International Journal of Computational Intelligence and Informatics, Vol. 2: No. 2, July - September 2012

A Weighted Utility Framework for Mining Association Rules using Closed Item sets

J.Kasthuri
Computer Science and Engineering
SNS College of Technology
Coimbatore, India
j9.kasthu@gmail.com

***Abstract*-Association rule discovery is used to identify relationship between the items from transaction databases. A traditional Association Rule Mining concentrates on qualitative aspects of attributes (significance, utility) as compared to quantitative attributes (no of appearances in a database). The qualitative approach is used for finding the best item sets. This approach does not yield a company's profit because the frequency of occurrence of items may be less. In Association Rule Mining the weight is associated with each item set by considering the significance of that item set in profit as well as frequency of occurrences of items in transactions. The name of this association rule mining is called Weighted Utility Association Rule Mining. The main challenge is weighted and utility framework does not hold anti-monotonic property. This framework produces many redundant rules. The proposed framework is used to generate non-redundant rules using a closed frequent item sets. This item sets are not losing any interesting and significant item sets.**

Keywords: association rules, closed itemsets, frequent itemsets, utility mining, weighted support.

I. INTRODUCTION

Data mining and knowledge discovery in databasesis an interesting areadeveloped in thelast fifteen years. Association Rule Mining (ARM) is one of the most important and well researched techniques of data mining. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Association rules are widely used in various areas such as telecommunication networks, market, risk management and inventory control. For example, a market basket database, it would beinteresting for decision support to know the fact that 30%of customers who bought coca powder and sugar alsobought butter. This analysis may be used to increase the sales. It also used to introduce fromfree schemes like, if 3kg of sugar is bought then 100gbutter free. In a census database, the inferenceacquired is that20% of persons who worked last year earned more thanthe average income, or in a medical database, that 35% ofpatients who have cold also have sinus.

Association Rule Mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. The problem is usually decomposed into two sub problems. One is to find those item sets whose occurrences exceed a predefined threshold in the database, those item sets are called frequent or large item sets. The second problem is to generate association rules from those large item sets with the constraints of minimal confidence. Suppose one of the large item sets is L_k, $L_k = \{I_1, I_2\ldots, I_k\}$, association rules with this item sets are generated in the following way. The first rule is $\{I_1, I_2,\ldots, I_{k-1}\} \rightarrow \{I_k\}$, by checking the confidence this rule can be determined as interesting or not. Then other rule are generated by deleting the last items in the antecedent and inserting it to the consequent, further the confidences of the new rules are checked to determine the interestingness of them. The second sub problem is quite straight forward, most of the researches focus on the first sub problem. The first sub-problem can be further divided into two sub-problems. Candidate large item sets generation process and frequent item sets generation process. The item sets whose support exceeds the support threshold as large or frequent item-sets, those item sets that are expected or have the hope to be large or frequent are called candidate item sets.

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 data set, they sometimes fail to associate user objectives and business values with the outcomes of the ARM analysis. For example, in retail mining application, frequent item sets identified by the standard ARM 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. Mostly all algorithms that are proposed to generate association rules are based on Apriori mining method. The performance of such algorithms is good for weakly correlated data as market basket data but is bad for correlated data such as census data.

The significance of the attributes in a transaction within the whole item space is considered to be same without its significance in traditional association rule mining. If the association rules are generated in this fashion, some interesting rules are missed. For example, [wine → salmon, 1%, 80%] may be more important than [bread → milk, 3%, 80%] even though the former holds a lower support. This is because those items in the first rule usually come with more profit per unit sale, but the standard ARM simply ignores this difference. Many techniques and algorithms have been proposed for mining association rules that consider the qualitative properties of attributes in the databases. The main challenge in mining weighted and utility association rules is that the anti-monotonic property 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.

