Fuzzy Weighted Association Rule Mining with Weighted Support and Confidence Framework

M. Sulaiman Khan[1], Maybin Muyeba[1], Frans Coenen[2]
[1] Liverpool Hope University, School of Computing, Liverpool, UK
[2] The University of Liverpool, Department of Computer Science, Liverpool, UK

{khanm@hope.ac.uk, muyebam@hope.ac.uk, frans@csc.liv.ac.uk}

Abstract.
In this paper we extend the problem of mining weighted association rules. A classical model of boolean and fuzzy quantitative association rule mining is adopted to 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 DCP so far struggle with invalid downward closure property and some assumptions are made to validate the property. We generalize the problem of downward closure property and propose a fuzzy weighted support and confidence framework for boolean and quantitative items with weighted settings. The problem of invalidation of the DCP is solved using an improved model of weighted support and confidence framework for classical and fuzzy association rule mining. Our methodology follows an Apriori algorithm approach and avoids pre and post processing as opposed to most weighted ARM algorithms, thus eliminating the extra steps during rules generation. The paper concludes with experimental results and discussion on evaluating the proposed framework.

Keywords: Association rules, fuzzy, weighted support, weighted confidence, downward closure.

1 Introduction

Association rules (ARs) [11] have been widely used to determine customer buying patterns from market basket data. The task of mining association rules is mainly to discover 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]. The typical approach is to first generate all large (frequent) itemsets (attribute sets) from which the set of ARs is 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. The number of ARs generated can therefore be influence by careful selection of the support and confidence thresholds, however great 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 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 U Y to the number of transactions that contain X, Conf (X → Y) = Supp (X U Y)/ Supp (X). For non-boolean 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, Y) 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, the above ARM framework assumes that all items have the same significance of importance i.e. their weight within a transaction or record is the same (weight=1) which 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 of 1%. 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.

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	2,3,4

Weighted ARM deals with the importance of individual items in a database [2, 3, 4]. For example, some products are more profitable or may be under promotion, therefore more interesting as compared to others, and hence rules concerning them are of greater value. Items are assigned weights (w) according to their significance as shown in table 1.

These weights may be set according to an item's profit margin. This generalized version of ARM is called Weighted Association Rule Mining (WARM). From table 1, we can see that the rule Computer→Printer is more interesting than Computer → Scanner because the profit of a printer is greater than that of a scanner. The main challenge in weighted ARM is that "downward closure property" which is crucial for efficient iterative process of generating and pruning frequent itemsets from subsets.

In this paper we address the issue of downward closure property in WARM. We generalize and solve the problem of downward closure property and propose a weighted support and confidence framework for both boolean and quantitative items for classical and fuzzy WARM (FWARM). We evaluate our proposed framework with experimental results.

The paper is organised as follows: section 2 presents background and related work; section 3 gives problem definition one; section 4 gives problem definition 2; section 5 details weighted downward closure property; section 6 reviews experimental results and section 7 concludes the paper with directions for future work.

2 Background and Related Work

Classical ARM data items are viewed as having equal importance but recently some approaches generalize this where items are given weights to reflect their significance to the user [4]. The weights may correspond to special promotions on some products or the profitability of different items etc. Currently, two approaches exist: pre- and post-processing. Post processing solves first the non-weighted problem (weights=1 per item) and then prunes the rules later. Pre-processing prunes the non-frequent itemsets earlier using weights after every iteration. The issue post-processed weighted ARM is that first, items are scanned without considering their weights. Finally, the rule base is checked for frequent weighted ARs. This gives us a very limited itemset pool to check weighted ARs and may miss many potential itemsets.

In pre-processing, classical ARM prunes itemsets by checking frequent ones against weighted support after every scan. In pre-processing, less rules are obtained 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 downward closure property. Even that didn't guarantee every subset of a frequent set being frequent unless the k-support bound value of (K-1) subset was higher than (K).

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 association rules can be generated by examining the weighting factors of the items included in generated frequent itemsets.

Similar techniques for weighted fuzzy quantitative association rule mining [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, WARM, that handles the DCP problem, is proposed [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 the confidence or interestingness issue of the rules produced

In this paper we present a fuzzy weighted support and confidence framework to mine weighted boolean and quantitative data (by fuzzy means) to address the issue of invalidation of downward closure property. We then show that using the proposed framework, rules can be generated efficiently with a valid downward closure property without biases made by pre- or post-processing approaches.

