Weighted Association Rule Mining using Weighted Support and Significance Framework

Feng Tao
Department of Electronics and Computer Science
University of Southampton
Southampton, UK
ft@ecs.soton.ac.uk

Fionn Murtagh
School of Computer Science
Queen's University Belfast
Belfast, UK
F.Murtagh@qub.ac.uk

Mohsen Farid
School of Computer Science
Queen's University Belfast
Belfast, UK
m.farid@acm.org

ABSTRACT

We address the issues of discovering significant binary relationships in transaction datasets in a weighted setting. Traditional model of association rule mining is adapted to handle weighted association rule mining problems where each item is allowed to have a weight. The goal is to steer the mining focus to those significant relationships involving items with significant weights rather than being flooded in the combinatorial explosion of insignificant relationships. We identify the challenge of using weights in the iterative process of generating large itemsets. The problem of invalidation of the "downward closure property" in the weighted setting is solved by using an improved model of weighted support measurements and exploiting a "weighted downward closure property". A new algorithm called WARM (Weighted Association Rule Mining) is developed based on the improved model. The algorithm is both scalable and efficient in discovering significant relationships in weighted settings as illustrated by experiments performed on simulated datasets.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications – *Data Mining*

Keywords

Weighted Association Rule Mining, Weighted Support, Significant relationship, weighted downward closure property, WARM algorithm.

1. Introduction

Association Rule is an important type of knowledge representation revealing implicit relationships among the items present in large number of transactions. Given $I = \{i_1, i_2, ..., i_n\}$ as the items' space, which is a set of items, a transaction may be defined as a subset of *I*, and a dataset may therefore be defined as a set *D* of transactions. X and Y are non-empty subsets of *I*. The support of an itemset *X* in a dataset *D*, denoted as *supportD(X)*, is defined *as countD(X)/|D|*, where *countD(X)* is the number of transactions in *D* containing *X*. An itemset is said to be frequent (large) if its support is larger than a user-specified value (also called minimum support (*min_sup*)). An association is an implication of the form [$X \rightarrow Y$, *sup*, *conf*], where $X \subset I, Y \subset I$, and $X \cap Y = \emptyset$. The support of $X \cup Y$ (*sup*) in the transactions is larger than min_sup, furthermore when *X* appears in a transaction, *Y* is likely to appear in the same transaction with a probability *conf*. Given a threshold of minimum support and confidence, methods of discovering association rules [4, 5, 6, 9] have become active research topics since the publication of Agrawal, Imielinski and Swami and Agrawal and Srikant papers [2, 3].

SIGKDD 2003

However, the traditional association rule mining (ARM) model assumes that items have the same significance without taking account of their weight/attributes within a transaction or within the whole item space. But this is not always the case. For example, [wine→ salmon, 1%, 80%] may be more important than [bread → milk, 3%, 80%] even though the former holds a lower support. This is because those items in the first rule usually come with more profit per unit sale, but the standard ARM simply ignores this difference.

Several initiatives have been made. We identify the main challenge of adapting traditional association rule mining model in a weighted setting as the invalidation of the "downward closure property", which is used to justify the efficient iterative process of generating and pruning large itemsets from its subsets.

In order to tackle this challenge, we made adaptation on the traditional association rule mining model under the "significant – weighted support" metric framework instead of the "large – support" framework used in previous works. In this new proposed model, the iterative generation and pruning of significant itemsets is justified by a "weighted downward closure property".

2. Background and related work

Most of the current work on the traditional Apriori algorithm [2] make use of the "large – support" metric framework. However these works still view items as having equal weights though trying to distinguish them using various methods.

Wei Wang et al. proposed an efficient mining methodology for Weighted Association Rules (WAR) [12]. The idea is inspired by the fact that a numerical attribute can be assigned for every item which in turn judges the weight of the item in a

particular weight domain. For example, soda[4,6] → snack [3,5] is a targeted weighted association rule meaning that if a customer purchases soda in the quantity between 4 and 6 bottles, he is likely to purchase 3 to 5 bags of snacks. WAR uses a two-fold approach where the frequent itemsets are generated through standard association rule mining algorithms without considering weight. Post-processing is then applied on the frequent itemsets during rule-generation to derive the maximum WARs. WAR doesn't interfere with the process of generating frequent itemset. Rather, it focuses on how weighted association rules can be generated by examining the weighting factors of the items included in generated frequent itemsets. Therefore, we could classify this type of weighted association rule mining methods as a technique of post-processing or maintaining association rules.