II. BACKGROUND AND RELATED WORK

One major issue in association rule mining with weighted or utility settings is the invalidation of anti-monotonic property of item sets. Previous works considered item weights as their utility to reflect their significance in the dataset. This approach is different from all these in that define utility differently by considering the frequency of occurrences of databaseattributes 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. It defines item weight as a weighting function to signify an item differently in different domains. The weight reflects the significance of an item that is independent of transactions. This way it extracts those rules that have significant weight and high utility.The goals of this workshop are to find out the main implementation aspects (Bodon, 2003) of the Frequent Item set Mining problem for all, closed and maximal pattern mining tasks.Traditional association rule problem is extended in (Wang et al, 2000) the intensity of the item in the transaction is considered and a weight attribute is associated with each item based on its intensity. The rule generated from the items associated with weight is referred as weighted association rule (WAR). They also discussed how the confidence of the rules can be improved and effective target marketing can be achieved if the customers are divided based on their potential degree of loyalty or the volume they purchase. In WAR, the frequent item sets are found by ignoring the weight and the weight is associated during the generation of association rules.

(Rakesh & Ramakrishnana, 1994) Traditional association rule problem is extended in two new algorithms for solving the problem of discovering association rules between items in a large database of sales transactions that are fundamentally different from the known algorithms. Empirical evaluation shows that these algorithms outperform the known algorithms by factors ranging from three for small problems to more than an order of magnitude for large problems. The best features of the two proposed algorithms can be combined into a hybrid algorithm, called Apriori Hybrid.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 (Chan et al, 2003). Standard Downward Closure Property (DCP) is not valid in theWeighted Utility ARM 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 association rule deals with the importance of individual items in a database. Weighted Association Rule Mining where each item is assigned a weight according to its significance with respect to some user defined criteria. Weights may be set according to an item's profit margin. This generalized version of ARM is called Weighted Association Rule Mining (WARM). WARM model is given that uses a modified Apriori approach (Sulaiman et al, 2008) for binary and quantitative attributes.

Traditional model of ARM is adapted to handle weighted ARM problems where each item is allowed to have a weight. The goal is to steer the mining focus to those significant relationships involving items with significant weights rather than being flooded in the combinatorial explosion of insignificant relationships (Murtagh &Farid, 2003). The approach also has a valid DCP. But this model only considers 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.

III. Weighted Utility Association Rule Mining

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 item sets. 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.

A. Item Weights in Weighted ARM

Item Weight is a non-negative real value w(ij) given to each item ij ranging in [0…..1] with somedegree of importance, such that w(ij) = W(ij) 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.

B. Computing Transaction Weighted Utility

Transaction weighted utility is the aggregated weighted utilities of all the items present in a single transaction. Transaction weighed utility can be calculated as:

$$twu(t_1) = \frac{\sum_{j=1}^{|t_i|} t_i[(w(i_j), u)]}{|t_i|} \quad (1)$$

C. Computing Weighted Utility Support

Weighted utility support of an item set X Y is the fraction of transaction weighted utilities that contain both X and Y relative to the transactional weighted utility of all $S = \{s \mid s \in T, XUY \in S\}$ transactions. It can be formulated as

$$wus(XY) = \frac{\sum_{i=1}^{|s|} twu\ (t_i)}{\sum_{i=1}^{|t|} twu\ (t_i)} \quad (2)$$

Where

wus - weighted utility support

twu - transaction weighted utility

ti - transaction of item

Weighted Utility Support is modeled to measure the actual contribution of an item set in the dataset in WUARM.

D. Weighted Utility Anti-Monotonic Property

The difference between ARM is that Utility mining considers the quantity of every item, but ARM does not. The frequent item sets just reflect the number of transactions, which contain the item sets in databases. The item may not be frequent but a high utility item. WUARM, only the item sets are frequent with frequent sub sets. The monotonic property of item sets is always valid in the Weighted Utility framework and is stated using the lemma as follows:

Lemma: If an item set is not frequent them its superset cannot be frequent and wus (subset) ≥ wus (superset) is always true.

Proof: Given an item set X not frequent i.e. wus (X) min_ wus. For any item set Y, where X Y, i.e. superset of X, if a transaction t has all the items in Y, i.e. Y t, then that transaction must also have all the items in X, i.e. X t. The tx is to denote a set of transactions each of which has all the items in X, i.e. {tx | tx T, (ttx, X t)}. Similarly {ty | ty T, (t ty, Y t)}. Since X Y, txty. Therefore wus (tx) ≥ wus (ty). According to the definition of weighted utility support, the denominator stays the same, therefore wus(X) ≥ wus (Y). Because wus(X) < min_ wus, get wus(Y) < min_ wus, it proves that Y is not frequent if its subset is not frequent.

