu *et al*. [16] have put forward the MS Apriori Algorithm based on multiple supports for frequent pattern discovery. References [17], [18] use the maximum constraint and propose a simple algorithm based on the Apriori approach for finding the large-item sets and association rules under the constraint. Wang *et al*. [19] propose an array based tail node tree structure (namely AT-Tree) to maintain transaction item sets, and a pattern-growth based algorithm named AT-Mine for FIM over uncertain dataset. Yang *et al*. [20] propose the IA spam algorithm that not only can handle a set of items at a time but also can incrementally mine across-streams sequential patterns. Kiran *et al*. [21] also have presented an improved approach on account of the notion of "support difference", where each item is given suitable minsup to mine rare frequent patterns.

As the two problems mentioned above are not opposite, in further studies, for the sake of settling the two considerations simultaneously, Duan *et al*. [22] propose the algorithm AMWARMS to excavate weighted frequent patterns with multiple supports. However, the multiple supports considered in the algorithm is not the frequency of different items but itemsets. Then, Zou *et al*. [23] come up with a new model of weighted frequent patterns mining and design the algorithm DWARMMS, which allows the user to set different weight and minsup for the item according to their importance and frequency in the database. Yun *et al*. [24] present a new strategy WIP, which not only gives a balance between the two measures of weight and support, but also considers weight affinity and support affinity between items within patterns.

Yet, these algorithms are all based on the support-confidence framework without taking the users' real preference into account. Therefore, this paper proposes an efficient algorithm IWFPM(Interested Weighted Frequent Pattern Mining), and adds an interest measure reflecting users' favors to the algorithm in that it can excavate more users' interest frequent patterns.

The remainder of paper is organized as follows. In Section 2, we provide some background information on association rules technology. Section 3 develops the mining algorithm IWFPM, and extensive experimental results are shown in Section 4. Finally, Section 5 reports the conclusion and future work.

2. Related Works

2.1. Definition of Association Rules

Association rules have been first initiated by Agrawal *et al*. [2] and continuing as the research hotspot in data mining all the same. The basic terminology about association rules is as follows. Let $I = \{i_1, i_2, ..., i_n\}$ be a unique set of items. A transaction database, T, is a set of transactions and each transaction is denoted as a tuple $< tid, X >$, where tid is the identifier of the transaction and $X \subseteq I$. We call X a k-pattern if it contains k items. A frequent pattern is an implication of the form, $X \Rightarrow Y$ that satisfying $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$, holds in T with support s if $s\%$ of the transactions in T contain $X \cup Y$. Similarly, rule $X \Rightarrow Y$ holds in T with confidence c if $c\%$ of transaction in T that support X and Y. With regarding to support and confidence in discovering the association rules, the user shall set the minimum support (minsup) and minimum confidence (min conf) as critical values that providing the baselines for discovery. That is, for a rule R, if $\sup(R) \geq \min\sup$ and $conf(R) \geq \min conf$, there exists a pattern R.

As association rules have been researched and applied in diverse ways. In this paper, we take the foundational association rules algorithms Apriori and FP-growth for example to explain the process of mining association rules in different angles.

2.2. Apriori Algorithm

Apriori algorithm employs an iterative level-wise search for generating frequent item sets. The most significant characteristic of Apriori approach is that it constitutes from the previous frequent item sets rather than all the data items accessed in the transaction when selecting candidate item sets. The frequent item sets refer to the item sets whose supports are greater than or equal to the user's specified minsup. Here, C_k is the candidate item sets, where k is the number of item in the item set. Likewise, L_k represent a k-frequent item set. The Apriori algorithm executes as follows: 1) C_k is generated. 2) L_k is generated from C_k by pruning the item sets. 3) C_{k+1} is generated by joining L_k with itself. Then, repeats the steps from 1) to 3) starting with k=1 till no more frequent item sets are found [3].

The process of mining frequent item sets using Apriori is depicted in Fig. 1.

