A Weighted Utility Framework for Mining Association Rules

M. Sulaiman Khan[1, 2], Maybin Muyeba[1], Frans Coenen[2]
[1]*School of Computing, Liverpool Hope University, UK*
[2]*Department of Computer Science, University of Liverpool, UK*
khanm@hope.ac.uk, muyebam@hope.ac.uk, frans@liv.ac.uk

Abstract

Association rule mining (ARM) identifies frequent itemsets from databases and generates association rules by assuming that all items have the same significance and frequency of occurrence in a record i.e. their weight and utility is the same (weight=1 and utility=1) which is not always the case. However, items are actually different in many aspects in a number of real applications such as retail marketing, nutritional pattern mining etc. These differences between items may have a strong impact on decision making in many application unlike the use of standard ARM. Our framework, Weighted Utility ARM (WUARM), considers the varied significance and different frequency values of individual items as their weights and utilities. Thus, weighted utility mining focuses on identifying the itemsets with weighted utilities higher than the user specified weighted utility threshold. We conduct experiments on synthetic and real data sets using standard ARM, weighted ARM and Weighted Utility ARM (WUARM) and present analysis of the results.

1. Introduction

Data mining and knowledge discovery in databases is an interesting research area only developed in the last fifteen years. Association rule mining [1] is a popular data mining technique because of its wide application in marketing and retail communities as well as other more diverse fields. Researchers from the data mining community are more concerned with qualitative aspects of attributes (e.g. significance, utility) as compared to considering only quantitative ones (e.g. number of appearances in a database etc) because qualitative properties are required in order to fully exploit the attributes present in the dataset. Classical association rules mining techniques treat all items in the database equally by considering only the presence within a transaction without taking into account their significance to the user or business and also their utility as frequency of occurrences in each record. Although standard ARM algorithms are capable of identifying distinct patterns from a dataset, they sometimes fail to associate user objectives and business values with the outcomes of the ARM analysis. For example, in a retail mining application, frequent itemsets identified by the standard association rule mining algorithm may contribute only a small portion of the overall company profit because high profit and luxury items normally do not frequently appear in transactions and thus do not appear in rules with high support count values.

Given weighted items in table 1, we see from table 2 that the rule [jeans → suit, 50%] may be more important than [shirt → suit, 75%] 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 and jeans appear twice in transaction 2, which doubles the profit for jeans in that transaction. In contrast, standard ARM simply ignores this difference.

Table 1. Weighted items table

ID	Item	Profit	Weight	…
1	Shirt	£10	0.1	…
2	Jean	£25	0.3	…
3	Jacket	£50	0.6	…
4	Suit	£80	0.9	…

Table 2. Customers transactions

Tid	Shirt	Jean	Jacket	Suit
1	1	1	0	1
2	0	2	1	0
3	1	1	2	1
4	1	0	1	1

Many techniques and algorithms have been proposed for mining association rules that consider the qualitative properties of attributes in the databases. However, proposed techniques mostly compromise either on quality of rules or efficiency of algorithms.

The main challenge in mining weighted and utility association rules is that the anti-monotonic property [2] does not hold. Also the rules generated using these techniques are not guaranteed as high quality rules. These issues give rise to a new approach for identifying correct patterns from databases considering their significance and utilities as quality constraints. To our knowledge, there seems to be no work addressing both weighted and utility mining frameworks in a hybrid fashion.

Weighted Utility association rule mining (WUARM) is the extension of weighted association rule mining in the sense that it considers items weights as their significance in the dataset and also deals with the frequency of occurrences of items in transactions. Thus weighted utility association rule mining is concerned with both the frequency and significance of itemsets. Here weighted utility mining is helpful in identifying the most valuable and high selling items which contribute more to the company's profits. Weighted Utility of an item set depends upon two factors:

Transactional Utility: It is the frequency of occurrences or quantity of an item in a transaction.