IV. PROPOSED WORK

In general, the association rules are generated in two steps. First, frequent item sets are found and secondly, rules are generated using the frequent item sets found in first step. Frequent item sets are also very huge in order to perform any analysis or for generating association rules. Instead, we are generating closed frequent item sets from which association rules can be formed.Generally, in generating closed frequent item sets, minimum support is only considered. But if minimum support alone is considered, some interesting / important items whose support < minimum support are lost. So, we consider a special attribute referred as weight which is associated with each item and has a value based on its durability / expiry / significance. For example, the lifetime of bread, cheese is less when compared to the lifetime of wheat, riceetc in market basket database.The quantity of purchase also can be considered as a weight. Closed frequent item sets are used to determine a reduced set of association rules. This item sets are not losing any interesting and significant item sets.

A. Closed Itemsets

Let T and I, $\mathrm{T} \subseteq D$ and $\mathrm{I} \subseteq I$, be subsets of all the transactions and items appearing in D, respectively. The concept of closed itemset is based on the two following functions f and g:

$$\mathrm{f}(\mathrm{T}) = \{i \in I \mid \forall t \in \mathrm{T}, i \in t\} \quad (3)$$

$$\mathrm{g}(\mathrm{I}) = \{t \in D \mid \forall i \in \mathrm{I}, i \in t\}$$

Function f returns the set of items included in all the transactions belonging to T, while function g returns the set of transactions supporting a given itemset I.

Definition 1: An itemset I is said to be closed if and only if $c(\mathrm{I}) = f(g(\mathrm{I})) = f \circ g(\mathrm{I}) = \mathrm{I}$ where the composite function $c = f \circ g$ is called Galois operator or closure operator.

The closure operator defines a set of equivalence classes over the lattice of frequent itemsets: two itemsets belong to the same equivalence class iff they have the same closure that is they are supported by the same set of transactions. It can also show that an itemset I is closed iff no supersets of I with the same support exist. Therefore mining the maximal elements of all the equivalence classes corresponds to mine all the closed itemsets. The itemsets with the same closure are grouped in the same equivalence class. Each equivalence class contains elements sharing the same supporting transactions, and closed itemsets are their maximal elements. All the algorithms for mining frequent closed itemsets adopt a strategy based on two main steps: Search space browsing, and Closure computation. In fact, they browse the search space by traversing the lattice of frequent itemsets from an equivalence class to another, and compute the closure of the frequent itemsets visited in order to determine the maximal elements (closed itemsets) of the corresponding equivalence classes.

Browsing the search space: The goal of an effective browsing strategy should be to identify exactly a single itemset for each equivalence class. It could in fact mine all the closed itemsets by computing the closure of just this single representative itemset for each equivalence class, without generating any duplicate. Let us call representative itemsets closure generators. Some algorithms choose the minimal elements (or key patterns) of each equivalence class as generators. Key patterns form a lattice, and this lattice can be easily traversed with a simple Apriori-like algorithm.Unfortunately, an equivalence class can have more than one minimal element leading to the same closed itemset. For example, the closed itemset {A,B,C,D} of could be mined twice, since it can be obtained as the closure of two minimal elements of its equivalence class, namely {A,B} and {B,C}.Other algorithms use a different technique, which we call closure climbing. As soon as a generator is devised, its closure is computed, and new generators are built as supersets of the closed itemset discovered so far.

Since closed itemsets are maximal elements of their own equivalence classes, this strategy always guarantees to jump from an equivalence class to another. Unfortunately, it does not ensure that the new generator belongs to an equivalence class that was not yet visited. Hence, similarly to the approach based on key patterns, we can visit multiple times the same equivalence class. For example, both {A,C} and {C,D} are generators of the same closed itemset {A,C,D}, and they can be obtained as supersets of the closed itemsets {C} and {D}, respectively. Hence, regardless of the strategy adopted, we need to introduce some duplicate checking technique in order to avoid generating multiple times the same closed itemset.

