Efficient Mining of Weighted Association Rules (WAR)

Wei Wang
IBM Watson Research Center
ww1@us.ibm.com

Jiong Yang
IBM Watson Research Center
jiyang@us.ibm.com

Philip S. Yu
IBM Watson Research Center
psyu@us.ibm.com

ABSTRACT
In this paper, we extend the tradition association rule problem by allowing a weight to be associated with each item in a transaction, to reflect interest/intensity of the item within the transaction. This provides us in turn with an opportunity to associate a weight parameter with each item in the resulting association rule. We call it weighted association rule (WAR). WAR not only improves the confidence of the rules, but also provides a mechanism to do more effective target marketing by identifying or segmenting customers based on their potential degree of loyalty or volume of purchases. Our approach mines WARs by first ignoring the weight and finding the frequent itemsets (via a traditional frequent itemset discovery algorithm), and is followed by introducing the weight during the rule generation. It is shown by experimental results that our approach not only results in shorter average execution times, but also produces higher quality results than the generalization of previous known methods on quantitative association rules.

Categories and Subject Descriptors
H.2.8 [**Information Systems**]: Database Management—*database applications*

General Terms
Weighted association rules, Ordered shrinkage

1. INTRODUCTION
Association rule discovery has been an active research topic during recent years. However, the traditional association rules focus on binary attribute. This model only considers whether an item is present in a transaction, but does not take into account the weight/intensity of an item within a transaction. For example, a customer may purchase 10 bottles of soda and 5 bags of snacks and another may purchase 4 bottles of soda and 1 bag of snacks at a time. These two transactions will be treated the same in the conventional association rule approach. This could lead to the loss of some vital information. Assume, for example, that if a customer buys more than 7 bottles of soda, he is likely to purchase 3 or more bags of snacks. Otherwise, the purchase tendency of soda is not strong. The traditional association rule cannot express this type of relationship. With this knowledge, the supermarket manager may set a promotion such as "if a customer buys 8 bottles of soda, he can get two free bags of snacks."

In this paper, we first extend the traditional association rule problem by allowing a weight to be associated with each item in a transaction, to reflect interest/intensity of each item within the transaction. In turn, this provides us with an opportunity to associate a weight parameter with each item in a resulting association rule. We call them weighted association rules (WAR). For example, $soda[4,6] \longrightarrow snack[3,5]$ is a weighted association rule indicating 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. Thus WAR can not only improve the confidence in the rules, but also provide a mechanism to do more effective target marketing by identifying or segmenting customers based on their potential degree of loyalty or volume of purchases.

Previous work dealing with numerical attributes includes the quantitative association rule approach and optimized association rule approach. These approaches are not designed for weighted association rules [4]. In the problem we study in this paper, there can be a very large number of items and every item has a numerical attribute, although only a small fraction of items are present in a transaction. Thus, in our approach, the frequent itemsets are first discovered (without considering weights), and then the weighted association rules for each frequent itemset are generated. Our goal is to segment the weight domain of each item in the itemset so that rules with higher confidence can be discovered. Moreover, the specified weight interval of each attribute in a weighted association rule should also coincide with the natural distribution of the data and human intuition. In most case, only the weight interval combinations that represent a significant number of transactions are interesting. Our goal can be transformed into finding highly populated regions. Thus, we use another metric *density* for such purpose. As a result, the weight domain space of each frequent itemset is partitioned into fine grids. A density threshold is used to separate the transaction concentrating regions from the rest. WARs can be identified based on these "dense" regions. The contributions of this paper are summarized as follows.

- A new class of association rule problems — WAR is proposed.
- Due to the nature of this problem, the mining process is accomplished by a twofold approach: first generating frequent itemsets and then deriving WARs from each frequent itemset.
- During the WAR derivation process,
  - The concept of density is employed to separate transaction concentrated regions from the rest.
  - An efficient *ordered shrinkage* algorithm is proposed to derive WARs from a high density region through shrinkages to meet the confidence requirement.