Han et al. [7] proposed a solution where a concept hierarchy was used and association rules were classified into multiple conceptual levels of granularity. This idea inspires the work in [8] where the existing association rule model is extended to allow users to specify multiple threshold supports. In the extended model, the threshold support is expressed in terms of *minimum item supports (MIS)* of the items that appear in the rule. The main feature of this technique is that the user can specify a different threshold item support for each item, similar to the scenario of assigning weights to items. This technique can discover rare item rules without causing frequent items to generate too many unnecessary rules. Liu's model also breaks the "downward closure property". The problem is solved by using a "sorted closure property" where the items in the item space are sorted in ascending order of their MIS values.

3. Preliminaries

Let $I = \{i_1, i_2, ..., i_3\}$ be a set of distinct items and W be a set of non-negative real numbers. A pair (x, w) is called a weighted item where $x \in I$ is an item and $w \in W$ is the weight associated with x. A transaction is a set of weighted items, each of which may appear in multiple transactions with different weights.

3.1 Weight settings

Definition-1 Weighted attributes: *weighting attributes* $A(a_1, a_2, ..., a_k)$ are variables selected to calculate weights. Depending on the domain, there could be any variable ranging from item's price in a supermarket domain to visitor page dwelling time in a web log mining domain.

There are two types of weights – the *item weight* and the *itemset weight*:

Definition-2 Item weight: *Item weight* is a value attached to an item representing its significance. We denote it as *w(i)*. For example, in a supermarket setting, it could be the profit per unit sale of a certain item. In the web log mining setting where each item is a page visited in a click-stream/transaction, the weight can be related to a users average dwelling time on that page. In other words, the item weight is a function of selected weighting attributes therefore denoted as *w(i) = f(a)*.

Definition-3 Itemset weight: Based on the item weight w(i), the weight of an itemset, denoted as w(is), can be derived from the weights of its enclosing items. One simple way is to calculate the average value of the item weights, denoted as:

$$w(is) = \frac{\sum_{k=1}^{|is|} w(i_k)}{|is|}$$

Also bear in mind that an item weight is a special itemset weight when the itemset has only one item.

Definition-4 Transaction weight: *Transaction weight* is a type of itemset weight. It is a value attached to each of the transactions. Usually the higher a transaction weight, the more it contributes to the mining result. In a supermarket scenario, the weight can be the "significance" of a customer who made a certain transaction.

3.2 Weighting spaces

Items can be weighted within different weighting spaces depending on different scenarios and mining focus.

Definition-5 Weighting space: *weighting space WS* is the context within which the weights are evaluated

(1*) Inner-transaction* space *WSt:* this space refers to the host transaction that an item is weighted in.

(2*) Item space* WS_I: this space refers to the space of the item collection that covers all the items appears in the transactions.

(3) *Transaction* space WS_T: This space is defined for transactions rather than for items.

4. Improved Model - Weighted Association Rule Mining

In order to make use of the weight in the mining process, several new concepts have been adapted. Support is used in association rule mining. In weighted association rule mining (WARM), itemsets are no longer simply counted as they appear in a transaction. This change of counting mechanism makes it necessary to adapt traditional support to weighted support. The goal of using weighted support is to make use of the weight in the mining process and prioritize the selection of target itemsets according to their significance in the dataset, rather than their frequency alone.

4.1 Weighted support – significant framework vs. support – large framework

An itemset is denoted *large* if its support is above a pre-defined minimum support threshold. In the WARM context, we say an itemset *is significant* if its *weighted support* is above a pre-defined minimum weighted support threshold.

