Weighted Association Rule Mining from Binary and Fuzzy Data

M. Sulaiman Khan[1], Maybin Muyeba[1], and Frans Coenen[2]

[1] School of Computing, Liverpool Hope University, Liverpool, L16 9JD, UK
[2] Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK
{khanm, muyebam}@hope.ac.uk, frans@csc.liv.ac.uk

Abstract. A novel approach is presented for mining weighted association rules (ARs) from binary and fuzzy data. We address the issue of invalidation of downward closure property (DCP) in weighted association rule mining where each item is assigned a weight according to its significance w.r.t some user defined criteria. Most works on weighted association rule mining so far struggle with invalid downward closure property and some assumptions are made to validate the property. We generalize the weighted association rule mining problem for databases with binary and quantitative attributes with weighted settings. Our methodology follows an Apriori approach [9] and avoids pre and post processing as opposed to most weighted association rule mining algorithms, thus eliminating the extra steps during rules generation. The paper concludes with experimental results and discussion on evaluating the proposed approach.

Keywords: Association rules, fuzzy, weighted attributes, apriori, downward closure.

1 Introduction

Association rules (ARs) [11] are a popular data mining technique used to discover behaviour from market basket data. The technique tries to association rules (with strong support and high confidence) in large databases. Classical Association Rule Mining (ARM) deals with the relationships among the items present in transactional databases [9, 10]. Typically, the algorithm first generates all large (frequent) itemsets (attribute sets) from which association rule (AR) sets are derived. A large itemset is defined as one that occurs more frequently in the given data set than a user supplied support threshold. To limit the number of ARs generated, a confidence threshold is used to limit the number of ARs generated by careful selection of the support and confidence thresholds. By so doing, care must be taken to ensure that itemsets with low support but from which high confidence rules may be generated are not omitted.

Given a set of items $I = \{i_1, i_2, ...i_m\}$ and a database of transactions $D = \{t_1, t_2, ...t_n\}$ where $t_i = \{I_{i1}, I_{i2}, ...I_{ip}\}$, $p \leq m$ and $I_{ij} \in I$, if $X \subseteq I$ with k = |X| is called a k-itemset or simply an itemset. Let a database D be a multi-set

P. Perner (Ed.): ICDM 2008, LNAI 5077, pp. 200–212, 2008.

of subsets of I as shown. Each $T \in D$ supports an itemset $X \subseteq I$ if $X \subseteq T$ holds. An association rule is an expression X => Y, where X, Y are item sets and $X \cap Y = \varnothing$ holds. Number of transactions T supporting an item X w.r.t D is called support of X, $Supp(X) = |\{T \in D \mid X \subseteq T\}| / |D|$. The strength or confidence (c) for an association rule X → Y is the ratio of the number of transactions that contain $X \cup Y$ to the number of transactions that contain X, $Conf(X \rightarrow Y) = Supp(X \cup Y) / Supp(X)$. For non-binary items, fuzzy association rule mining was proposed using fuzzy sets such that quantitative and categorical attributes can be handled [12]. A fuzzy quantitative rule represents each item as (item, value) pair. Fuzzy association rules are expressed in the following form:

If X is A satisfies Y is B

For example,

if (age is young) ➔ (salary is low)

Given a database *T*, attributes *I* with itemsets $X \subset I, Y \subset I$ and $X = \{x_1, x_2, ...x_n\}$ and $Y = \{y_1, y_2, ...y_n\}$ and $X \cap Y = \varnothing$, we can define fuzzy sets $A = \{fx_1, fx_2, ..., fx_n\}$ and $B = \{fx_1, fx_2, ..., fx_n\}$ associated to X and Y respectively. For example (X, A) could be *(age, young)*, *(age, old)*, *(salary, high)* etc. The semantics of the rule is that when the antecedent "X is A" is satisfied, we can imply that "Y is B" is also satisfied, which means there are sufficient records that contribute their votes to the attribute fuzzy set pairs and the sum of these votes is greater than the user specified threshold.

However, classical ARM framework assumes that all items have the same significance or importance i.e. their weight within a transaction or record is the same (weight=1) which is not always the case. For example, from table 1, the rule [printer → computer, 50%] may be more important than [scanner → computer, 75%] even though the former holds a lower support because those items in the first rule usually come with more profit per unit sale. In contrast, standard ARM simply ignores this difference.