Computing Closures: To compute the closure of a generator , we have to apply the Galois operator . Applying requires to intersect all the transactions of the dataset including . Another way to obtain this closure is suggested by the following lemma:

Lemma 2: Given an item set X and an item $i \in I$, $g(X) \subseteq g(i) \Leftrightarrow i \in c(X)$

Proof:$(g(X) \subseteq g(i) \Rightarrow i \in c(X))$:

Since $g(X \cup i = g(X) \cap g(i))$, $g(X) \subseteq g(i) \Rightarrow g(X \cup i) = g(X)$.

Therefore, if $g(X \cup i) = g(X)$ then

$f(g(X \cup i)) = f(g(X)) \Rightarrow c(X \cup i) = c(X) \Rightarrow i \in c(X)$.

$(i \in c(X) \Rightarrow g(X) \subseteq g(i))$:If $i \in c(X)$, then $g(X) = g(X \cup i)$.

Since $g(X \cup i) = g(X) \cap g(i)$, $g(X) \cap g(i) = g(X)$ holdstoo.

Thus it can deduce that $g(X) \subseteq g(i)$.

From the above lemma, we have that if $g(X) \subseteq g(i)$ then $i \in c(X)$.Therefore, by performing this inclusion check for all the items in I not included in X , we can incrementally compute $c(X)$. Note that the set $g(i)$ can be represented by a list of transaction identifiers, i.e., the tidlist associated with i . This suggests the adoption of a vertical format for the input dataset in order to efficiently implement the inclusion check $g(X) \subseteq g(i$).Closure computations can be performed off-line or online. In the former case we first retrieve the complete set of generators, and then compute their closures. In the latter case, as soon as a new generator is discovered, its closure is computed on-the-fly. The algorithms that compute closures online are generally more efficient than those that adopt an offline approach, since the latter ones usually exploit key patterns as generators. Key patterns are the minimal item sets of the equivalence class, and thus are the shortest possible generators. Conversely, the on-line algorithms usually adopt the closure climbing strategy, according to which new generators are recursively created from closed item sets. These generators are likely longer than key patterns. Obviously, the longer a generator is, the fewer checks (on further items to add) are needed to get its closure.

B. Association Rule Generation

Generating rules is much less expensive than discovering closed frequent itemsets as it does not require examination of the data. Given a closed frequent itemset L, rule generation examines each non-empty subset a and generates the rule a ⇒ (L – a) with support = support (L) and confidence = support (L)/support (a). This computation can efficiently be done by examining the largest subsets of L first and only proceeding to smaller subsets if the generated rules have the required minimum confidence. For example, given a closed frequent

itemset ABCD, if the rule ABC $\Rightarrow$ D does not have Minimum confidence, neither will AB $\Rightarrow$ CD, and so we need not consider it.

V. Experimental Evaluation

Experiments were undertaken using three different association rule mining techniques. Three algorithms were used for each approach, namely Weighted ARM, Weighted Utility ARM and Weighted Utility with Closed ARM (WUCARM). Two types of experiments were carried out based on quality measures and performance measures. The quality measures are compared to the number of Association rules generated using three algorithms described with real data. In the second experiment, it showed the scalability of the proposed WUCARM algorithm by comparing the execution time of three algorithms with varying support thresholds.

A. Association Rules Comparison

The quality measure, each item is assigned a weight range between [0-1] according to their significance in the dataset. To generate artificial frequencies of items for real data to obtain items utilities in transactions. The x-axis shows minimum support from 20% to 100% and on the y-axis the numbers of association rules. WARM using weighted datasets and applying a post processing approach. If an item set is not frequent, and then its superset cannot be frequent and is always true. The results show quite similar behavior of the three algorithms to weighted ARM. As expected, the number of association rules increases as the minimum support decreases in all cases. The number of association rules generated using the weighted utility ARM algorithm are always less than the number of association rules generated by weighted ARM. Because weighted utility ARM uses association rules generated by weighted ARM. This generates less association rules and misses many potential ones.