In fact, the threshold values specified by the user are from the margin of significance of cost point of view. This method may be more meaningful than only specifying relatively arbitrary support threshold. For example, in the supermarket scenario, suppose we assign a weight to each of the items according to

the profit it generates to the store, rather than simply counting and calculating the percentage of transactions that contain itemset. We calculate this according to the weighted support

Definition-6 Weighted support: Weighted support *WSP* of an itemset. A set of transactions T respects a rule R in the form A → B, where A and B are non-empty sub-itemsets of the item space I and they share no item in common. Its weighted support is the fraction of the weight of the transactions that contains both A and B relative to the weight of all transactions. This can be formulated as:

$$wsp\ (AB) = \frac{\sum_{k=1}^{|WS_T|\&(A\cup B)\subseteq t_k} weight\ (t_k)}{\sum_{k=1}^{|WS_T|} weight\ (t_k)}$$

By this means, weighted support is modelled to quantify the actual quota of an itemset in the transaction space in weighted association rule mining scenario.

The weighted support of an itemset can be defined as the product of the total weight of the itemset (sum of the weights of its items) and the weight of the fraction of transactions that the itemset occurs in. The goal of the weighted association rule mining is then changed to determining all rules that are above a user specified minimum weighted support threshold holding a minimum user specified confidence. In order to calculate weighted support of an itemset, we need a method to evaluate transaction weight.

The transaction weight (t_k) can be derived from weights of the items presented in the transaction. One may formulate it easily as the average weight of the items presented in the transaction. Note that $WS_t(t_k)$ denotes the inner-transaction space for the k_{th} transaction in transaction space WS_T.

$$weight(t_k) = \frac{\sum_{i=1}^{|WS_t(t_k)|} weight(item(i))}{|WS_t(t_k)|}$$

This value is used to calculate the weighted support of a potentially significant itemset described in Definition-6. The itemset is then validated as significant if its weighted support is above the pre-defined minimum weighted support. This is further described in the following section relating to algorithm design.

4.2 Challenges – Invalidation of the downward closure property

The critical assumption made in the Apriori algorithm is that if the itemset is large, then all its subsets are also large. This allows the algorithm to build large itemsets of increasing size by adding items to itemsets that are already found to be large.

In the case where item weight is used to adjust support values of the potentially large itemsets, the situation turns out to be considerably more complex. The assumption discussed in the preceding subsection does no longer hold. Because of the adjustment of the support, an itemset may be large even though some of its subsets are not large. This violates against the downward closure property as can be illustrated in Figure 1. This also demonstrates that we cannot simply use the weight to bias the support value in the mining process.

As shown in Figure 1, the weights of item A, C and D are deliberately biased so that A and C represent something of less important while D's relatively high weight granting it more significance in the item space. We now inspect two large

min_sup= 0.3	
items	weight
A	0.85
B	1
C	0.85
D	1.55
E	1

transactions				
A	B	C	D	
B	D			
A	D			
A	B	D	E	
A	B	C	D	E
B	C	E		

large 2&3 itemsets	weight	support	large?	by weight	large?
AC	0.85	0.333333	yes	0.283333	no
AD	1.2	0.666667	yes	0.8	yes
ACD	1.0833333	0.333333	yes	0.361111111	yes

Figure 1 support adjusted by item weight

itemsets of size two and one large itemset of size 3 which is the combination of two *large_2_itemsets*. In traditional ARM when the weight is not considered, all of the three itemsets are large as their supports are above the threshold min_sup. However, if we consider item weights, calculate the weights of itemset according to Definition-3 and bias the support by multiplying it with the itemset weight, a new set of adjusted support (*AS*) values are obtained. Figure 1 shows that although the AS of itemset "AC" is now below the minimum support (0.3) and therefore "AC" no longer large, we cannot rule out the possibility of its superset, "ACD" being large as we do in traditional ARM. In this example, the high weight of item D gives rise to the weight of itemset "ACD" which in the end biases its adjusted support to be above the minimum support.

The violation of the "downward closure property" was also addressed in [10] where a factor is assigned to adjust the minimum support threshold accordingly so that it relaxes the border restrained by the downward closure property. However, the degree of the relaxation varies in different circumstances. This requires a very delicate mechanism to provide a suitable value of the factor and in many cases, as the author also implies, it is extremely difficult to find a generic mechanism to determine the relaxation factor.