The main challenge in weighted ARM is validating the "downward closure property (DCP)" which is crucial for the efficient iterative process of generating and pruning frequent itemsets from subsets.

Table 1. Weigted items database

ID	Item	Profit	Weight	…
1	Scanner	10	0.1	…
2	Printer	30	0.3	…
3	Monitor	60	0.6	…
4	Computer	90	0.9	…

Table 2. Transactions

TID	Items
1	1,2,4
2	2,3
3	1,2,3,4
4	1,3,4

In this paper we address the issue of DCP in Weighted ARM. We generalize and solve the problem of downward closure property for databases with binary and quantitative items and evaluate the proposed approach with experimental results.

The paper is organised as follows: section 2 presents background and related work; section 3 gives problem definition for weighted ARM with binary and fuzzy data and details weighted downward closure property; section 4 gives frameworks comparison; section 5 reviews experimental evaluation and section 8 concludes paper.

2 Background and Related Work

In literature on association rule mining, weights of items are mostly treated as equally important i.e. weight one (1) is assigned to each item until recently where some approaches generalize this and give items weights to reflect their significance to the user [4]. The weights may be attributed to occasional promotions of such items or their profitability etc. There are two approaches for analyzing data sets with weighted settings: pre- and post-processing. Post processing handles firstly the non-weighted problem (weights=1) and then perform the pruning process later. Pre-processing prunes the non-frequent itemsets after each iteration using weights. The issue in post-processing weighted ARM is that first; items are scanned without considering their weights and later, the rule base is checked for frequent weighted ARs. By doing this, we end up with a very limited itemset pool to check weighted ARs and potentially missing many itemsets.

In pre-processed classical ARM, itemsets are pruned by checking frequent ones against weighted support after every scan. This results in less rules being produced as compared to post processing because many potential frequent super sets are missed. In [2] a post-processing model is proposed. Two algorithms were proposed to mine itemsets with normalized and un-normalized weights. The k-support bound metric was used to ensure validity of the DCP but still there is no guarantee that every subset of a frequent set will be frequent unless the k-support bound value of (k-1) subsets was higher than (k).

An efficient mining methodology for Weighted Association Rules (WAR) is proposed in [3]. A Numerical attribute was assigned for each item where the weight of the item was defined as part of a particular weight domain. For example, soda[4,6] → snack[3,5] means 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 post-processing approach by deriving the maximum weighted rules from frequent itemsets. Post WAR doesn't interfere with the process of generating frequent itemsets but focuses on how weighted AR's can be generated by examining weighting factors of items included in generated frequent itemsets.

Similar techniques for weighted fuzzy quantitative association rule mining are presented in [5, 7, 8]. In [6], a two-fold pre processing approach is used where firstly, quantitative attributes are discretised into different fuzzy linguistic intervals and weights assigned to each linguistic label. A mining algorithm is applied then on the resulting dataset by applying two support measures for normalized and un-normalized cases. The closure property is addressed by using the z-potential frequent subset for

each candidate set. An arithmetic mean is used to find the possibility of frequent k+1itemset, which is not guaranteed to validate the valid downward closure property.

Another significance framework that handles the downward closure property (DCP) problem is proposed in [1]. Weighting spaces were introduced as inner-transaction space, item space and transaction space, in which items can be weighted depending on different scenarios and mining focus. However, support is calculated by only considering the transactions that contribute to the itemset. Further, no discussions were made on interestingness issue of the rules produced.

In this paper we present an approach to mine weighted binary and quantitative data (by fuzzy means) to address the issue of invalidation of DCP. We then show that using the proposed technique, rules can be generated efficiently with a valid DCP without any biases found in pre- or post-processing approaches.

3 Problem Definition

The problem definition consists of terms and basic concepts to define item's weight, itemset transaction weight, weighted support and weighted confidence for both binary (boolean attributes) and fuzzy (quantitative attributes) data. Technique for binary data is termed as Binary Weighted Association Rule Mining (BWARM) and technique for fuzzy data is termed as Fuzzy Weighted Association Rule mining (FWARM). Interested readers can see [14] for the formal definitions and more details.