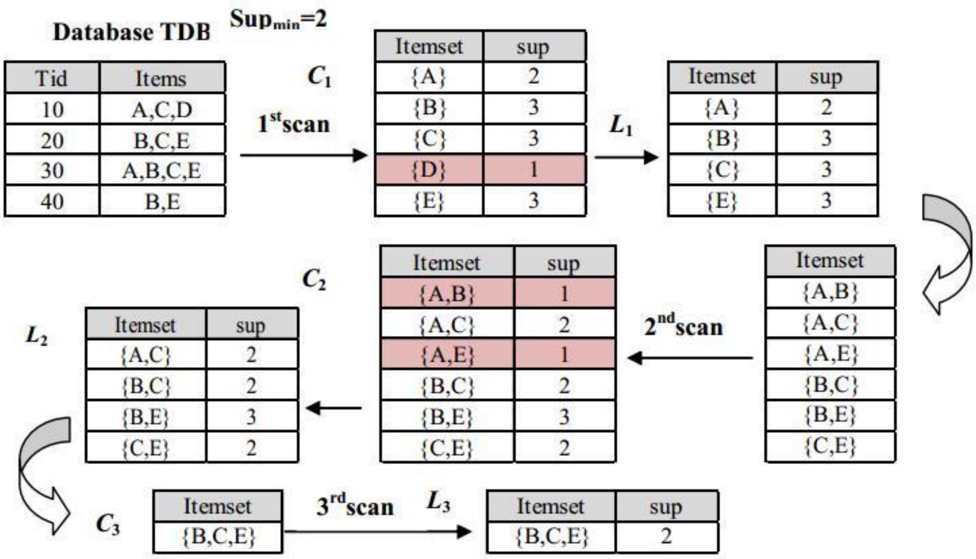


Fig. 1. The working of Apriori algorithm.

As can be seen from the above, in the case of massive data, Apriori has two defects: 1) it has to repeatedly scan a large amount of original database in order to check whether a candidate item set is frequent, and time cost is very high; 2) there is a huge number candidate k-frequent item sets generated by the connection of k-1-frequent item sets so that space cost is also very high.

2.3. FP-Growth Algorithm

For figuring out the above issues, Han *et al.* [5] propose the frequent pattern tree (FP-tree) structure for storing compressed, crucial information about frequent patterns, and develop the FP-growth algorithm. There are two differences with Apriori algorithm: 1) generates no candidate item sets; 2) only needs to traverse database twice. Its procedures are as follows: 1) finds the frequent items and calculates their frequencies, then sorts items with frequency in descending order for each transaction; 2) constructs a novel FP-tree structure; 3) uses the FP-tree structure build conditional trees recursively for mining frequent item sets [25].

The process of FP-growth algorithm is shown in Fig. 2.

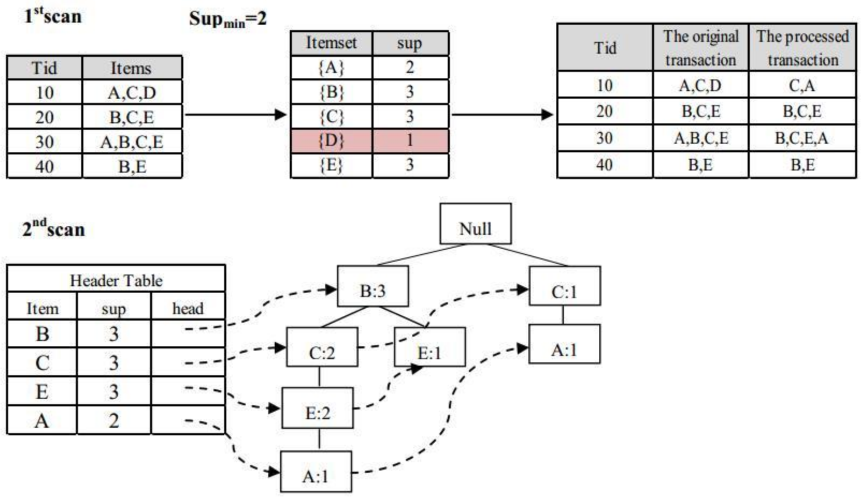


Fig. 2. The process of FP-growth algorithm.