Item significance: It is the value representing significance of an item (value, profit etc) and it holds across the dataset.

Items weights are stored in a weighted table (see table 1). Using transactional utilities and item weights, we can extract weighted utility rules.

The paper is organized as follows: section 2 gives a background and related work, section 3 gives a problem definition, section 4 discusses the downward closure property (DCP) and weighted utility property, section 5 shows experimental evaluation and a conclusion in section 6.

2. Background and related work

One major issue in association rule mining with weighted or utility settings is the invalidation of anti-monotonic property of itemsets. Previous works [3, 4, 5, 6] considered item weights as their utility to reflect their significance in the dataset. Our approach is different from all these in that we define utility differently by considering the frequency of occurrences of database attributes in a single record. The weight shows the significance of an item in a dataset e.g. profit margin of an item or items under promotional offers etc. We define item weight as a weighting function to signify an item differently in different domains (see next section, definition 2). This way we can extract those rules that have significant weight and high utility.

In [6] an object oriented mining approach is proposed that takes into account the items utilities' as the objective defined by the user to generate top-K high utility association rules, where K is the number of user specified rules. Standard DCP is not valid in the proposed model but instead a condition based weaker DCP approach is used. Also, the significance of items is not taken into account while generating the utility association rules.

A most recent framework for mining weighted ARs is presented in [2] where a generalised weighted ARM model is given that uses a modified Apriori approach [1] for binary and quantitative attributes. The approach also has a valid DCP. But this model only considers an items significance and not their utilities. In real world applications, transactional databases hold item utilities as well but classical and weighted ARM simply ignores these.

3. Problem definition

In this section, a formal description of the weighted utility mining problem is given and related concepts are described.

Definition 1 (Weighted Utility Mining) 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 $W = \{w_1, w_2, \cdots, w_{|I|}\}$ associated with each item in I.

Each i^{th} transaction t_i is some subset of I and a weight w is attached to each item i_j. Thus each item i_j will have associated with it a weight from the set W, i.e. a pair (i, w) is called a weighted item where $i \in I$ and $w \in W$. Weight for the "j^{th}" item in the "i^{th}" transaction is given by $t_i[w(i_j)]$ with u as the utility (frequency of occurrence) of an item in a transaction from a set U and represented with non negative integers. Weighted Utility mining is thus a triple $< \mathrm{I}, \mathrm{W}, \mathrm{U} >$.

Definition *2 Item Weight* IW is a non-negative real value $w(i_j)$ given to each item i_j ranging in [0..1] with some degree of importance, such that $w(i_j) = W(i_j)$, where W is a weighting function, a function relating specific values in a domain to user preferences. The weight reflects the significance of an item that is independent of transactions.

Definition 3 ***Weight Table*** is a two dimensional table $WT(I,W)$ over a collection of items I where W is the set of positive real numbers $w(i_j)$ given to each item $i \in I$.

Definition ***4 Item Utility*** of an item i_j in a transaction t_q is denoted as $t_q(i_j,u)$. Item utility reflects the frequency of an item in a transaction and is transaction dependent.

Definition ***5 Item Weighted Utility*** IWU is the integrated weight w and utility u value of an item i_j in a transaction t_i denoted by $t_i[(w(i_j),u)]$.

Definition ***6 Transaction Weighted Utility*** TWU is the aggregated weighted utilities of all the items present in a single transaction. Transaction weighed utility can be calculated as:

$$twu(t_i) = \frac{\sum_{j=1}^{|t_i|} t_i[(w(i_j),u)]}{|t_i|}$$

Definition ***6 Weighted Utility Support*** wus of an itemset X→Y is the fraction of transaction weighted utilities that contain both X and Y relative to the transactional weighted utility of all transactions. It can be formulated as:

$$wus(XY) = \frac{\sum_{i=1}^{|S|} twu(t_i)}{\sum_{i=1}^{|T|} twu(t_i)}$$

where

$$S = \{S \mid S \subseteq T,\ X \cup Y \in S\}.$$

By this means, weighted utility support is modeled to measure the actual contribution of an itemset in the dataset in weighted utility association rule mining scenario.