3.1 Binary Weighted Association Rule Mining (BWARM)

Let the input data D have transactions $T = \{t_1, t_2, t_3, \cdots, t_n\}$ with a set of items $I = \{i_1, i_2, i_3, \cdots, i_{|I|}\}$ and a set of positive real number weights $W = \{w_1, w_2, \cdots, w_{|I|}\}$ attached to each item i . Each i^{th} transaction t_i is some subset of I and a weight w is attached to each item $t_i[i_j]$ (" j^{th} " item in the " i^{th} " transaction).

Thus each item i_j will have associated with it a weight corresponding to the set W , i.e. a pair (i, w) is called a weighted item where $i \in I$. Weight for the " j^{th} " item in the " i^{th} " transaction is given by $t_i[i_j[w]]$.

Table 3. Transactional database

T	Items	T	Items
t_1	A B C D	T_6	A B C D E
t_2	B D	T_7	B C E
t_3	A D	T_8	D E
t_4	C	T_9	A C D
t_5	A B D E	T_{10}	B C D E

Table 4. Items with weights

Items *i*	Weights *(IW)*
A	0.60
B	0.90
C	0.30
D	0.10
E	0.20

We illustrate the terms and concepts using tables 3 and 4. Table 3 contains 10 transactions for 5 items. Table 4 has corresponding weights associated to each item *i* in *T*. We use sum of votes for each itemset by aggregating weights per item as a standard approach.

Item Weight IW is a non-negative real value given to each item i_j ranging [0..1] with some degree of importance, a weight $i_j[w]$.

Itemset Transaction Weight ITW is the aggregated weight of all the items in the itemset present in a single transaction. Itemset transaction weight for an itemset X can calculated as:

$$\text{vote for } t_i \text{ satisfying } X = \prod_{k=1}^{|X|} {}_{(\forall [i[w]] \in X)}\, t_i[i_k[w]] \tag{1}$$

Itemset transaction weight of itemset (A, B) is calculated as: $ITW(A,B) = 0.6 \times 0.9 = 0.54$.

Weighted Support WS is the aggregated sum of itemset transaction weight ITW of all the transactions in which itemset is present, divided by the total number of transactions. It is calculated as:

$$WS(X) = \frac{\sum_{i=1}^{n} \prod_{k=1}^{|X|} {}_{(\forall [i[w]] \in X)}\, t_i[i_k[w]]}{n} \tag{2}$$

WS of itemset (A, B) is calculated as: $\frac{1.62}{10} = 0.16$

Weighted Confidence WC is the ratio of sum of votes satisfying both $\mathrm{X} \cup \mathrm{Y}$ to the sum of votes satisfying X. It is formulated (with $Z = X \cup Y$) as:

$$WC(X \rightarrow Y) = \frac{WS(Z)}{WS(X)} = \sum_{i=1}^{n} \frac{\prod_{k=1}^{|Z|} {}_{(\forall [z[w]] \in Z)}\, t_i[z_k[w]]}{\prod_{k=1}^{|X|} {}_{(\forall [i[w]] \in X)}\, t_i[x_k[w]]} \tag{3}$$

Weighted Confidence of itemset (A, B) is calculated as: $WC(A,B) = \frac{0.16}{0.30} = 0.54$

3.2 Fuzzy Weighted Association Rule Mining (FWARM)

A fuzzy dataset D' consists of fuzzy transactions $T' = \{t'_1, t'_2, t'_3, ..., t'_n\}$ with fuzzy sets associated with each item in $I = \{i_1, i_2, i_3, \cdots, i_{|I|}\}$, which is identified by a set of

Table 5. Fuzzy transactional database

TID	X		Y	
	Small	**Medium**	**Small**	**Medium**
t_1	0.5	0.5	0.2	0.8
t_2	0.9	0.1	0.4	0.6
t_3	1.0	0.0	0.1	0.9
t_4	0.3	0.7	0.5	0.5