4.3 Weighted downward closure property

In this paper, the idea of replacing the support with significance is proposed for the first time and we argue that a "*weighted downward closure property*" can be retained by using weighted support.

As illustrated in Figure 2, items in the transaction dataset are assigned with weights. We use the similar approach of building lattice tree for significant itemsets, i.e., itemsets with weighted support above threshold.

As can be noted in Figure 3, for each itemset, weighted support (the number at the bottom of each itemset box) is calculated by using the formula given in Definition-6. If an itemset's weighted support is below the threshold, the itemset is not significant and we mark it in dotted background comparing to the broken edge which means that it is not large (support below threshold).

Items/weight					
Items	A	B	C	D	E
Weight (by WS_i)	1	1.1	1.03	1.02	1.5

Database of transactions						
Transaction	1	2	3	4	5	6
Itemset	ABCD	BDE	ABCD	ABDE	ABCDE	BCE
Transaction weight weight(t)	1.0375	1.207	1.0375	1.155	1.13	1.21

$$weight(t_k) = \frac{\sum_{i=1}^{|WS_t(t_k)|} weight(item(i))}{|WS_t(t_k)|}$$

Figure 2 Data source with weights

As may be noted, if an itemset is marked with dotted background, then any of its supersets in the upper layer of the lattice can not be significant. This property, denoted as "weighted downward closure property", is valid under the "weighted support – significant" framework. It justifies the efficient mechanism of generating and pruning significance iteratively. We will give the algorithm in section 5.

We also briefly prove that the "weighted downward closure property" is always valid in the "weighted support – significant" framework.

Proof: Given an itemset *X* not significant over the transaction space WS_T, i.e., $wsp(X) < \min\_wsp$. For any itemset *Y*, $X \subset Y$, i.e. superset of *X*, if a transaction *t* has all the items in Y, i.e. $Y \subset t$, then that transaction must also have all the items in X, i.e. $X \subset t$. We use *tt1* to denote a set of transactions each of which has all the items in X, i.e. $\{tt1 | tt1 \subseteq T, (\forall t \in tt1, X \subset t)\}$. Similarly we have $\{tt2 | tt2 \subseteq T, (\forall t \in tt2, Y \subset t)\}$. Since $X \subset Y$, we have $tt1 \subseteq tt2$. Therefore $\sum_{t \in tt1} Weight(t) \geq \sum_{t \in tt2} Weight(t)$. According to the definition of weighted support, $wsp(X) = \frac{\sum_{t \in T}^{X \subset t} Weight(t)}{\sum_{t \in T} Weight(t)}$, the denominator stays the same, therefore we have $wsp(X) \geq wsp(Y)$. Because *wsp(X)* < *min_wsp*, we get *wsp(Y)* < *min_wsp*, we have proved that Y is not significant.

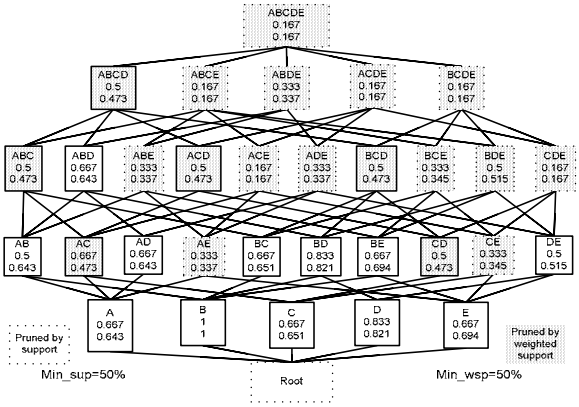

Figure 3 The lattice of significant itemsets using weighted downward closure property

We provide a concrete example to illustrate this in Figure 3. Itemset AC appears in transaction 1, 3 and 5 in Figure 2, therefore the WSP(AC)=(1.0375+1.0375+1.13)/6.777=0.473. It can be easily found that the occurrence of its superset ACE is only possible when AC appears in that transaction. In this case, itemset set ACE only appears in transaction 5, therefore WSP(ACE)=1.13/6.667=0.169, which is obviously less than WSP(AC), so if AC is not significant, its superset ACE is impossible to be significant, hence there is no need to calculate its weighted support.