Fig. 2 illustrates that FP-growth algorithm applies a pattern growth method to avoid the costly candidate generation by concatenating frequent items in the FP-tree. Then we acquire the conditional pattern-base of each item from the FP-tree recursively, as is depicted in Table 1.

Table 1. Conditional Pattern-Base of Each Item

Item	Conditional pattern-base	Conditional FP-tree	Frequent patterns
A	{(C:1),(B,C,E:1)}	{(C:2)}|A	C,A:2
E	{(B:1),(B,C:2)}	{(B:3,C:2)}|E	B,E:3;C,E:2;B,C,E:3
C	{(B:2)}	{(B:2)}|C	B,C:2
B	Φ	Φ	Φ

In view of the above analysis, though FP-growth algorithm has overcome disability of Apriori algorithm, there still exist deficiencies. First, it is not feasible to construct a main memory-based FP-tree if the database is large and sparse. Second, it needs to build a conditional pattern base and conditional sub-FP-trees for each shorter pattern, thus it is time and space consuming operation. Therefore, the paper presents an efficient approach IWFPM to extract more interest frequent patterns.

3. The Proposed Approach

In this section, weighted frequent pattern, weighted minsup and interest are described. Then, the efficient mining algorithm IWFPM for excavating more interesting weighted frequent pattern is given.

3.1. Definitions

3.1.1. Weighted frequent pattern

On the basis of the definition of association rules in section II, we redefine weighted frequent pattern in our approach. Analogously, $I=\{i_1,i_2,\ldots,i_n\}$ is a set of items and $W=\{w_1,w_2,\ldots w_m\}(w_i\in[0,1])$, a set of non-negative real numbers, which refers to the different importance of each item. After a single item has the weight, the item set containing the items also has the corresponding weight. A pair (X, w(X)) is called a weighted item set, where $X\subset I$ and $w(X)\subset W$. The weighted support of an item set X, denoted as $w\sup(X)=\sup(X)*w(X)$, where $\sup(X)$ is the percentage of the number of transactions in database containing X, and the weight of item set X is defined as

$$w(X)=\frac{\prod_{k=1}^{|X|} w(i_k)}{\sum_{k=1}^{|X|} w(i_k)}, \quad i_k \in X \tag{1}$$

Similarly, the weight support of item set $X \cup Y$ is calculated as:

$$w\sup(X \cup Y)=\sup(X \cup Y) * \frac{\prod_{k=1}^{|X \cup Y|} w(i_k)}{\sum_{k=1}^{|X \cup Y|} w(i_k)}, \quad i_k \in X \cup Y \tag{2}$$

Therefore, weighted confidence of rule $X \Rightarrow Y$ is defined as $wconf(X \Rightarrow Y)=\frac{w\sup(X \cup Y)}{w\sup(X)}$, where $X \subset I$, $Y \subset I$. At the same time, rules' definition has also changed when compared with the previous definition. $X \Rightarrow Y$ is a valid rule if its weighted support $w\sup(X \cup Y)$ and weighted confidence $wconf(X \Rightarrow Y)$ satisfying minimum weighted support (minwsup) and minimum weighted confidence (minwconf).

3.1.2. Weighted minsup

In this definition, the minsup is not a single specified value that set according to users' experience, but a value determined by the importance and frequency of item in the database. Hence, for each item i_k, the calculation of weighted minsup ($MIS(i_k)$) is as follows:

$$\begin{aligned} MIS(i_k) &= w(i_k) \times (sup(i_k) - SD) \quad when \quad (sup(i_k)-SD) > LS \\ &= w(i_k) \times LS \quad otherwise \end{aligned} \tag{3}$$

where $sup(i_k)$ refers to support of item i_k, support difference (SD) indicates the acceptable deviation of an item from its frequency, and LS refers to the user-specified least support. We define the least support (LS) mainly for preventing MIS value of each item from reaching 0 or lower. Least support is the lowest minsup that an item or item set needs to satisfy for being frequent, and LS takes a value in [0, 1].