Table 6. Fuzzy items with weights

Fuzzy Items *i[l]*	Weights *(IW)*
(X, Small)	0.9
(X, Medium)	0.7
(Y, Small)	0.5
(Y, Medium)	0.3

linguistic *labels* $L = \{l_1, l_2, l_3, ..., l_{|L|}\}$ (for example $L = \{small, medium, l\arg e\}$). We assign a weight w to each l in L associated with i. Each attribute $t_i'[i_j]$ is associated (to some degree) with several fuzzy sets. The degree of association is given by a *membership degree* in the range $[0..1]$, which indicates the correspondence between the value of a given $t_i'[i_j]$ and the set of *fuzzy linguistic labels*. The "k^{th}" weighted fuzzy set for the "j^{th}" item in the "i^{th}" fuzzy transaction is given by $t_i'[i_j[l_k[w]]]$.

We illustrate the fuzzy weighted ARM definition terms and concepts using tables 5 and 6. Table 5 contains transactions for 2 quantitative items discretised into two overlapped intervals with fuzzy values. Table 6 has corresponding weights associated to each fuzzy item *i[l]* in *T*.

Fuzzy Item Weight FIW is a value attached with each fuzzy set. It is a non-negative real number value in $[0..1]$ wrt some degree of importance (table 6). Weight of a fuzzy set for an item i_j is denoted as $i_j[l_k[w]]$.

Fuzzy Itemset Transaction Weight $FITW$ is the aggregated weights of all the fuzzy sets associated with items in the itemset present in a single transaction. Fuzzy Itemset transaction weight for an itemset *(X, A)* can be calculated as:

$$\text{vote for } t_i' \text{ satisfying } X = \prod_{k=1}^{|L|}{}_{(\forall[i[l[w]]] \in X)}\, t_i'[i_j[l_k[w]]] \tag{4}$$

Let's take an example of itemset <(X, Medium), (Y, Small)> denoted by (X, Medium) as *A* and (Y, Small) as *B*. Fuzzy Itemset transaction weight $FITW$ of itemset (A, B) in transaction 1 is calculated as:

$$FITW(A,B) = (0.5 \times 0.7) \times (0.2 \times .05) = (0.35) \times (0.1) = 0.035$$

Fuzzy Weighted Support FWS is the aggregated sum of $FITW$ of all the transaction's itemsets present divided by the total number of transactions, represented as:

$$FWS(X) = \frac{\sum_{i=1}^{n} \prod_{k=1}^{|L|}{}_{(\forall[i[l[w]]] \in X)}\, t_i'[i_j[l_k[w]]]}{n} \tag{5}$$

FWS of itemset (A, B) is calculated as: $FWS(A,B) = \frac{0.297}{4} = 0.074$

Fuzzy Weighted Confidence FWC is the ratio of sum of votes satisfying both $X \cup Y$ to the sum of votes satisfying X with $Z = X \cup Y$ and given as:

$$FWC(X \rightarrow Y) = \frac{FWS(Z)}{FWS(X)} = \sum_{i=1}^{n} \frac{\prod_{k=1}^{|Z|} \left(\forall [z[w]] \in Z\right) t_i'[z_k[w]]}{\prod_{k=1}^{|X|} \left(\forall [i[w]] \in X\right) t_i'[x_k[w]]} \quad (6)$$

FWC of itemset (A, B) is calculated as: $FWC(A,B) = \frac{0.074}{0.227} = 0.325$

3.3 Weighted Downward Closure Property (DCP)

In classical ARM algorithm, it is assumed that if the itemset is large, then all its subsets should be large, a principle called downward closure property (DCP). For example, in classical ARM using DCP, it states that if AB and BC are not frequent, then ABC and BCD cannot be frequent , consequently their supersets are of no value.as they will contain non-frequent itemsets. This helps algorithm to generate large itemsets of increasing size by adding items to itemsets that are already large. In the weighted ARM where each item is given a weight, the DCP does not hold in a straightforward manner. Because of the weighted support, an itemset may be large even though some of its subsets are not large and we illustrate this in table 7.

In table 7, all frequent itemsets are generated using 30% support threshold. In column two, itemset {ACD} and {BDE} are frequent with support 30% and 36% respectively. And all of their subsets {AC}, {AD}, {CD} and {BD}, {BE}, {DE} are frequent as well. But in column 3 with weighted settings, itemsets {AC}, {CD} and {DE} are no longer frequent and thus violates the DCP.