5. Simulation

In this section, we use Excel to simulate the process of constructing significant itemset lattice. This simulation not only helps analyzing the new model behavior but also illustrates the key operations in weighted association rule mining.

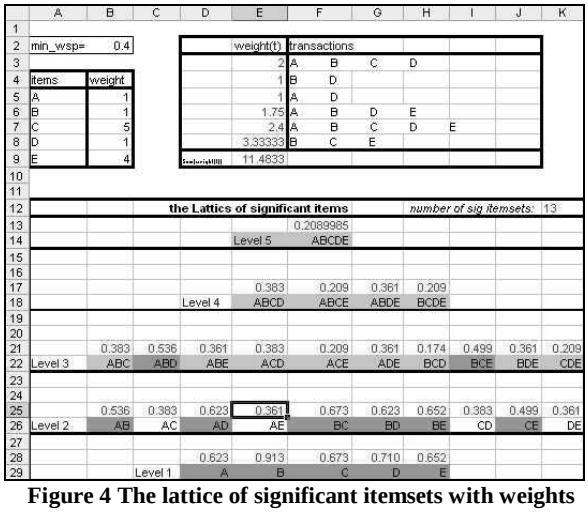

Figure 4 The lattice of significant itemsets with weights adjustment (Scheme 1 - high weights for "C","E")

In Figure 4, item C and E are assigned relatively high weights, which are denoted as weighting scheme 1. This is compared to simulation illustrated in Figure 5 where items A and C are highlighted with relatively high weights (weighting scheme 2).

min_wsp= 0.4

items	weight	nomalized over WS$_I$
A	5	0.385
B	1	0.077
C	5	0.385
D	1	0.077
E	1	0.077
sum	13	1.000

original transaction weight weight(t_k)	Transaction weight using normed item weight weight'(t_k)	nomalized over WS$_T$	transactions
3	0.231	0.215	A B C D
1	0.077	0.072	B D
3	0.231	0.215	A D
2	0.154	0.144	A B D E
2.6	0.200	0.187	A B C D E
2.33333333	0.179	0.167	B C E
sum 13.9333333	1.072	1.000	

the Lattics of significant items *number of sig itemsets:* 16

					10.51%					
					0.187					
				Level 5	ABCDE					
				17.15%	10.51%	-0.87%	10.51%			
				0.402	0.187	0.330	0.187			
			Level 4	ABCD	ABCE	ABDE	BCDE			
	17.15%	8.33%	-0.87%	17.15%	10.51%	-0.87%	22.44%	5.95%	-0.87%	10.51%
	0.402	0.545	0.330	0.402	0.187	0.330	0.215	0.354	0.330	0.187
Level 3	ABC	ABD	ABE	ACD	ACE	ADE	BCD	BCE	BDE	CDE
	8.33%	17.15%	12.33%	-0.87%	12.18%	-8.06%	-0.48%	17.15%	5.95%	-0.87%
	0.545	0.402	0.761	0.330	0.569	0.617	0.498	0.402	0.354	0.330
Level 2	AB	AC	AD	AE	BC	BD	BE	CD	CE	DE
			12.33%	-6.16%	12.18%	0.00%	-0.48%			
			0.761	0.785	0.569	0.833	0.498			
		Level 1	A	B	C	D	E			

Figure 5 The lattice of significant itemsets with weights adjustment (Scheme 2 - high weights for "A","C")

The structure of the algorithm resembles the Apriori [2], the different point is the use of weighted support justified by the "weighted downward closure property" under the adapted framework. Due to the space limitation, interested readers are suggested to read [11] for detailed information.

It can be concluded that the by assigning weights to items and using WARM, the selection of significant itemsets is steered to those itemsets containing or having relationships to high weight items.

6. Experiments

Various synthetic datasets are generated using procedure described in [1]. Part of the items in the item space are selected and assigned with a relatively high weight. X-axis refers to the high weight used. Y-axis denotes the number of significant itemsets generated using WARM. Scalability is also studied by scale up dataset size.