For a given database, the value of SD can be calculated as (4).

$$SD=\alpha(1-\beta) \tag{4}$$

where α represents the parameter like mean, median, mode, maximum support of items and β ranges from 0 to 1, SD takes the value in (0, α).

Both α and β play major role in determining SD. This paper, we select mean support of all items as β. Therefore, after specifying MIS value for each item as (3), the frequent item sets are generated using (5).

$$w\sup(X) \geq \min(MIS(i_1), MIS(i_2), \ldots, MIS(i_k)), \quad i_k \in X \tag{5}$$

The Equation (5) guarantees the extraction of frequent item sets from frequent items, rare items efficiently. That is to say, a constant difference between support of item and their MIS can be confirmed irrespective of the support values. So it can be used in all kinds of datasets whose item supports vary widely.

3.1.3. Interest

Usually, the basic evaluations of association rules are support and confidence. However, there exists a problem, if the confidence of rule $X \Rightarrow Y$ tells us that the possibility of X and Y appearing in the meantime is high, whether X accelerates the emergence of Y. Reference [25], [26] has given the detailed explanation. The conclusion is that rule $X \Rightarrow Y$ cannot correctly reflect the trend of appearance of Y. Thus, we define the interest and add it to the mining algorithm based on the difference between information expressed by association rules and information supported by all original records.

The accuracy of information shown by association rule $X \Rightarrow Y$ is determined by its confidence, while the information supported by all original records can be represented by sup(X). So we define the interest as (6).

$$Interest_R = \frac{C_R - \sup(Y)}{\max\{C_R, \sup(Y)\}} \tag{6}$$

where C_R is the confidence of rule $|X \cup Y| / |X|, X \Rightarrow Y$ is the support, right part of rule. Obviously, C_R and sup(Y) do not exist relationship in any quantity, therefore, $Interest_R$ may be greater than or less than 0. We introduce the denominator value as a normalization factor that making the interest be in -1 and 1, and this paper only considers the interest to be greater than 0. Then we again modify the definition of rule, namely a valid rule to be interest that its interest must meet the minimum interest pre-defined by users.

3.2. Algorithm

With the definition of weighted frequent pattern, weighted minsup, interest above, we now present the efficient algorithm, and introduces the interest as a measure to evaluate the frequent patterns mined to be interest or not.

Algorithm: IWFPM (Interested Weighted Frequent Pattern Mining)

Require: T: transaction dataset, W: weight of each item, SD: support difference, LS: least support, min wconf: minimum weighted confidence, min Interest: minimum interest

Ensure: Interest weighted frequent patterns

1) Generate candidate 1-itemset C_1
2) Calculate weighted support $w\sup$ for item sets in C_1
3) $MIS = calculate\_mis(w\sup, SD, LS)$
4) $L_1 = \{<i> \mid i \in C_1, w\sup(i) \ge MIS(i)\}$
5) for k=2; $L_{k-1} \neq \varnothing$; ++k do
6) if (k = 2)
7) $C_2 = lev2\_gen\_candidate(C_1)$
8) else
9) $C_k = gen\_candidate(L_{k-1})$
10) end if
11) for transaction $t \in T$ do
12) $C_t = subset(C_k, t)$

13) for each candidate $c \in C_t$ do

14) update $w\sup(c)$

15) end for

16) end for

17) $L_k = \{c \mid c \in C_k, w\sup(c) \geq \min(MIS(c)\}$

18) $L = L \cup L_k$

19) end for

20) $Rules\_Set = \min e\_association\_rules(L)$

The algorithm has the following instructions.

1) $MIS = calculate\_mis(w\sup, SD, LS)$: the procedure for calculating MIS value of each item according to its wsup, SD, LS.