4. 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) or anti-monotonic property of itemsets. For example, in standard 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 the algorithm to generate large itemsets of increasing size by adding items to itemsets that are already large. In the weighted utility framework where each item is given a weight with several occurrences, 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 5.

In table 5, all frequent itemsets are generated using 30% support threshold. In column two (i.e. Standard ARM), itemset {ACD} and {BDE} are frequent with support 30% and all of their subsets {AC}, {AD}, {CD} and {BD}, {BE}, {DE} respectively are frequent as well. But in column 3 with weighted settings, itemsets {AC} and {BE} are no longer frequent and thus violate the DCP.

4.1. Weighted Utility anti-monotonic property

We argue that the DCP with weighted utility framework can be validated. We prove this by showing that if an itemset is not frequent, then its superset cannot be frequent and is always true (see table 1, column 4, Weighted Utility ARM, only the itemsets are frequent with frequent subsets).

We also briefly prove that the monotonic property of itemsets is always valid in the proposed framework and is stated using the lemma as follows:

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

Proof: Given an itemset X not frequent i.e. $wus(X) < \min\_wus$. For any itemset $Y, where\ X \subset Y$ 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 $wus(tx) \geq wus(ty)$. According to the definition of weighted utility support, the denominator stays the same, therefore we have $wus(X) \geq wus(Y)$. Because $wus(X) < \min\_wus$, we get $wus(Y) < \min\_wus$, this then proves that Y is not frequent if its subset is not frequent.

4.2. Simulated Example

We demonstrate an example to simulate the process of weighted utility mining framework with valid DCP. Table 3 is the weighted items table with weights associated with each item according to some profit margin.

Table 3. Weighted items table

Items *i*	Profit	Weights *w*
A	£60	0.6
B	£10	0.1
C	£30	0.3
D	£90	0.9
E	£20	0.2

Table 4 is a transaction database with 10 records. The last column in table 4 shows the transaction weighted utilities for each transaction and the last row shows the total transactional utilities sum.

Table 4. Transaction database with transactional weighted utilities of items

Items	A	B	C	D	E	*twu*
1	1	1	4	1	0	0.700
2	0	1	0	3	0	1.400
3	2	0	0	1	0	1.050
4	0	0	1	0	0	0.300
5	1	2	0	1	3	0.575
6	1	1	1	1	1	0.420
7	0	2	3	0	1	0.433
8	0	0	0	1	2	0.650
9	7	0	1	1	0	1.800
10	0	1	1	1	1	0.375
					Weighted Utility count	**7.703**

Table 5 shows all possible itemsets generated using table 3. Itemsets with classical ARM support are shown in column 2, itemsets with weighted ARM support are shown in column 3 and column 4 shows itemsets with weighted utility ARM support. Column 1 in table 5 shows the itemsets ids. Support threshold for classical ARM and weighted ARM is set to 30% and for weighted utility ARM it is set to 0.3 (as equivalent to 30%). Itemsets with highlighted background are frequent itemsets.

This simulation illustrates the effect of an item's utility and its weight on the generated rules. Using a standard ARM technique without considering items' utilities and their weights, rules generated with 30% support are shown in column 2. It is interesting to note that the rules generated with 30% support using weighted ARM framework (column 3) are all also frequent using the classical ARM technique. This is due to the fact that WARM uses already generated frequent itemsets with standard ARM approach and thus misses many potential ones as shown in table 5. But the proposed framework overcomes this problem by using weights and utilities for itemsets pruning using the Apriori approach, thus considers potential itemsets which WARM ignores.