3 Problem Definition One (Boolean)

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 non-negative, real number weights $W = \{w_1, w_2, w_3, \cdots, w_{|I|}\}$ associated with each item. Each ith transaction t_i is some subset of I and a weight w is attached to each item $t_i[i_j]$ (the "jth" item in the "ith" transaction).

Table 3. Transactional Database

T	Items *I*
t_1	1 2 3 4 5
t_2	1 3 5
t_3	2 4
t_4	1 4 5
t_5	1 2 3 4

Table 4. Items with weights

Items *i*	Weights *(IW)*
1	.9
2	.7
3	.5
4	.3
5	.1

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 “jth” item in the “ith” transaction is given by $t_i[i_j[w]]$.

We illustrate the concept and definitions using tables 3 and 4. Table 3 contains transactions for 5 items. Table 4 has corresponding weights associated to each item *i* in *T*. In our definitions, we use sum of votes for each itemset by multiplying weight occurrence per item as a standard approach.

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

Definition 2 **Itemset Transaction Weight** ITW is the product of weights of all the items in the itemset present in a single transaction. Itemset transaction weight for an itemset X can calculated as:

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

Itemset transaction weight ITW of itemset (2, 4) is calculated as:

$$ITW(2,4) = .7 \times .3 = 0.21$$

Definition 3 **Weighted Support** WS is the sum of itemset transaction weight ITW of all the transactions in which itemset is present, divided by the total number of transactions. It is denoted as:

$$WS(X) = \frac{\text{Sum of votes satisfying } X}{\text{Number of records in } T} = \frac{\sum_{i=1}^{n} \prod_{k=1}^{|X|} {}_{(\forall[i[w]]\in X)}\, t_i[i_k[w]]}{n} \qquad \textbf{(2)}$$

Let’s take an example of itemset (2, 4), and find its itemset transaction weight, weighted support and weighted confidence. Weighted Support WS of itemset (2, 4) is calculated as:

$$WS(2,4) = \frac{\text{Sum of votes satisfying } (2,4)}{\text{Number of records in } T} = \frac{(.7 \times .3) + (.7 \times .3) + (.7 \times .3)}{5}$$

$$= \frac{.63}{5} = 0.126$$

Definition 4 **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 WC of itemset (2, 4) is calculated as:

$$WC(2,4) = \frac{WS(Z)}{WS(X)} = WC(2,4) = \frac{WS(X \cup Y)}{WS(X)} = WC(2,4) = \frac{0.126}{0.14} = 0.9$$

4 Problem Definition Two (Quantitative/Fuzzy)

Let a dataset D consists of a set of transactions $T = \{t_1, t_2, t_3, \cdots, t_n\}$ with a set of items $I = \{i_1, i_2, i_3, \cdots, i_{|I|}\}$. A fuzzy dataset D' consists of fuzzy transactions $T' = \{t'_1, t'_2, t'_3, \ldots, t'_n\}$ with fuzzy sets associated with each item in I, which is identified by a set of linguistic *labels* $L = \{l_1, l_2, l_3, \ldots, l_{|L|}\}$ (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 "kth" weighted fuzzy set for the "jth" item in the "ith" fuzzy transaction is given by $t'_i[i_j[l_k[w]]]$. Thus each label l_k for item i_j would have associated with it a weight, i.e. a pair $([i[l]], w)$ is called a weighted item where $[i[l]] \in L$ is a label associated with item i and $w \in W$ is the weight associated with label l.

Table 5. Fuzzy Transactional Database

TID	X		Y	
	Small	**Medium**	**Small**	**Medium**
1	0.5	0.5	0.2	0.8
2	0.9	0.1	0.4	0.6
3	1.0	0.0	0.1	0.9
4	0.3	0.7	0.5	0.5

Table 6. Fuzzy Items with weights

Fuzzy Items *i[l]*	Weights *(IW)*
(X, Small)	.9
(X, Medium)	.7
(Y, Small)	.5
(Y, Medium)	.3

We illustrate the fuzzy weighted ARM concept and definitions using tables 5 and 6. Table 5 contains transactions for 2 quantitative items further discretised into two overlapped intervals with fuzzy vales. Table 4 has corresponding weights associated to each fuzzy item *i[l]* in *T.*

Definition 5 Fuzzy Item Weight FIW is a value attached with each fuzzy set. It is a real number value range $[0..1]$ w.r.t some degree of importance (table 6). Weight of a fuzzy set for an item i_j is denoted as $i_j[l_k[w]]$.