2) $C_2 = lev2\_gen\_candidate(C_1)$: generates 2-frequent item sets. As there may be 1-infrequent item sets becoming the candidate of 2-frequent item sets, that is C_2 derives from C_1 rather than L_1, therefore, generating the 2-candidate item sets require to be implemented solely.

3) $C_k = gen\_candidate(L_{k-1})$: generates candidate item sets.

4) $C_t = subset(C_k, t)$: generates all $k-1$ subsets of the candidate C_k.

5) $Rules\_Set = \min e\_association\_rules(L)$: produces the interest frequent patterns that meeting the minwconf and minInterest from frequent item sets in L.

4. Performance Evaluation

In order to evaluate the IWFPM algorithm, the operating environment chosen is as follows: Intel（R）Xeon（R）CPU E7-4820@2.00 GHz, 32G memory, Linux O/S, and we use Matlab R2011b implement the algorithm. We also select two kinds of datasets: synthetic dataset and real world dataset. The synthetic dataset is the famous UCI dataset Mushroom that containing 119 items and 8124 transactions. The real world dataset is the processed log data, which is a week of log of the server of Chongqing Agricultural and Rural Information Network, and it contains 1590 items and 50192 transactions.

As each item in real world dataset refers to a page of website, we choose relative browsing time and access frequency as the two factors to measure the importance of the page. The weight of page i is defined as:

$$w(i) = \frac{2time(i) \cdot freq(i)}{time(i) + freq(i)} \tag{6}$$

Weights allocation of each item is shown in Fig. 3.

In the experiment, the proposed IWFPM approach requires two parameters, LS and β. Therefore, first of all, the performance of IWFPM measured by elapsed time is evaluated by β value, and then we fix the LS value at 1%, 3%and 5%. The value of α is defined as the mean of all items. Fig. 4(a) shows the performance of IWFPM at various β values for synthetic dataset. It can be observed that at low β value, the time consumption for mining weighted frequent patterns is high. When β is high, the time decreases, which is consistent with the single minimum support algorithm. Fig. 4(b) shows the performance of IWFPM at various β values for real world dataset.

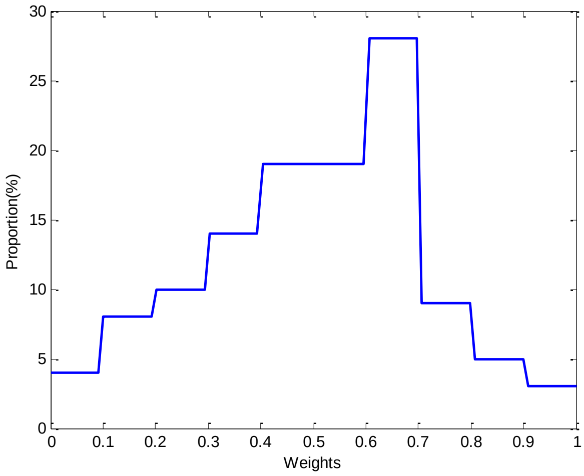
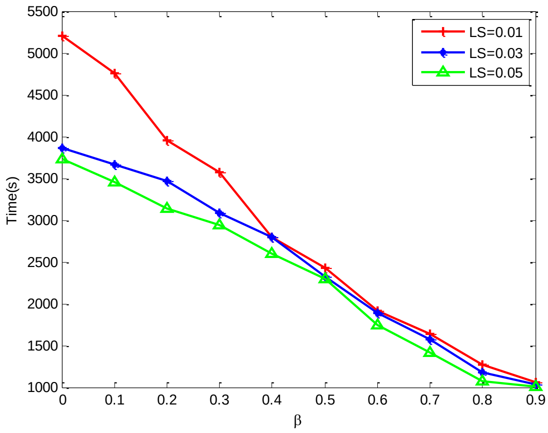
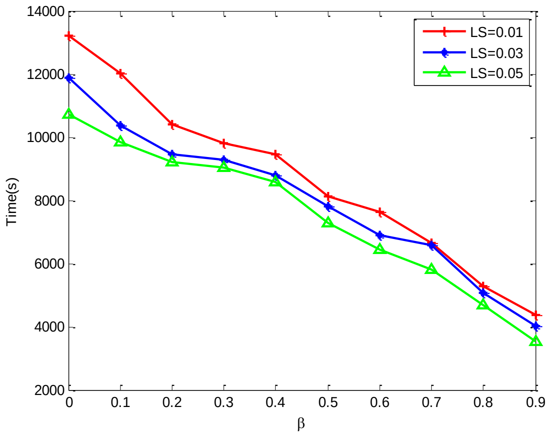