We argue that the DCP with binary and quantitative data can be validated using the proposed approach. We prove this by showing that if an itemset is not frequent, then its superset cannot be frequent and $WS(subset) \geq WS(sueprset)$ is always true (see table 7, column 4, Proposed Weighted ARM, only the itemsets are frequent with frequent subsets). A formal proof of the weighted DCP can be found in [14].

4 Frameworks Comparison

In this section, we give a comparative analysis of frequent itemset generation between classical ARM, weighted ARM and the proposed binary and fuzzy ARM frameworks. In table 7 all the possible itemsets are generated using tables 3 and 4 (i.e. 31 itemsets from 5 items), and the frequent itemsets generated using classical ARM (column 2), weighted ARM (column 3) and proposed weighted ARM framework (column 4). Column 1 in table 7 shows itemset's ids.

Table 7. Frequent itemsets comparison

ID	Classical ARM	Classical Weighted ARM	Proposed Weighted ARM
1.	A (50%)	A (30%)	A (0.300)
2.	A→B (30%)	A→B (45%)	A→B (0.162)
3.	A→B→C (20%)	A→B→C (36%)	A→B→C (0.032)
4.	A→B→C→D (20%)	A→B→C→D (38%)	A→B→C→D (0.003)
5.	A→B→C→D→E (10%)	A→B→C→D→E (21%)	A→B→C→D→E (0.000)
6.	A→B→C→E (10%)	A→B→C→E (20%)	A→B→C→E (0.003)
7.	A→B→D (30%)	A→B→D (48%)	A→B→D (0.016)
8.	A→B→D→E (20%)	A→B→D→E (36%)	A→B→D→E (0.002)
9.	A→B→E (20%)	A→B→E (34%)	A→B→E (0.022)
10.	A→C (30%)	A→C (27%)	A→C (0.054)
11.	A→C→D (30%)	A→C→D (30%)	A→C→D (0.005)
12.	A→C→D→E (10%)	A→C→D→E (12%)	A→C→D→E (0.000)
13.	A→C→E (10%)	A→C→E (11%)	A→C→E (0.004)
14.	A→D (50%)	A→D (35%)	A→D (0.030)
15.	A→D→E (20%)	A→D→E (18%)	A→D→E (0.002)
16.	A→E (20%)	A→E (16%)	A→E (0.024)
17.	B (60%)	B (54%)	B (0.540)
18.	B→C (40%)	B→C (48%)	B→C (0.108)
19.	B→C→D (30%)	B→C→D (39%)	B→C→D (0.008)
20.	B→C→D→E (20%)	B→C→D→E (30%)	B→C→D→E (0.001)
21.	B→C→E (30%)	B→C→E (42%)	B→C→E (0.016)
22.	B→D (50%)	B→D (50%)	B→D (0.045)
23.	B→D→E (30%)	B→D→E (36%)	B→D→E (0.005)
24.	B→E (40%)	B→E (44%)	B→E (0.072)
25.	C (60%)	C (18%)	C (0.180)
26.	C→D (40%)	C→D (16%)	C→D (0.012)
27.	C→D→E (20%)	C→D→E (12%)	C→D→E (0.001)
28.	C→E (30%)	C→E (15%)	C→E (0.018)
29.	D (80%)	D (8%)	D (0.080)
30.	D→E (40%)	D→E (12%)	D→E (0.008)
31.	E (50%)	E (10%)	E (0.100)

A support threshold for classical ARM is set to 30% and for classical WARM and proposed Weighted ARM it is set to 0.3 and 0.03 respectively). Itemsets with a highlighted background indicate frequent itemsets. This experiment is conducted in order to illustrate the effect of item's occurrences and their weights on the generated rules.

Frequent itemsets in column 3 are generated using classical weighted ARM pre-processing technique. In this process all the frequent itemsets are generated first with count support and then those frequent itemsets are pruned using their weights. In this case only itemsets are generated from the itemset pool that is already frequent using their count support. Itemsets with shaded background and white text are those that WARM does not consider because they are not frequent using count support. But with weighted settings they may be frequent due to significance associated with them. Also, the generated itemsets do not hold DCP as described in sect. 3.2.