Definition 6 Fuzzy Itemset Transaction Weight $FITW$ is the product of weights of all the fuzzy sets associated to items in the itemset present in a single transaction. Fuzzy Itemset transaction weight for an itemset *(X, A)* can be calculated as:

$$\text{vote for } t'_i \text{ satisfying } X = \prod_{k=1}^{|L|} {}_{(\forall[i[l[w]]]\in X)} t'_i[i_j[l_k[w]]] \qquad (\mathbf{4})$$

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 $FITW(A, B) = (.5\times.7)\times(.2\times.5) = (.35)\times(.1) = .035$

Definition 7 Fuzzy Weighted Support FWS is the sum of weight $FITW$ of all the transactions in which itemset is present, divided by the total number of transactions. It is denoted as:

$$FWS(X) = \frac{\text{Sum of votes satisfying } X}{\text{Number of records in } T} = \frac{\sum_{i=1}^{n}\prod_{k=1}^{|L|} {}_{(\forall[i[l[w]]]\in X)} t'_i[i_j[l_k[}{n} \qquad (\mathbf{5})$$

Weighted Support FWS of itemset (A, B) is calculated as:

$$FWS(A, B) = \frac{\text{Sum of votes satisfying (A, B)}}{\text{Number of records in } T}$$

$$= \frac{(.5\times.7)(.2\times.5) + (.1\times.7)(.4\times.5) + (.0\times.7)(.1\times.5) + (.7\times.7)(.5\times.5)}{4}$$

$$= \frac{.297}{4} = 0.074$$

Definition 8

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

$$FWC(X \rightarrow Y) = \frac{FWS(Z)}{FWS(X)} == \sum_{i=1}^{n} \frac{\prod_{k=1}^{|Z|} {}_{(\forall[z[w]]\in Z)} t'_i[z_k[w]]}{\prod_{k=1}^{|X|} {}_{(\forall[i[w]]\in X)} t'_i[x_k[w]]}$$

Fuzzy Weighted Confidence (FWC) of itemset (A, B) is calculated as:

$$FWC(A,B) = \frac{WS(Z)}{WS(X)} = \frac{WS(X \cup Y)}{WS(X)} = \frac{0.074}{0.227} = 0.325$$

5 Downward Closure Property (DCP)

In a classical Apriori algorithm it is assumed that if the itemset is large, then all its subsets should also be large and is called Downward Closure Property (DCP). This helps algorithm to generate large itemsets of increasing size by adding items to itemsets that are already large. In the weighted ARM case where each item is assigned weight, the DCP does not hold. Because of the weighted support, an itemset may be large even though some of its subsets are not large. This violates DCP (see table 7).

Table 7. Frequent itemsets with invalid DCP

T	Items
t_1	A B C D E
t_2	A C E
t_3	B D
t_4	A D E
t_5	A B C D

Items *i*	Weights (IW)
1	.1
2	.3
3	.6
4	.9
5	.7

min_supp=40%   weighted_supp=0.4

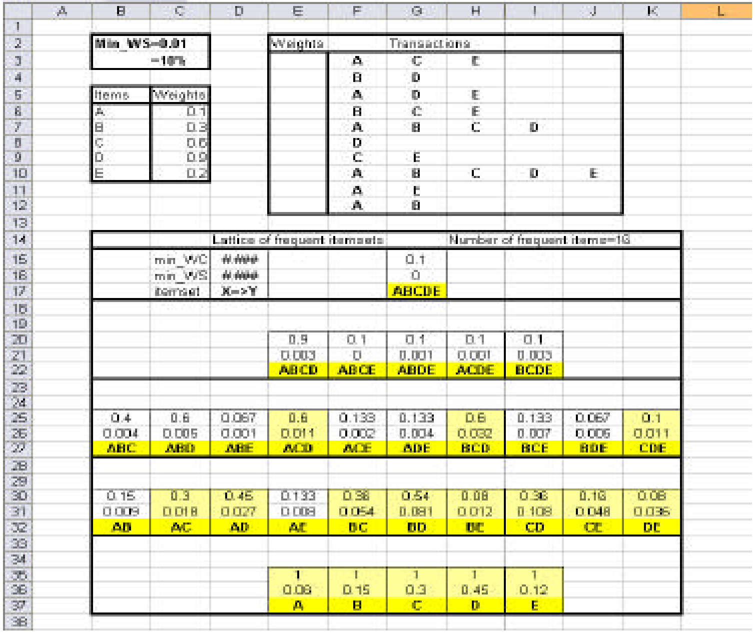
Large Itemsets	Support	Large?	Weighted Support	Large
AB	40%	Yes	0.16	No
AC	60%	Yes	0.42	Yes
ABC	40%	Yes	0.4	Yes
BC	40%	Yes	0.36	No
BD	60%	Yes	0.72	Yes
BCD	40%	Yes	0.72	Yes