Fig. 3. Weights allocation in synthetic dataset.

Then, we compare the IWFPM algorithm to IMSApriori algorithm presented in reference [20]. As the two algorithms have the common parameters, LS and β, and the difference is that our method gives the weight for each item. Hence, as is shown in Fig. 5, we take β=0.2, and observe the performance of the two approaches at various LS value. Fig. 5(a) shows the comparison of IWFPM and IMSApriori algorithm in synthetic dataset. It is known that IWFPM is better than IMSApriori on time performance. Fig. 5(b) shows the time consumption varied with different LS value for IWFPM and IMSApriori approaches in real world dataset.

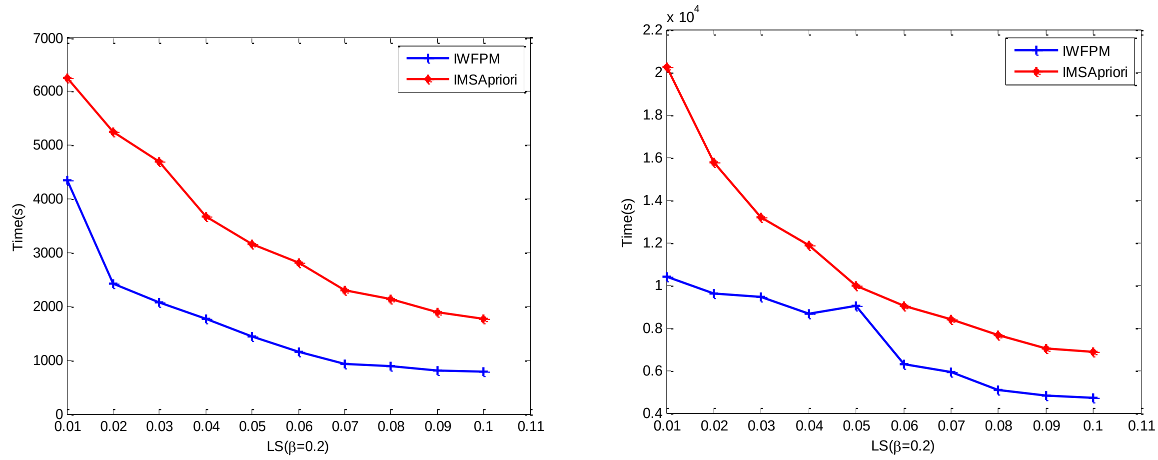




(a) The elapsed time of IWFPM at various β values in synthetic dataset (b) The elapsed time of IWFPM at various β values in real world dataset

Fig. 4. The elapsed time of IWFPM at various β values.