The remainder of this paper is organized as follows. The problem is formulated in Section 2, while Section 3 outlines the general approach. Sections 4 and 5 present the space partition and WAR generation, respectively. Section 6 shows the experimental results. A conclusion is drawn in Section 7.

2. PROBLEM FORMULATION

Let $\Im = \{i_1, i_2, \ldots, i_M\}$ be a set of items and $\wp$ be the set of non-negative integers. A pair $\langle x, w\rangle$ is called a **weighted item**, where $x \in \Im$ is an item and $w \in \wp$ is the weight associated with x. A transaction is a set of weighted items. For example, $T_1 = \{\langle fashion, 15\rangle, \langle sports, 10\rangle\}$ and $T_2 = \{\langle fashion, 20\rangle, \langle book, 5\rangle\}$ are two transactions. An interval weighted item is a triple $\langle x, l, u\rangle$. This denotes the fact that the weight associated with the item x is within the range $[l, u]$ where l and u are non-negative integers and $l \leq u$. Note that we can always view a weighted item $\langle x, w\rangle$ as a special case of the interval weighted item $\langle x, l, u\rangle$ where $w = l = u$. Therefore, we will use the term *weighted item* if no ambiguity will occur. Given two weighted items $I_1 = \langle x_1, l_1, u_1\rangle$ and $I_2 = \langle x_2, l_2, u_2\rangle$, we call I_1 a **generalization** of I_2 (and I_2 is a **specialization** of I_1) iff $x_1 = x_2$ and $l_1 \leq l_2 \leq u_2 \leq u_1$. For example, $\langle fashion, 10, 20\rangle$ is a specialization of $\langle fashion, 10, 25\rangle$. Note that any weighted item $\langle x, l, u\rangle$ can be view as a specialization of the item x.

We use the term **weighted itemset** to represent a set of weighted items. Let $item(X)$ denote the set of items that are involved in a weighted itemset X, i.e., $item(X) = \{x \mid x \in \Im, \langle x, l, u\rangle \in X\}$. Given two weighted itemsets X_1 and X_2, X_1 is a *specialization* of X_2 (or X_2 is a *generalization* of X_1) iff $item(X_1) = item(X_2)$ and each weighted item in X_1 is a specialization of a weighted item in X_2. For instance, $\{\langle fashion, 10, 20\rangle, \langle book, 5, 7\rangle\}$ is a specialization of $\{\langle fashion, 10, 20\rangle, \langle book, 5, 10\rangle\}$. Given a transaction T and a weighted item $\langle x, l, u\rangle$, we say that T **supports** this weighted item iff there exists a weighted item $\langle x, w\rangle \in T$ such that $\langle x, w\rangle$ is a specialization of $\langle x, l, u\rangle$. Similarly, we say that a transaction T **supports** a weighted itemset X iff T supports each individual weighted item in X. For instance, if $X = \{\langle fashion, 10, 20\rangle, \langle book, 5, 10\rangle\}$, then T_2 supports X while T_1 does not. Given a weighted itemset X and a set of transactions, referred to as $\Re$, we say X has **support** s in $\Re$ iff $s\%$ of transactions in $\Re$ support X. Note that the support of a weighted itemset is always less than or equal to the support of any of its generalization.

A **weighted association rule (WAR)** is an implication $X \longrightarrow Y$ where X and Y are two weighted itemsets and $item(X) \cap item(Y) = \emptyset$. A transaction is said to *support* a WAR $X \longrightarrow Y$ iff this transaction supports the weighted itemset $X \cup Y$. In turn, we define the *support* of the WAR as the support of $X \cup Y$. In addition, we say that the WAR $X \longrightarrow Y$ holds in the transaction set $\Re$ with *confidence c* iff $c\%$ of the transactions in $\Re$ that support X also support Y. In other words, the confidence of the WAR is the ratio of the support of $X \cup Y$ over the support of X. The *density* of a WAR is defined as the ratio of the actual support of the WAR and the "expected" support of the WAR. We will elaborate on the density definition in Section 5. In this paper, we assume that Y only contains one weighted item for the sake of simplicity.