Table 7 shows four large itemsets of size 2 (AB, AC, BC, BD) and two large itemsets of size 3 (ABC, BCD, which are a combination of two large itemsets. In classical ARM, when the weights are not considered, all of the six itemsets are large. But if we consider item weights and calculate the weighted support of itemsets according to definition 3 and 7, a new set of support values are obtained. In table 7, although the classical support of all itemsets is large, if ABC and BCD are frequent then their subsets will also be large according to classical ARM. But weighted support of AB and BC are no longer frequent.

In classical ARM using DCP, we assume that if AB and BC are not frequent, then ABC and BCD cannot be frequent so we don't consider generating the supersets that contain non-frequent itemsets.

5.1 Weighted Downward Closure Property (DCP)

We now argue that the DCP with boolean and quantitative data can be validated by using this new weighted framework. We give a proof and an example to illustrate this. Consider figure 1, where items in the transaction are assigned weights with supports above a user threshold. In figure 1, for each itemset, weighted support WS (the number above each itemset) is calculated by using definition 3 and weighted confidence WC (the number on top of each itemset) is calculated by using definition 4. If an itemset's weighted support is above the threshold, the itemset is frequent and we mark it with colour background and compared to the white background, meaning that it's not large.



	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		Min_WS=0.01			Weights		Transactions					
3			=10%			A	C	E				
4						B	D					
5		Items	Weights			A	D	E				
6		A	0.1			B	C	E				
7		B	0.3			A	B	C	D			
8		C	0.6			D						
9		D	0.9			C	E					
10		E	0.2			A	B	C	D	E		
11						A	E					
12						A	B					
13												
14				Lattice of frequent itemsets				Number of frequent items=16				
15			min_WC	#.###			0.1					
16			min_WS	#.###			0					
17			itemset	X=>Y			ABCDE					
18												
19												
20					0.9	0.1	0.1	0.1	0.1			
21					0.003	0	0.001	0.001	0.003			
22					ABCD	ABCE	ABDE	ACDE	BCDE			
23												
24												
25		0.4	0.6	0.067	0.6	0.133	0.133	0.6	0.133	0.067	0.1	
26		0.004	0.005	0.001	0.011	0.002	0.004	0.032	0.007	0.006	0.011	
27		ABC	ABD	ABE	ACD	ACE	ADE	BCD	BCE	BDE	CDE	
28												
29												
30		0.15	0.3	0.45	0.133	0.38	0.54	0.08	0.38	0.16	0.08	
31		0.009	0.018	0.027	0.008	0.054	0.081	0.012	0.108	0.048	0.036	
32		AB	AC	AD	AE	BC	BD	BE	CD	CE	DE	
33												
34												
35					1	1	1	1	1			
36					0.08	0.15	0.3	0.45	0.12			
37					A	B	C	D	E			
38												

Fig. 1. The lattice of frequent itemsets

It can be noted that if an itemset is with white background i.e. not frequent, then any of its supersets in the upper layer of the lattice can not be frequent. Thus "weighted downward closure property", is valid under the "weighted support" framework. It justifies the efficient mechanism of generating and pruning significance iteratively.

We also briefly prove that the DCP is always valid in the proposed framework. The following lemma applies to both boolean and fuzzy (quantitative) data and is stated as:

Lemma

If an itemset is not frequent them its superset cannot be frequent and $WS(subset) \geq WS(sueprset)$ is always true.

Proof

Given an itemset *X* not frequent i.e. $ws(X) < \min\_ \ ws$. 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 *tx* to denote a set of transactions each of which has all the items in X, i.e. $\{tx \mid tx \subseteq T, (\forall t \in tx, X \subset t)\}$. Similarly we have $\{ty \mid ty \subseteq T, (\forall t \in ty, Y \subset t)\}$. Since $X \subset Y$, we have $tx \subset ty$. Therefore $WS(tx) \geq WS(ty)$. According to the definition of weighted support, $\mathrm{WS}(X) = \frac{\sum_{i=1}^{n} \prod_{k=1}^{|X|} {}_{(\forall [i[w]] \in X)} t_i[i_k[w]]}{n}$ the denominator stays the same, therefore we have $WS(X) \geq WS(Y)$. Because $ws(X) < \min\_ \ ws$, we get $ws(Y) < \min\_ \ ws$. This then proves that Y is not frequent if its subset is not frequent.