In column 4 frequent itemsets are generated using proposed weighted ARM framework. It is noted that the itemsets generated are mostly frequent using count support technique and interestingly included fewer rules like {AB→C} that is not

frequent, which shows that the non-frequent itemsets can be frequent with weighted settings due to their significance in the data set even if they are not frequent using count support.

In column 4, itemsets {A→B} and {B→C} are frequent due to high weight and support count in transactions. It is interesting to have a rule {B→D} because D has very low weight (0.1) but it has the highest count support i.e. 80% and it appears more with item B than any other item i.e. with 50% support. Another aspect to note is that, B is highly significant (0.9) with high support count (60%). These kinds of rules can be helpful in "Cross-Marketing" and "Loss Leader Analysis" in real life applications.

Also the itemsets generated using our approach holds valid DCP as shown in sect. 3.2. Table 7 gives a concrete example of our approach and we now perform experiments based on this analysis.

5 Experimental Evaluation

To demonstrate the effectiveness of the approach, we performed several experiments using a real retail data set [13]. The data is a binary transactional database containing 88,163 records and 16,470 unique items. Weights were generated randomly and assigned to all items in the dataset to show their significance.

Experiments were undertaken using four different association rule mining techniques. Four algorithms were used for each approach, namely Binary Weighted ARM (BWARM), Fuzzy Weighted ARM (FWARM), standard ARM as Classical ARM and WARM as post processing weighted ARM algorithm.

The BWARM and FWARM algorithms belongs to *breadth first traversal* family of ARM algorithms, uses tree data structures and works in fashion similar to the Apriori algorithm [9]. Both algorithms consist of several steps. For more details on algorithms and pseudo code, refer to [14].

In this paper, an improvement from [14] is that we have used a real dataset in order to demonstrate performance of the proposed approach. We performed two types of experiments based on quality measures and performance measures. For quality measures, we compared the number of frequent itemsets and the interesting rules generated using four algorithms described above. In the second experiment, we showed the scalability of the proposed BWARM and FWARM algorithms by comparing the execution time with varying user specified support thresholds.

5.1 Quality Measures

For quality measures, the binary retail dataset described above was used. Each item is assigned a weight range between $[0..1]$ according to their significance in the dataset. For fuzzy attributes we used approach described in [15] to obtain fuzzy dataset. With fuzzy dataset each attribute is divided into five different fuzzy sets.

In figure 1, the x-axis shows support thresholds from 2% to 10% and on the y-axis the number of frequent itemsets. Four algorithms are compared, BWARM (Binary Weighted ARM) algorithm using weighted binary datasets; FWARM (Fuzzy Weighted ARM) algorithm using fuzzy attributes and weighted fuzzy linguistic values; Classical ARM using standard ARM with binary dataset and WARM using

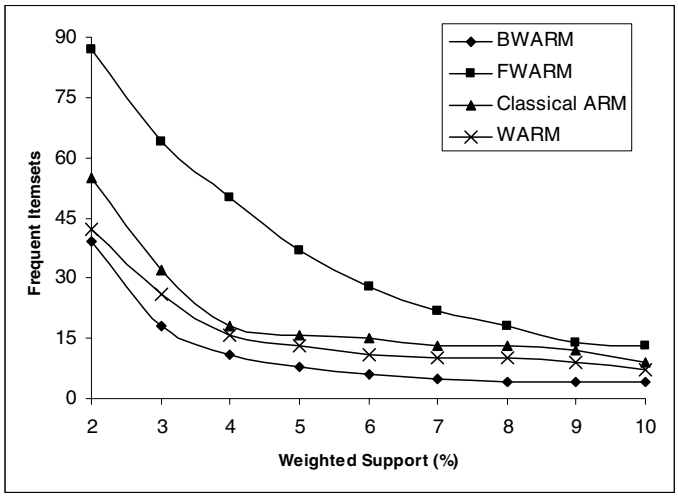

Fig. 1. No. of frequent Itemsets generated using user specified support threshold

weighted binary datasets and applying a post processing approach. Note that the weight of each item in classical ARM is 1 i.e. all items have equal weight.