Given a transaction set $\Re$, our objective is to find a set of **Weighted Association Rules** (WAR) which have support, confidence, and density greater than or equal to some user-specified minimum support (referred to as *minsup*), minimum confidence (referred to as *minconf*), and minimum density (referred to as d). Since there could be a huge number of qualified WARs, in this paper, we aim at mining **maximum WAR**. A qualified WAR $X \longrightarrow Y$ is a maximum WAR if for any generalization X' of X and Y' of Y where $X' \neq X$ and $Y' \neq Y$, neither of $X' \longrightarrow Y$, $X \longrightarrow Y'$, nor $X' \longrightarrow Y'$ is a qualified WAR.

3. GENERAL APPROACH

Clearly, there can be a huge number of potential WARs, due to the numerical nature of the weight. Efficient pruning of such a huge search space becomes a crucial task in the mining process. Unlike [3], we design a twofold approach based on an observation we made in Section 2: *the support of a weighted itemset is always less than or equal to the support of any of its generalizations.* This indicates that, for any weighted itemset I, its support is always less than or equal to the support of $item(I)$. This suggests that we can first calculate frequent itemsets (without considering the weights) and then examine the weight factor for each frequent itemset to generate the WARs. A frequent (weighted) itemset is a (weighted) itemset whose support is larger than or equal to the threshold *minsup*. Thus, we employ a twofold approach. (1) Generate frequent itemsets. In this step, we ignore the weight associated with each item in the transaction set. (2) For each frequent itemset, find the WAR(s) that meets the support, confidence, and density thresholds.

Since the first phase is mainly the frequent itemset counting (as in the traditional association rule mining), we will not elaborate it in this paper. After obtaining the set of frequent itemsets, referred to as F, we examine them to generate the weighted association rules. Given an itemset I of cardinality n, the domain of the weights of all items forms an n dimensional space where each dimension corresponds to the weight of one item. (For simplicity, we assume that the weight range on each dimension to be the same.) Each specialization of I corresponds to an n-dimensional (rectangular) box within this space. Our objective is to find the maximum box(es) so that support, confidence, and *density* are satisfied. To facilitate this process, we discretize the space into a set of grids and divide the second phase further.

1. Space partition and counter generation: The goal is to identify, for each frequent itemset, those (dense) grids that satisfy the density requirement. An efficient pruning technique is provided to reduce the number of grids that need to be evaluated and maintained.
2. WARs generation: The goal is to generate the maximum WARs from the dense boxes enclosing adjacent dense grids. As the dense boxes generally do not satisfy the confidence requirement, an ordered shrinkage approach is developed to shrink the dense boxes in an orderly way to meet the confidence requirement.

4. SPACE PARTITION

The density concept is introduced to develop effective pruning techniques for identifying candidate boxes for WAR mining. We want to keep the additional parameters that need to be specified by the users to a minimum. The intent is to introduce one density threshold that is applicable to all grids, regardless of the number of dimensions. A straightforward partitioning method is to divide each dimension into a fixed number of partitions. However, under this approach, as the number of dimension increases, the number of grids grows exponentially, while the average density of each grid drops rapidly. This implies that different density thresholds would be needed for grids of different dimensionalities. We hence use an alternative approach that keeps the number of grids N fixed regardless of the number of dimensions. Another advantage of this partitioning method is that we can easily control the number of counters by varying the value of N. This provides us with the capability to take full advantage of the available storage space dynamically.