Table 5. Weighted utility mining comparison

#	Standard ARM	Weighted ARM	Weighted Utility ARM
1.	A (50%)	A (30%)	A (0.59)
2.	A→B (30%)	A→B (21%)	A→B (0.22)
3.	A→B→C (20%)	A→B→C (20%)	A→B→C (0.14)
4.	A→B→C→D (20%)	A→B→C→D (38%)	A→B→C→D (0.14)
5.	A→B→C→D→E(10%)	A→B→C→D→E(21%)	A→B→C→D→E (0.05)
6.	A→B→C→E (10%)	A→B→C→E (12%)	A→B→C→E (0.05)
7.	A→B→D (30%)	A→B→D (48%)	A→B→D (0.22)
8.	A→B→D→E (20%)	A→B→D→E (36%)	A→B→D→E (0.13)
9.	A→B→E (20%)	A→B→E (18%)	A→B→E (0.13)
10.	A→C (30%)	A→C (27%)	A→C (0.38)
11.	A→C→D (30%)	A→C→D (54%)	A→C→D (0.38)
12.	A→C→D→E (10%)	A→C→D→E (20%)	A→C→D→E (0.05)
13.	A→C→E (10%)	A→C→E (11%)	A→C→E (0.05)
14.	A→D (50%)	A→D (75%)	A→D (0.590)
15.	A→D→E (20%)	A→D→E (34%)	A→D→E (0.13)
16.	A→E (20%)	A→E (16%)	A→E (0.13)
17.	B (60%)	B (6%)	B (0.51)
18.	B→C (40%)	B→C (16%)	B→C (0.25)
19.	B→C→D (30%)	B→C→D (39%)	B→C→D (0.19)
20.	B→C→D→E (20%)	B→C→D→E (30%)	B→C→D→E (0.10)
21.	B→C→E (30%)	B→C→E (18%)	B→C→E (0.16)
22.	B→D (50%)	B→D (50%)	B→D (0.45)
23.	B→D→E (30%)	B→D→E (36%)	B→D→E (0.18)
24.	B→E (40%)	B→E (12%)	B→E (0.23)
25.	C (60%)	C (18%)	C (0.52)
26.	C→D (40%)	C→D (48%)	C→D (0.43)
27.	C→D→E (20%)	C→D→E (28%)	C→D→E (0.10)
28.	C→E (30%)	C→E (15%)	C→E (0.16)
29.	D (80%)	D (72%)	D (0.90)
30.	D→E (40%)	D→E (44%)	D→E (0.26)
31.	E (50%)	E (10%)	E (0.32)

Rules {A→C}, {AC→D} and {A→D} in column 4 are frequent under Weighted Utility framework because of their high weight and utility in transactions. But it is interesting to get a rule B→D, because B has least weight and low utility count. Justification for this kind of rule is that, though B has low weight (0.1), it has the second highest count support (i.e. 60%) and it appears more with item D than any other item (i.e. 50%). Another aspect to note is that D has the highest weight (0.9) and count support (80%). These kinds of rules can help in "Cross-Marketing" and "Loss Leader Analysis" in real world applications.

Further, the rules generated using our approach holds a valid DCP and the monotonic property of itemsets as proved in section 4.1 and table 5 illustrates a concrete example of this. Itemset BD appears in transaction 1, 2, 5, 6, 9 and 10 with high utility, therefore the $wus(BD) = 0.45$. Intuitively, the occurrence of its superset BDE is only possible when

BD appears in that transaction. Itemset BDE only appears in transactions 5, 6 and 10, thus $wus(BDE) = 0.18$, were $wus(BDE) < wus(BD)$. Summatively, if BD is not frequent, it's superset BDE is impossible to be frequent; hence there is no need to calculate its weighted utility support.

5. Experimental Evaluation

In this section we report our performance study for the WUARM approach. In particular, we compare the quality and efficiency of WUARM algorithm with Apriori version of standard and weighted ARM, a well known algorithm for mining frequent itemsets.