Figure 2 illustrates a concrete example. Itemset AC appears in transaction 1, 5 and 8, therefore the WS (AC) = (.06+.06+.06)/10=0.018. Intuitively, the occurrence of its superset ACE is only possible when AC appears in that transaction. But itemset set ACE only appears in transactions 1 and 8, thus WS (ACE)=.012+.012/10=0.0024, where WS (ACE) <WS (AC). Summatively, if AC is not frequent, it's superset ACE is impossible to be frequent; hence there is no need to calculate its weighted support.

6 Experimental Results

For fuzzy weighted association rule mining standard ARM algorithms can be used or at least adopted after some modifications. An efficient algorithm is required because a significant amount of processing is undertaken to the application of fuzzy weighted association rule mining. The proposed Weighted ARM (WARM) and Fuzzy Weighted ARM (FWARM) algorithms belong to the *breadth first traversal* family of ARM algorithms, developed using tree data structures [13] and works in a fashion similar to the Apriori algorithm [10].

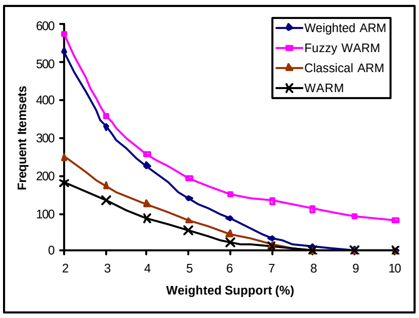
We performed several experiments using a T10I4N10KD100k (average of 10 items per transaction, average of 4 items per interesting set, 1000 attributes and 100,000 transactions/records) artificial data set. The data set was actually generated using the IBM Quest data generator. The data is a transactional database containing 100K records and 10K items. Two sets of experiments were undertaken with four different algorithms namely Weighted ARM, Fuzzy WARM, Classical Apriori ARM and Classical WARM shown in the results below:

1. In the first experiment we tested using both boolean and fuzzy datasets and compared the outcome with classical ARFM and WARM algorithms. The results show quite similar behaviour to classical ARM. Results are better than WARM because we consider the whole itemset space (pool) to generate frequent items unlike the pre- or post-processing WARM approaches. Experiments show (i) the number of frequent sets generated (using four algorithms), (ii) the number of rules generated (using weighted confidence) and (iii) execution time using all four algorithms.
2. Comparison of execution times using different weighted supports and data sizes.

6.1. Experiment One: (Quality Measures)

For experiment one, the T10I4D100K dataset described above was used with 50 weighted attributes. Each item is assigned a weight range between$[0..1]$. With fuzzy dataset each attribute is divided into five different fuzzy sets. Figure 3 shows the number of frequent itemsets generated using (i) weighted boolean dataset and (ii) with weighted quantitative attributes with fuzzy partitions (iii) classical ARM with boolean dataset and (iv) and WARM with weighted boolean datasets. A range of support thresholds was used.

As expected the number of frequent itemsets increases as the minimum support decreases in all cases. In figure 2, Weighted ARM shows the number of frequent itemsets generated using weighted boolean datasets. Fuzzy WARM shows the number of frequent itemsets using attributes with fuzzy linguistic values, Classical Apriori shows the number of frequent itemset using boolean dataset and classical WARM shows number of frequent itemsets generated using weighted boolean datasets with different weighted support thresholds. More frequent itemsets and rules are generated because of the large itemset pool.



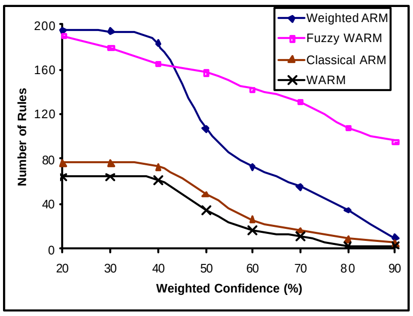


Fig. 2. No. of frequent Itemsets

Fig. 3. No. of Interesting Rules using Confidence

We do not use Apriori ARM to first find frequent itemsets and then re-prune them using weighted support measures. Instead all the potential itemsets are considered from beginning for pruning using Apriori approach in order to validating the DCP. In contrast classical WARM only considers frequent itemsets and prunes them (using pre or post processing). This generates less frequent itemsets and misses potential ones. Figures 3 shows the number of interesting rules generated using weighted confidence, fuzzy weighted confidence and classical confidence values respectively. In all cases, the number of interesting rules is less as compared to figure 2. This is because the interestingness measure generates fewer rules. Figure 4 shows the execution time of four algorithms.