The grid is taken as the granularity of our WAR mining process. Any n-dimensional *box* within this space, which is the union of a set of adjacent grids and is rectangular in shape, uniquely corresponds to a specialization of I. Thus, in the remainder of this paper, we will use the term "box" and "weighted itemset" interchangeably if no ambiguity will occur. Let α be the average number of transactions that support a grid. We have $\alpha = \frac{|\Re|}{N}$. The *density* of a grid is defined as the ratio of the support of this grid and α. Intuitively, density can be viewed as an indication of the relative concentration of transactions within the space. A grid is *dense* iff its density is above a threshold d, where $d > 1$ is a small real number. In order to limit the search space, we only investigate dense grids and the region formed by them. Unlike a box, a region does not have to be of rectangular shape. The motivation behind this is that we only want to report WARs which represent significant patterns in the data. Therefore, a range will not be included in a WAR if there is not enough evidence (density) to support it. A *dense region* is the union of a set of adjacent dense grids. Each box within a dense region, referred to as *dense box*, is a candidate of frequent weighted itemset. The *volume* of a box is defined as the number of distinct grids it contains.

Counter Pruning

Although this approach needs one density threshold for pruning, it does create additional complexities on the pruning algorithms, as the projection of an n-dimensional grid onto an $(n-1)$-dimensional subspace does not align on the grid boundaries of the $(n-1)$-dimensional subspace. In fact, such projection is always of a looser granularity than the grid in the $(n-1)$-dimensional space. This can be easily observed in Figure 1. The projection of a grid in Figure 1(1) overlaps with 9 grids in Figure 1(2).The levelwise pruning in [2] can still be adopted with one modification: given an n-dimensional grid g, if the support of the union of the $(n-1)$-dimensional grids that can cover the projection of g is below $d \times \alpha$, than g cannot be a dense grid. For example, if the support of the shaded area in Figure 1(2) is below $d \times \alpha$, then the grid in Figure 1(1) cannot be dense. After finding all dense grids, dense regions, which serve as the basis for WAR generation, can be easily identified by a depth-first traversal through neighboring dense grids similar to [2].

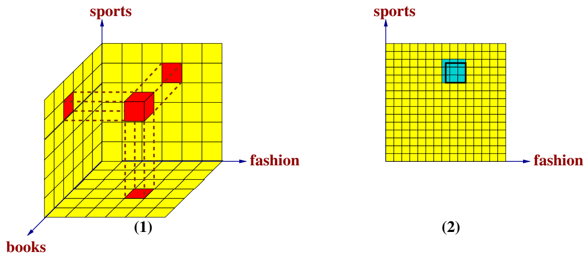

Figure 1: Partitioning and Counter Pruning

5. WAR GENERATION

Given an n-itemset $I = \{i_1, i_2, \ldots, i_n\}$, there are potentially n different WAR formats because each item might serve as the right hand side. In order to search for the range associated with each item, we examine each dense region sequentially. Given a dense region, if its support is below the required *minsup* threshold, we simply discard it, since no qualified WAR can be generated from this dense region. Otherwise, we start from the minimum bounding box of the dense region. The minimum bounding box is the box with smallest volume which contains the dense region. In general, such a box itself may not satisfy the confidence requirement, but may contain dense box(es) corresponding to qualified WAR(s). Thus, we choose to shrink the minimum bounding box towards the maximum WARs. An alternative algorithm is to pick a grid and grow towards the maximum WARs. Since the maximum WARs usually have large volumes, shrinking towards a maximum WAR is generally more efficient than growing towards it. To the best of our knowledge, this is the first algorithm to use shrinking instead of growing in mining association rules or clustering. A *shrinkage* is defined as the action that reduces the span of a box over one dimension by exactly one base interval. At each step, a candidate box is chosen to perform a shrinkage to generate a new dense box that may serve as a candidate for further shrinkage at a later step. The process ends when the newly generated dense box meets the confidence requirement or all remaining candidate boxes fall to meet the support requirement. Without loss of generality, we assume that there is only one dense region. Figure 2(a) is a dense region whose minimum bounding box is shown in Figure 2(b), where the dark shaded grids and light shaded grids are dense grids and non-dense grids, respectively. For each item, the range of its weight can be shrunken towards two directions: increasing the lowerbound or decreasing the upperbound. Thus, there are $2n$ different shrinkages applicable to a weighted n-itemset. As a result, $2n$ new weighted n-itemsets can be generated. Since these new weighted item sets are produced by one shrinkage from the original weighted itemset, we call them the *immediate specializations* of the original weighted itemset. The original weighted itemset is an *immediate generalization* of these new weighted itemsets. Figure 2(c) shows the six different directions to shrink a weighted 3-itemset and their corresponding outcomes.