6.1 Selection of significant itemsets

Figure 6 lists three of the graphs. Different datasets are used for different diagrams. The number of significant itemsets is illustrated in terms of the high weight being assigned to parts of the items. As can be noticed, increasing the high weight doesn't necessarily increase the overall amount of significant itemsets; rather, it always makes those itemsets containing high weight items more likely to have a higher weighted support, hence holding more chances to become significant. Those itemsets containing no high weight items become relatively less likely to be significant.

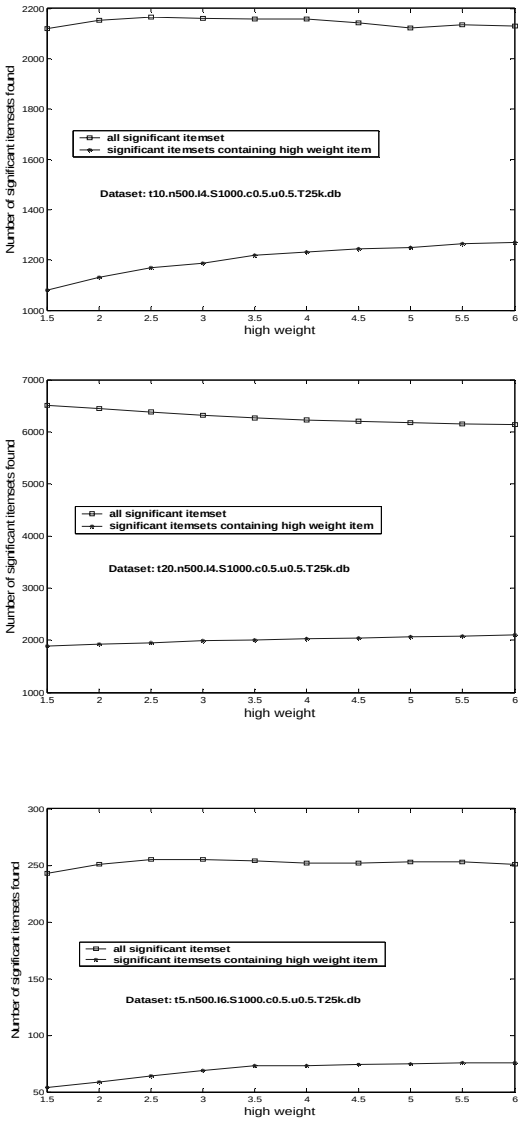

Figure 6 Number of significant itemsets biased with weight in various datasets

6.2 Scalability

We now show how the weighted associations rule mining scales up as the dataset size increases. Two factors are used to increase the dataset size. In Figure 7, the number of transactions increases from 0.1 million to 1 million. The datasets used for this experiment are t5.n1000.I4.S1000.c0.5.u0.5.T(100k-1000k).db.

In Figure 8, the average transaction size varies from 10 to 50. Five different minimum weighted supports ranging from 0.1% to 5% are used for both cases. The times are relative time which has been normalized against the first data point across

the x-axis. The graph shows that the new algorithm scales approximately linearly.