The results show quite similar behavior of the three algorithms to classical ARM. As expected the number of frequent itemsets increases as the minimum support decreases in all cases. Number of frequent itemsets generated using the WARM algorithm are always less than the number of frequent itemsets generated by classical ARM because WARM uses only generated frequent itemsets in the same manner as classical ARM. This generates less frequent itemsets and misses many potential ones.

We do not use classical ARM approach to first find frequent itemsets and then re-prune them using weighted support measures. Instead all the potential itemsets are considered from the beginning for pruning using Apriori approach [9] in order to validate the DCP. Results of proposed BWARM approach are better than WARM because less arguably better frequent itemsets and rules are generated as we consider both itemset weights and their support count. Moreover, BWARM, classical ARM and WARM utilise binary data. FWARM generates more rules because of the extended fuzzy attributes, and it considers degree of membership instead of attribute presence only (count support) in a transaction.

Figure 2 shows the number of interesting rules generated using confidence measures. In all cases, the number of interesting rules is less because the interestingness measure generates fewer rules.

FWARM produces more rules due to the high number of initially generated frequent itemsets due to the introduction of more fuzzy sets for each quantitative attribute. Given a high confidence, BWARM outperforms classical WARM because the number of interesting rules produced is greater than WARM. This is because BWARM generates rules with items more correlated to each other and consistent at a higher confidence unlike WARM, where rules keep decreasing even at high confidence.

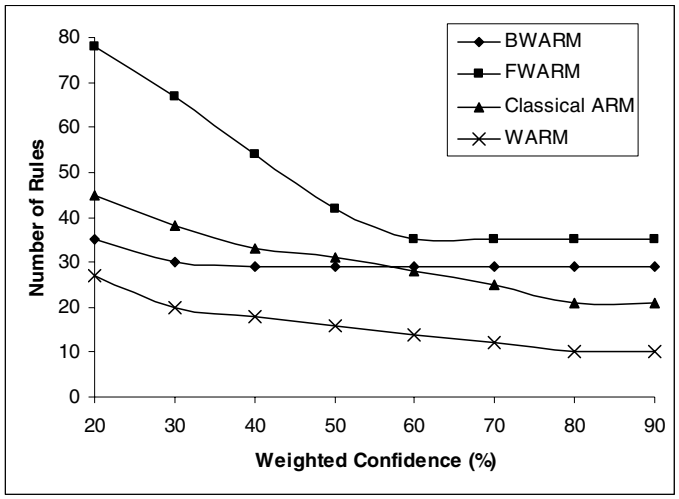

Fig. 2. No. of Interesting Rules generated using user specified confidence

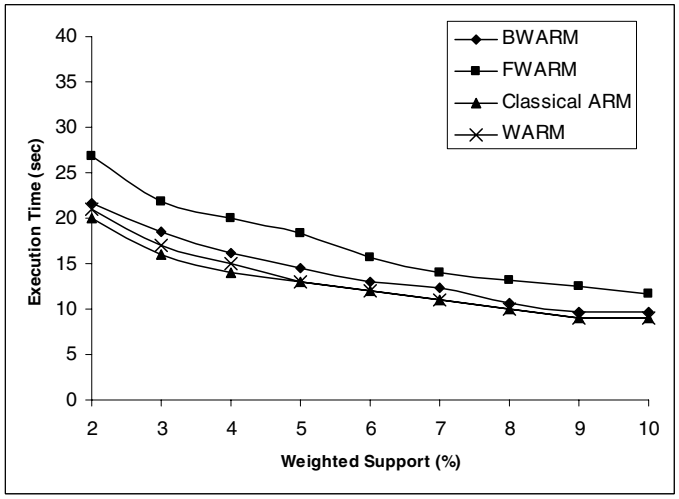

Fig. 3. Performance measures: varying weighted support (WS) threshold

The experiments show that the proposed framework produces better results as it uses all the possible itemsets and generates rules effectively using valid DCP. Further, the novelty is the ability to analyse both binary and fuzzy datasets with weighted settings.

5.2 Performance Measures

Experiment two compares the execution time of BWARM and FWARM algorithms with classical Apriori ARM and WARM algorithms. We investigated the effect on

execution time caused by varying the weighted support threshold with fixed data size (number of records). In figure 3, a support threshold from 2% to 10% is used again.