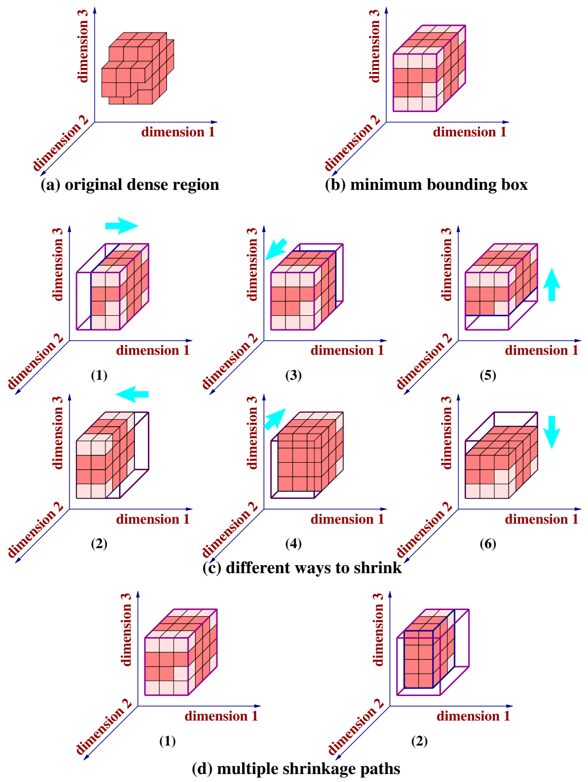

Figure 2: Region shrinkage

Clearly, each dense box can be reached by a set of shrinkages from the minimum bounding box of the dense region. Assuming that we can only perform one shrinkage on one candidate box at a time, the entire process can be viewed as a sequence of operations (B_j, H_j) $(j = 1, 2, \dots)$ where B_j is a candidate box and H_j is a shrinkage along some direction. Let Ψ_j be the set that includes the minimum bounding box of the dense region B_1 and all boxes generated via operations $(B_1, H_1), \dots, (B_{j-1}, H_{j-1})$. Ψ_j thus represents the set of candidate boxes (for further shrinking) available before the jth operation. Clearly, we have $B_j \in \Psi_j$ at each step. The sequence stops when all WARs are generated. For the box B_j visited at the jth step, there exists a subsequence of operations $(B_{j_1}, H_{j_1}), (B_{j_2}, H_{j_2}), \dots, (B_{j_r}, H_{j_r})$ such that B_{j_1} is the minimum bounding box of the dense region and B_{j_k} is the box produced by performing shrinkage $H_{j_{(k-1)}}$ on $B_{j_{(k-1)}}$ where $1 < k \leq r$ and $1 \leq j_1 < j_2 < \cdots < j_r < j$. $H_{j_1}, H_{j_2}, \dots, H_{j_r}$ is a *shrinkage path* to B_j from the minimum bounding box of the dense region. Note that, at each step, there are multiple candidate boxes in Ψ_j and different shrinking directions to choose from. As a result, different algorithms for picking candidate box and shrinking direction can produce different sequences of operations and hence could require different number of operations before all necessary WARs are generated. Therefore, the efficiency of an algorithm depends on the amount of time consumed at each operation and the number of operations needed.