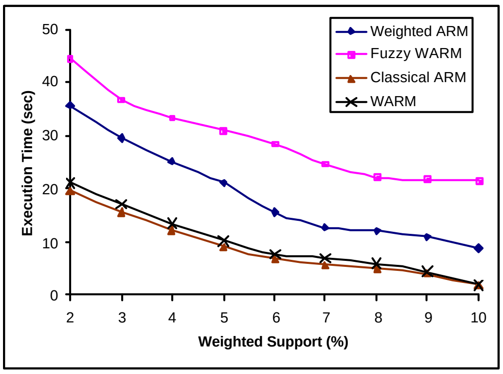
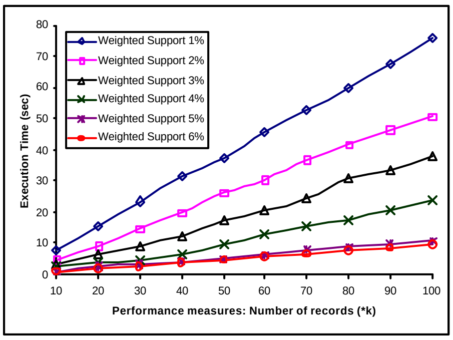


Fig. 4. Execution time to generate frequent itemsets

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

6.2. Experiment Two: (Performance Measures)

Experiment two investigated the effect on execution time caused by varying the weighted support and size of data (number of records).



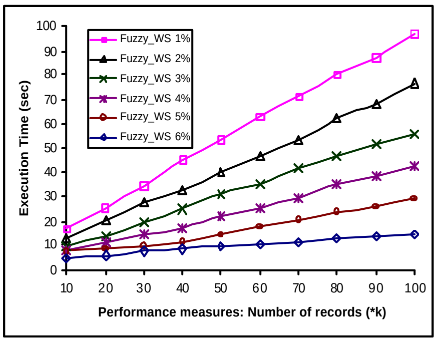


Fig. 5. Performance: weighted support (WS) **Fig. 6.** Performance: fuzzy WS

A support threshold from 0.1 to 0.6 and confidence 0.5 was used. Figures 5 and 6 show the effect of increasing the weighted support and number of records. To obtain different data sizes, we partitioned T10I4D100K into 10 equal partitions labeled 10K, 20K, ..,100K. Different weighted support thresholds were used with different datasets. Similarly from figure 6, the algorithm scales linearly with increasing fuzzy weighted support threshold and number of records, similar behaviour to Classical ARM.

7 Conclusion and future work

In this paper, we have presented a weighted support and confidence framework for mining weighted association rules (Boolean and quantitative data) by validating the downward closure property (DCP). We used classical and fuzzy ARM to solve the issue of invalidation of DCP in weighted ARM. We generalized the DCP and proposed a fuzzy weighted ARM framework. The problem of invalidation of downward closure property is solved using improved model of weighted support and confidence framework for classical and fuzzy association rule mining.

There are still some issues with different measures for validating DCP, normalization of values etc which are worth investigating.

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, pp. 661–666, Washington DC (2003).
2. Cai, C.H., Fu, A.W-C., Cheng, C. H, Kwong, W.W.: Mining Association Rules with Weighted Items. In: Proceedings of 1998 Intl. Database Engineering and Applications Symposium (IDEAS'98), pages 68--77, Cardiff, Wales, UK, July 1998
3. Wang, W., Yang, J., Yu, P. S.: Efficient Mining of Weighted Association Rules (WAR). In: Proceedings of the KDD, Boston, MA, August 2000, pp. 270-274
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, 4, pp. 2495--2499 (2003)
6. Gyenesei, A.: Mining Weighted Association Rules for Fuzzy Quantitative Items, Proceedings of PKDD Conference pp. 416--423 (2000).
7. Shu, Y. J., Tsang, E., Yeung, Daming, S.: Mining Fuzzy Association Rules with Weighted Items, 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, vol. 90, Melbourne, Florida, USA (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. Coenen, F., Leng, P., Goulbourne, G.: Tree Structures for Mining Association Rules. Data Mining and Knowledge Discovery, 8(1) 25--51 (2004)