FWARM has comparatively higher execution time due to the fact that it deals with fuzzy data as mentioned earlier. Classical ARM and WARM have almost similar timings as WARM initially uses classical ARM approach and uses already generated frequent sets for post processing. Results show that BWARM has almost similar execution time to WARM. The minor difference is due to the way it generates frequent sets i.e. it considers items weights and their count support. Similarly from figure 3, it can be noted that BWARM and FWARM algorithms scale linearly with increasing weighted support threshold, which is similar behavior to Classical ARM.

6 Conclusion

We have presented a generalised approach for mining weighted association rules from databases with binary and quantitative (fuzzy) attributes. A classical model of binary and fuzzy association rule mining is adopted to address the issue of invalidation of downward closure property (DCP) in weighted association rule mining. The problem of invalidation of the DCP is solved using an improved model. We used classical and weighted ARM examples to compare support and confidence measures and evaluated the effectiveness of the proposed approach experimentally. We have demonstrated the valid DCP with formal comparisons with classical weighted ARM. It is notable that the approach presented here is effective in analysing databases with binary and fuzzy (quantitative) attributes with weighted settings.

References

1. Tao, F., Murtagh, F., Farid, M.: Weighted Association Rule Mining Using Weighted Support and Significance Framework. In: Proceedings of 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, pp. 661–666 (2003)
2. Cai, C.H., Fu, A.W.-C., Cheng, C.H., Kwong, W.W.: Mining Association Rules with Weighted Items. In: Proceedings of Intl. Database Engineering and Applications Symposium (IDEAS 1998), Cardiff, Wales, UK, July 1998, pp. 68–77 (1998)
3. Wang, W., Yang, J., Yu, P.S.: Efficient Mining of Weighted Association Rules (WAR). In: Proceedings of the KDD, Boston, August, pp. 270–274 (2000)
4. Lu, S., Hu, H., Li, F.: Mining Weighted Association Rules. Intelligent data Analysis Journal 5(3), 211–255 (2001)
5. Wang, B.-Y., Zhang, S.-M.: A Mining Algorithm for Fuzzy Weighted Association Rules. In: IEEE Conference on Machine Learning and Cybernetics, vol. 4, pp. 2495–2499 (2003)
6. Gyenesei, A.: Mining Weighted Association Rules for Fuzzy Quantitative Items. In: Proceedings of PKDD Conference, pp. 416–423 (2000)
7. Shu, Y.J., Tsang, E., Yeung, D.S.: Mining Fuzzy Association Rules with Weighted Items. In: IEEE International Conference on Systems, Man, and Cybernetics (2000)
8. Lu, J.-J.: Mining Boolean and General Fuzzy Weighted Association Rules in Databases. Systems Engineering-Theory & Practice 2, 28–32 (2002)
9. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: 20th VLDB Conference, pp. 487–499 (1994)

10. Bodon, F.: A Fast Apriori implementation. In: ICDM Workshop on Frequent Itemset Mining Implementations, Melbourne, Florida, USA, vol. 90 (2003)
11. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In: 12th ACM SIGMOD on Management of Data, pp. 207–216 (1993)
12. Kuok, C.M., Fu, A., Wong, M.H.: Mining Fuzzy Association Rules in Databases. SIGMOD Record 27(1), 41–46 (1998)
13. Brijs, T., Swinnen, G., Vanhoof, K., Wets, G.: The use of association rules for product assortment decisions: a case study. In: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, San Diego, pp. 254–260 (1999)
14. Sulaiman Khan, M., Muyeba, M., Coenen, F.: Fuzzy Weighted Association Rule Mining with Weighted Support and Confidence Framework. In: Proc. of ALSIP Workshop (PAKDD), Osaka, Japan (to appear, 2008)
15. Sulaiman Khan, M., Muyeba, M., Coenen, F.: On Extraction of Nutritional Patterns (NPS) Using Fuzzy Association Rule Mining. In: Proc. of Intl. Conference on Health Informatics (HEALTHINF 2008), Madeira, Portugal, vol. 1, pp. 34–42. INSTICC press (2008)