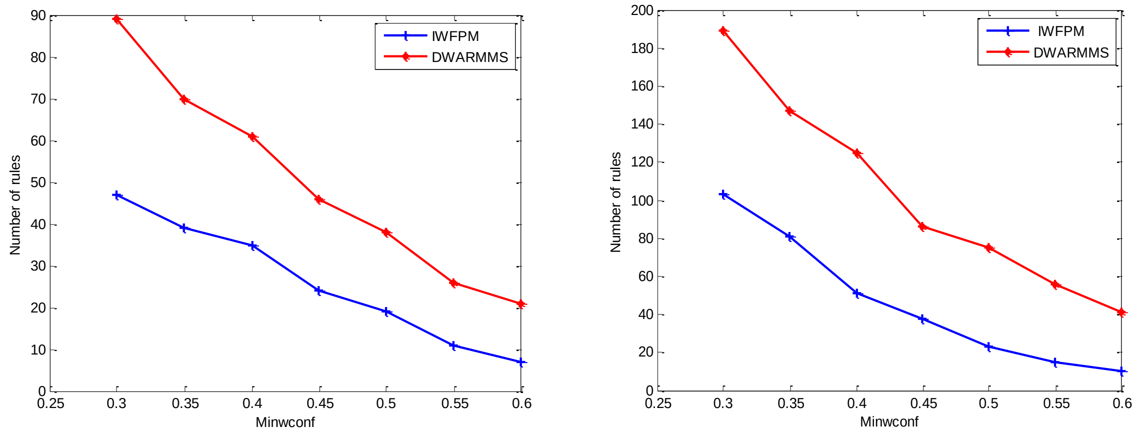
In addition, another innovation of IWFPM algorithm is to define the $Interest_R$, which reflects the users' likeability to the rules. Therefore, we also compare IWFPM algorithm to DWARMMS algorithm presented in reference [22] at various minwconf by setting LS=0.01, β=0.2 and minInterest=0.4. Fig. 6(a) shows the performance of IWFPM and DWARMMS approaches for synthetic dataset, and it illustrates that the number of rules diminishes as minwconf increases. What's more, the proposed approach excludes a large number of redundant rules and mines the interested rules with the same minwconf. Fig. 6(b) shows the performance of IWFPM and DWARMMS approaches in real world dataset.



(a) The comparison of IWFPM and IMSApriori in synthetic dataset

(b) The comparison of IWFPM and IMSApriori in real world dataset

Fig. 5. The comparison of IWFPM and IMSApriori.



(a). The number of rules generated with different wconf in synthetic dataset

(b). The number of rules generated with different wconf in real world dataset

Fig. 6. The number of rules generated with different minconf.

5. Conclusion and Future Work

In this paper, we have developed the efficient mining approach to extract frequent item sets for discovering interest association rules. Our approach assigns appropriate minimum support (minsup) value and weight for all items and extends the support-confidence framework. Meanwhile, an interest measure is defined to the mining algorithm for the sake of extracting more users' interest frequent patterns. Finally, the performances demonstrate that the proposed algorithm is superior to the existing algorithms. In the future, we are going to improve the efficiency of the algorithm and apply it in various fields. We also believe that further researches require more precise techniques to mine frequent patterns that reflecting users' real preferences.

Acknowledgment

This work is supported by the key technology integration and demonstration of rural Internet information services (2012BAD35B08), Chongqing, China

References

[1] Melli, G., Zaïane, O. R., & Kitts, B. (2006). Introduction to the special issue on successful real-world data mining applications. *SIGKDD Explorations*, *8(1)*, 1-2.

[2] Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, *22(2)*, 207-216.

[3] Weiss, S. M., & Kulikowski, C. A. (1990). Computer systems that learn: Classification and prediction methods from statistics. *Neural Nets, Machine Learning and Exp*.

[4] Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16(3)*, 645-678.

[5] Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM SIGMOD Record*, *29(2)*, 1-12

[6] Dong, J., & Han, M. (2007). BitTableFI: An efficient mining frequent itemsets algorithm. *Knowledge-Based Systems*, 20(4), 329-335.

[7] Zhang, L., Wang, M., & Zhai, Z. J. (2014). Mining frequent closed patterns using sample growth in resource effectiveness data. *Journal of Computers*, *9(5)*, 1150–1158.

[8] Song, Y. Q., Zhu, Y. Q., Sun, Z. H., *et al*. (2003). An algorithm and its updating algorithm based on FP-tree for mining maximum frequent itemsets. *Journal of software*, *14(9)*, 1586-1592.