Experiments were undertaken using three different association rule mining techniques. Three algorithms were used for each approach, namely Standard ARM as classical Apriori ARM, Weighted ARM (WARM) as post processing Apriori weighted ARM and Weighted Utility ARM (WUARM) as 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 generated using three algorithms described above with real and synthetic data. In the second experiment, we showed the scalability of the proposed WUARM algorithm by comparing the execution time of three algorithms with varying support thresholds.

Both real and synthetic datasets are used in experiments. For real data we used Retail dataset, a real market basket data [7] and T10I4D100K synthetic data is obtained from the IBM dataset generator [8].

5.1. Frequent Itemsets Comparison

For quality measure, both the dataset described above were used. Each item is assigned a weight range between [0-1] according to their significance in the dataset.

We generated artificial frequencies of items range [1-10] for both real and synthetic data to obtain items utilities in transactions. In figure 1 and 2, the x-axis shows support thresholds from 1% to 6% and on the y-axis the numbers of frequent itemsets are shown. Three algorithms as described above are compared. Weighted Utility ARM algorithm uses weighted datasets with items utilities; Standard ARM using binary dataset and WARM using weighted 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 and utilities of each item in Standard ARM and WARM is 1 i.e. all items with utility exactly one, which is not the case in real applications.

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

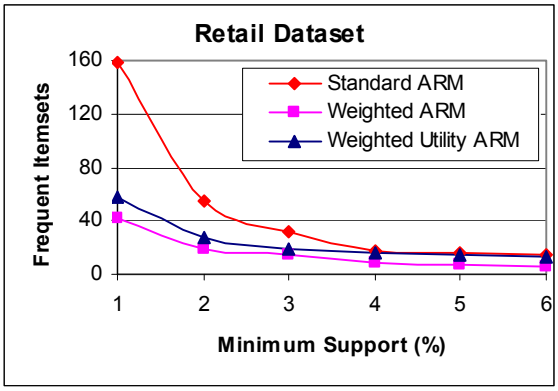

Figure 1. No. of frequent Itemsets generated using varying support threshold (real data)

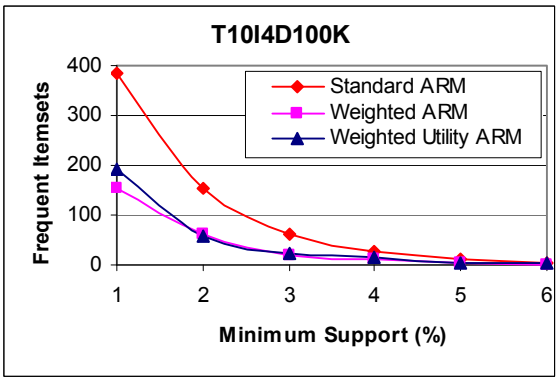

Figure 2. No. of frequent Itemsets generated using varying support threshold (synthetic data)

WUARM generated fewer rules than standard ARM but more rules than weighted ARM because it not only considers the items weight but also take into account the items utilities in each transaction and considers potential itemsets which weighted ARM ignores. Also we do not use standard ARM approach to first find frequent itemsets and then re-prune them using weighted utility support measures. Instead all the potential itemsets are considered from beginning for pruning using Apriori approach to validate the DCP.

Results of the proposed WUARM approach are better than weighted ARM because we consider all the possible itemsets and uses items weight and their utilities. Moreover, WUARM, Standard ARM and WARM utilises binary data.

5.2. Performance

For performance study, we compare the execution time of WUARM algorithm with classical Apriori ARM and WARM algorithms using both real and synthetic data.