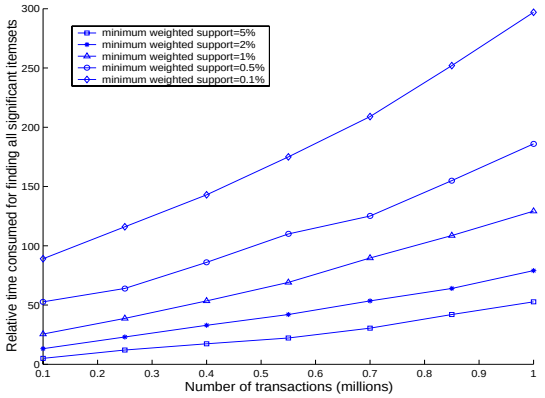

Figure 7 Scale-up experiment - number of transactions

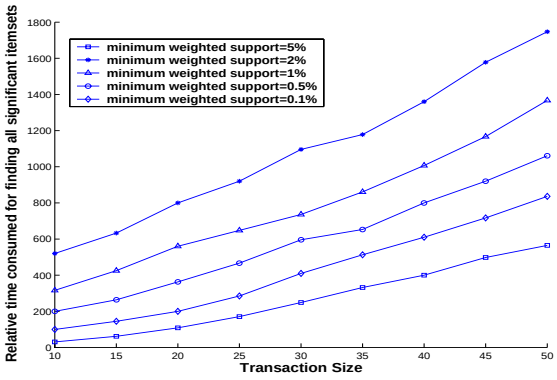

Figure 8 Scale-up experiment - average transaction size

7. Summary and conclusions

In this paper, we identify the limitation of the traditional Association Rule Mining model, in particular, its incapacity for treating units differently. We proposed that weight can be integrated in the mining process to solve this problem. We identify the challenge faced when making improvement towards using weight, in particular the invalidation of downward closure property.

A set of new concepts are proposed to adapt weighting in the new setting. Among them is the proposal of using “weighted downward closure property” as a replacement of the original “downward closure property”. This is proved as valid and justifies the effective mining strategy in the new framework of “weighted support – significant”. The new framework is designed to replace the original “support – large” framework in order to tackle the problem in weighted settings.

Through studying the simulation of the lattice building, conclusion is drawn that weight can be used to steer the mining focus to those important itemsets with high degree of significance. This is further proven by experiments on synthetic datasets. The experiments show that the mining results in the weighted setting conform to the expected hypothesis. The experiments also show that the algorithm is scalable.

References

1. R.Agrawal et al, "*The Quest Data Mining System*" Technical report, *IBM Almaden Research Center*, http://www.almaden.ibm.com/cs/quest/, 1996.

2. R.Agrawal, T.Imielinski, and A.Swami, "*Mining association rules between sets of items in large databases*", Proc. of the 1993 ACM SIGMOD Int’l Conf. on Management of Data, Washington, DC, 1993, pp. 207.

3. R.Agrawal and R.Srikant, "*Fast algorithms for mining association rules in large databases*", Proc. of the 20th Inte’l Conf. on Very Large Data Bases (VLDB'94), Santiago, Chile, 1994, pp. 487-499.

4. Fernando Berzal, Juan C. Cubero, Nicolas Marín, José-María Serrano, "TBAR: An efficient method for association rule mining in relational databases," *Data & Knowledge Engineering*, Vol. 37, No. 1, 2001, pp. 47-64.

5. Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, Shalom Tsur, "*Dynamic itemset counting and implication rules for market basket data*", Proc. of the ACM SIGMOD Int’l Conf. on Management of Data, Tucson,AZ, USA, 1997.

6. Toon Calders and Bart Goethals, "*Mining All Non-Derivable Frequent Itemsets*", Proc. of the 6th European Conf. on Principles of Data Mining and Knowledge Discovery, 2002, pp. 74-85.

7. Jiawei Han and Yongjian Fu, "*Discovery of Multiple-Level Association Rules from Large Databases* " in the Proceedings of the 1995 Int’l Conf. on Very Large Data Bases (VLDB'95), Zurich, Switzerland, 2002, pp. 420-431.

8. Bing Liu, Wynne Hsu, and Yiming Ma, "*Mining Association Rules with Multiple Supports*", Proc. of the ACM SIGKDD Int’l Conf. on Knowledge Discovery and Data Mining (KDD-99), SanDiego, CA, USA, 1999.

9. N.Pasquier, Y.Bastide, R.Taouil, and L.Lakhal, "Efficient mining of association rules using closed itemset lattices," *Information Systems*, Vol. 24, No. 1, 1999, pp. 25-46.

10. G.D.Ramkumar, Sanjay Ranka, and Shalom Tsur, "*Weighted Association Rules: Model and Algorithm*" KDD1998, 1998.

11. Feng Tao, "*Mining Binary Relationships from Transaction Data in Weighted Settings*" PhD Thesis, School of Computer Science, Queen's University Belfast, UK, 2003.

12. W. Wang, J. Yang and P. Yu "*Efficient mining of weighted association rules (WAR)*", Proc. of the ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, 270-274, 2000.