[9] Yang, Y., & Luo, Y. X. (2010). Improved algorithm based on FP-growth. *Computer Engineering and Design*, *31(7)*, 1506-1509.

[10] Cai, C. H., Fu, A. W. C., Cheng, C. H., *et al*. (1998). Mining association rules with weighted items. *Proceedings of IEEE on Database Engineering and Applications Symposium* (pp. 68-77).

[11] Wang, W., Yang, J., & Yu, P. S. (2000). Efficient mining of weighted association rules (WAR). *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 270-274).

[12] Chang, J. H. (2011). Mining weighted sequential patterns in a sequence database with a time-interval weight. *Knowledge-Based Systems*, *24(1)*, 1-9.

[13] Sun, K., & Bai, F. (2008). Mining weighted association rules without pre-assigned weights. *IEEE Transactions on Knowledge and Data Engineering*, *20(4)*, 489-495.

[14] Ahmed, C. F., Tanbeer, S. K., Jeong, B. S., *et al*. (2008). Handling dynamic weights in weighted frequent pattern mining. *IEICE Transactions on Information and Systems*, *91(11)*, 2578-2588.

[15] Pears, R., Koh, Y. S., Dobbie, G., *et al*. (2013). Weighted association rule mining via a graph based connectivity model. *Information Sciences*, *218*, 61-84.

[16] Liu, B., Hsu, W., & Ma, Y. (1999). Mining association rules with multiple minimum supports. *Proceedings of the Fifth ACM SIGKDD International Conference on knowledge Discovery and Data Mining* (pp. 337-341).

[17] Wang, K., He, Y., & Han, J. (2000). Mining frequent itemsets using support constraints. *VLDB* (pp. 43-52).

[18] Lee, Y. C., Hong, T. P., & Lin, W. Y. (2005). Mining association rules with multiple minimum supports using maximum constraints. *International Journal of Approximate Reasoning*, *40(1)*, 44-54.

[19] Wang, L., *et al*. (2013). At-mine: An efficient algorithm of frequent itemset mining on uncertain dataset. *Journal of Computers*, *6(63)*, 1417–1426,

[20] Yang, S., *et al*. (2011). Incremental mining of across-streams sequential patterns in multiple data streams. *Journal of Computers*, *8(6)*, 449–457,

[21] Uday, K., *et al*. (2009). An improved multiple minimum support based approach to mine rare association rules. *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining* (pp. 340-347).

[22] Duan, J., & Dai, J. F. (2006). Algorithm of mining weighted association rules based on multiple supports. Journal of Tianjin University of Science and Technology, *39(1)*, 114-118.
[23] Qishan, Z. L. Z. (2007). Algorithm of weighted association rules mining with multiple minimum supports. *Journal of Beijing University of Aeronautics and Astronautics*, *5*, 20.
[24] Yun, U. (2007). Efficient mining of weighted interesting patterns with a strong weight and/or support affinity. *Information Sciences*, *177(17)*, 3477-3499.
[25] Tsay, Y. J., Hsu, T. J., & Yu, J. R. (2009). FIUT: A new method for mining frequent itemsets. *Information Sciences*, *179(11)*, 1724-1737.
[26] Xin, Z., Feng, S. H. A. C., Yong, Z. H. U. Y., *et al*. (2000). Interest measure—Another threshold in association rules. *Journal of Computer Research and Development*, *5*, 21.

Xuyang Wei is a master degree candidate at Southwest University in Chongqing, China. Her major is web mining and personalized recommendation.

Guocai Yang is a professor at Southwest University of China. His main research interests include computer application technology, agricultural information technology and software testing technology. He has successively secured more than 10 research grants from various research funding agents.

Zhongliang Li is a master degree candidate at Southwest University in Chongqing, China. His major is computer application technology and its application in agriculture.

Tengfei Zou is a master degree candidate at Southwest University in Chongqing, China. His major is web mining.

Haoran Zhang is a master degree candidate at Southwest of China in Chongqing, China. Her research interest is in the application of big data Technology in agriculture.