One possible approach is the brute force algorithm. Let Ψ'_j be Ψ_j excluding those boxes which satisfy either of the following conditions: (1) all of its immediate specializations are in Ψ_j, or (2) it is a maximum WAR. At each step, the brute force algorithm randomly picks a box from Ψ'_j. The algorithm terminates when $\Psi'_j = \emptyset$. The computational complexity of the brute force algorithm is $O\left((2n)^{n \times \sqrt[n]{N}}\right)$. Therefore, this approach is indeed inefficient. Intuitively, this inefficiency is caused by the fact that a box may be visited multiple times via different shrinkage paths. This would lengthen the operation sequence and thus cause the inefficiency. For example, there are two paths to reach the box in Figure 2(d)(2) from the box in Figure 2(d)(1). One is "reducing the upper bound of dimension 2" followed by "reducing the upper bound of dimension 1", while the other is "reducing the upper bound of dimension 1" followed by "reducing the upper bound of dimension 2". The set of shrinkages of the two paths is the same. The only difference is that different paths adopt a different permutation of these shrinkages. In general, if it takes b shrinkages to reach from box B' from B (where B' is enclosed within B), then there exist $b!$ different shrinkage paths. In order to eliminate the redundancy of weighted itemset generation, we introduce an **ordered shrinkage** technique. This guarantees that each weighted itemset is generated exactly once.

Ordered Shrinkage

We first choose a permutation, referred to as Ω, of $2n$ different shrinking directions and retain this order during the entire process. For example, the six shrinking directions in Figure 2(c) can be ordered as *increasing lowerbound of dimension 1, decreasing upperbound of dimension 1, increasing lowerbound of dimension 2, decreasing upperbound of dimension 2, increasing lowerbound of dimension 3, decreasing upperbound of dimension 3*. Then, during the shrinking process, a shrinkage of the kth direction in Ω can be performed on a box B only if no shrinkage of direction after the kth one in Ω has been performed to generate B. We call such shrinkage a *valid* shrinkage. For instance, if we take the box in Figure 2(c)(4) as the candidate to generate new weighted itemsets, according to the order we pick, three shrinking directions can be applied: *decreasing upperbound of dimension 2, increasing lowerbound of dimension 3*, and *decreasing upperbound of dimension 3*. As a result, the only valid shrinkage path from Figure 2(d)(1) to Figure 2(d)(2) is "reducing the upper bound of dimension 1" followed by "reducing the upper bound of dimension 2". It is obvious that this ordered shrinkage approach completely eliminates the redundancy of box generation in the previous brute force approach, by providing each possible weighted itemset a unique shrinkage path from the root. The computational complexity of the ordered shrinkage for one itemset is $O(N^2)$. The detailed algorithm is shown in [4].

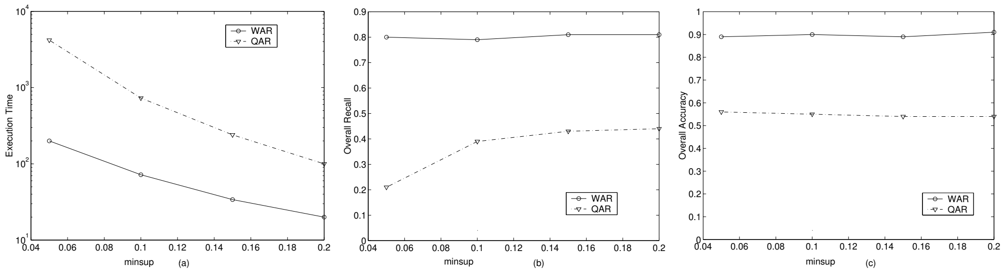

Figure 3: WAR vs. QAR

6. EXPERIMENTAL RESULTS

We implemented our algorithm in C and executed it on an IBM AIX RS6000 machine with a 333 MHz CPU. The data was placed on a disk with approximate 8 MB/s throughput. To analyze the performance of our proposed approaches, we generate the input data in a similar manner as in [1]. The data set contains 1 million transactions with 10000 different items. The number of items in a transaction is clustered around a mean 15 and a few transactions have many items. The volume of frequent weighted itemsets are also clustered around a mean 500. Refer to [4] for further information on data generation and additional experiments.