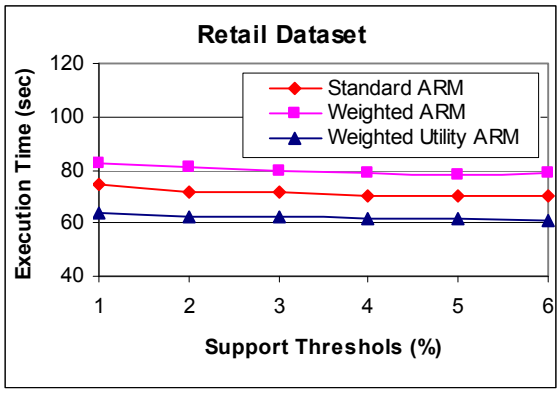

Figure 3. Execution time (real data)

We investigated the effect on execution time caused by varying the support threshold with fixed data size (number of records). In figure 3 and 4, a support threshold from 1% to 6% is used again.

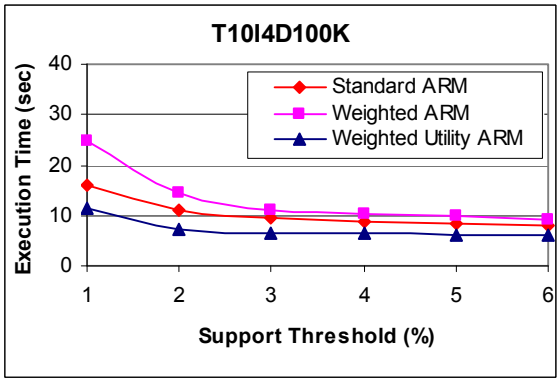

Figure 4. Execution time (synthetic data)

With both real and synthetic data WUARM has comparatively low execution time due to the fact that it generates fewer rules than standard ARM and do not use pre or post processing as mentioned earlier.

Weighted ARM has slightly higher execution time due to the fact that WARM initially uses classical ARM approach and then use already generated frequent sets for pruning, which takes computation time.

6. Conclusion

In this paper, we have presented classical and weighted Association Rule Mining, in particular, the weighted utility framework which has the ability to deal with item weights and utilities in a hybrid fashion. This framework can be integrated in the mining process, which is different to most utility and weighted ARM algorithms. To solve this problem, we identified the challenge faced while using weights and utilities together, in particular the invalidation of downward closure property.

Using a simulated example, we proved that weight and utility can be used together to steer the mining focus to those itemsets with significant weight and high utility. This is further proven by experiments conducted on real and synthetic datasets. We have showed that efficient WUARM algorithms can be developed by modifying the standard Apriori algorithm with weighted utility settings. The experiments also show that the algorithm is scalable.

7. References

[1] Bodon, F.: "A Fast Apriori implementation", *In ICDM Workshop on Frequent Itemset Mining Implementations*, vol. 90, Melbourne, Florida, USA (2003)

[2] M. Sulaiman Khan, M. Muyeba, F. Coenen, "Fuzzy Weighted Association Rule Mining with Weighted Support and Confidence Framework", *to appear in ALSIP (PAKDD)* 2008, Osaka, Japan.

[3] Hong Yao, Howard J. Hamilton, "Mining itemset utilities from transaction databases", *Data & Knowledge Engineering*, pp. 603- 626, Volume 59, Issue 3 (2006).

[4] H. Yao, H. J. Hamilton, and C. J. Butz, "A Foundational Approach to Mining Itemset Utilities from Databases", *4th Intl. conf. on Data Mining*, Florida, USA, (2004)

[5] Jianying Hu, Aleksandra Mojsilovic, "High-utility pattern mining: A method for discovery of high-utility item sets", *Pattern Recognition*, Elsevier Science Inc, Volume 40 , Issue 11, pp. 3317-3324, (2007).

[6] R. Chan, Q. Yang and Y-D. Shen, "Mining High Utility Itemsets", *In proc. of 3rd IEEE International Conference on Data Mining (ICDM'03)*, pp. 19-26, Melbourne, FL, (2003)

[7] FIMI, Frequent Itemset Mining Implementation Repository, http://fimi.cs.helsinki.fi/.

[8] IBM Synthetic Data Generator, http://www.almaden.ibm.com/software/quest/resources/index.html