The quantitative association rule (QAR) has been proposed for the general numerical attribute values that are associated with every database record. Even though we are investigating a different scenario, QAR is the only other approach which we know that may be generalized to mine WARs. Therefore, we compare our approach with the QAR approach. In essence, the quantitative association rule approach [3] first quantizes the domain of every item, and map it into a binary value. Since at the beginning it is unknown that which items will appear in an itemset, all attribute domains have to be partitioned. If a numerical attribute is partitioned into ρ intervals, then these intervals are mapped into $\frac{\rho^2+\rho}{2}$ items because each of the original intervals and the combined intervals is treated as a different item. For the synthetic dataset generated above, if the weight domain of each weighted item is partitioned into 20 intervals, then there are over 2 million binary items for the QAR approach. The QAR approach utilizes a parameter $maxsup$ to reduce the resulting set of rules. It is clear that $maxsup$ can improve the performance, but could potentially degrade the quality of the results significantly. To analyze the performance and quality of these two approaches, we set $\rho = 20$ and $maxsup = minsup + 0.2$ for QAR. For the WAR approach, the density d and the total number of grids (N) for each itemset are set to 1.5 and 1 million, respectively.

The quality of the resulting rules is analyzed via two measurements: recall and accuracy. Since the datasets are generated artificially, we know the set R of all existing rules. First, we compute the volume of each rule $r \in R$, denoted by $V(r)$. For any rule set R', the *recall* (respectively, *accuracy*) of R' with respect to R is computed as follows. For each $r \in R$ ($r' \in R'$), find the rule $r' \in R'$ ($r \in R$) such that the overlap of r and r' is the largest, i.e., $V(r \cap r')$ is the largest. Next, the individual recall (accuracy) of r' (r) with respect to r (r') is computed as $\frac{V(r \cap r')}{V(r)}$ ($\frac{V(r \cap r')}{V(r')}$). The overall recall (accuracy) of R' with respect to R is the average of the individual recalls (accuracies) for each $r \in R$.

Figure 3(a) shows the average execution time of the two approaches with respect to $minsup$. WAR on average is a couple order of magnitude faster than QAR, even though the execution time of both approaches increases exponentially with decrease of $minsup$ because the complexity is approximately linear to the number of counters, which usually increases exponentially with smaller $minsup$. (Note that the Y-axis is in log-scale.) Figure 3(b) shows the recall ratio of the two algorithms with respect to the minimum support threshold. WAR has a significant higher recall value than QAR because a significant amount of important rules are pruned by the $maxsup$ threshold in QAR. Finally, we examine the accuracy of the two approaches. The weight domain of each item in WAR is partitioned into much finer grids than that in QAR for 2, 3, and 4-itemsets for which most rules exist. As a result, with the finer partition, the WAR approach produces much more accurate rules than the QAR approach as shown in Figure 3(c).

7. CONCLUSION

In this paper, we have investigated a new class of interesting problem: weighted association rules, which have wide implications. We proposed a two-fold approach, where the frequent itemsets are first generated (without considering weight) and then the maximum WARs are derived using an "ordered" shrinkage approach. Experimental results show that our proposed approach not only outperforms direct generalization of previous work by an order of magnitude, but also produces better quality results.

8. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. 20th VLDB*, 1994.

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining application. *Proc. ACM SIGMOD*, 1998.

[3] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. *Proc. ACM SIGMOD*, 1996.

[4] W. Wang, J. Yang, and P. Yu. Efficient mining of weighted association rules (WAR). *IBM Research Report RC 21692(97734)*, March, 2000.