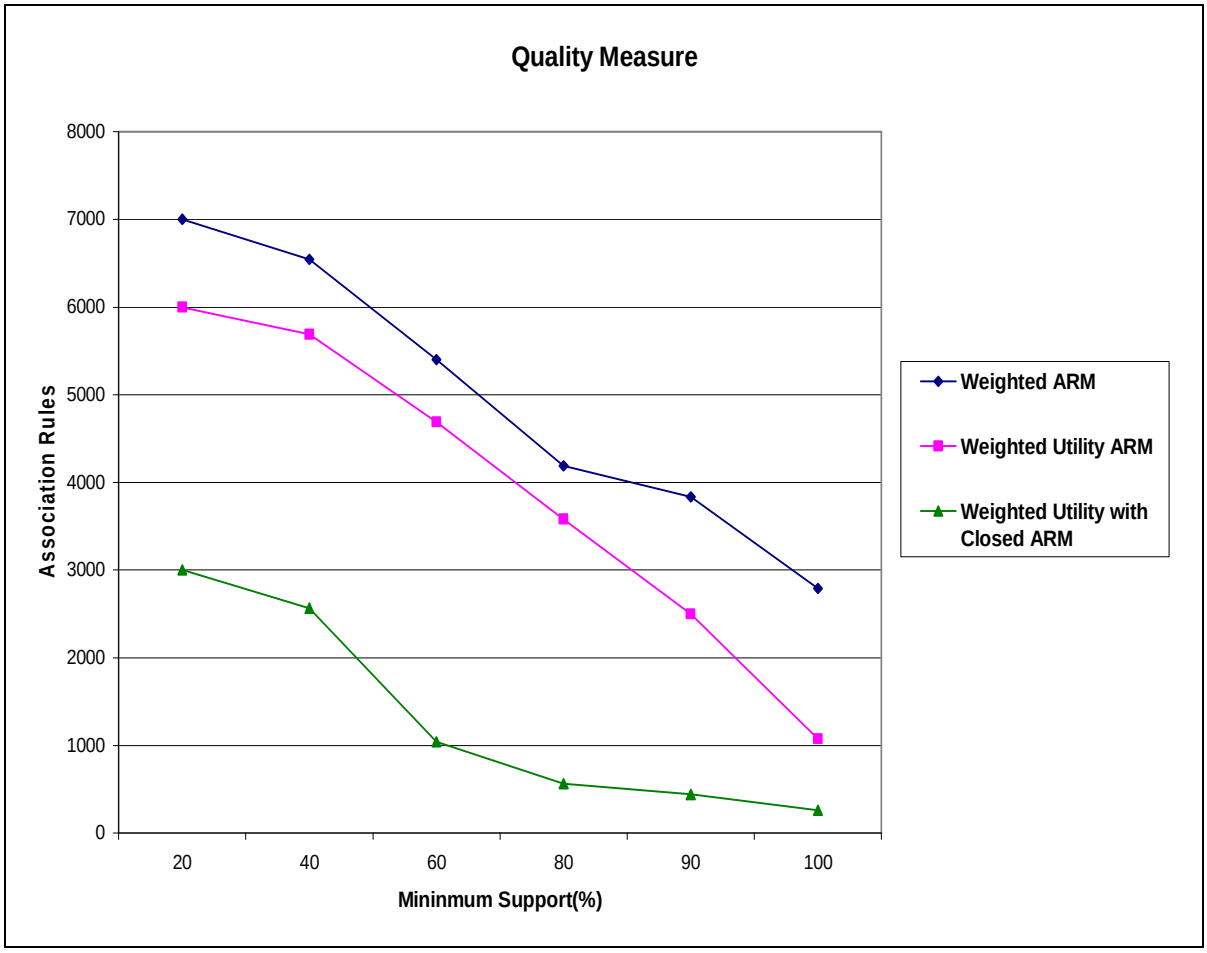

Figure.1 Number of Association Rules Generated using Varying Support Threshold

WUCARM generated fewer rules than weighted ARM and weighted utility ARM. Because it is not only considers the items weight but also take into account the items utilities in each transaction and considers potential item sets which weighted ARM ignores. Results of the proposed WUCARM approach are better than weighted utility ARM because it consider all the possible item sets, item weights and their utilities.

B. Performance Analysis

The performance measure, it compares the execution time of WUCARM algorithm with weighted ARM and weighted utility ARM algorithms using real data. It investigated the effect on execution time caused by varying the support threshold with fixed data size (number of records). Fig.1 and Fig.2, a support threshold from 20% to 100% is used again.WUCARM has comparatively low execution time due to the fact that it generates fewer rules than WUARM and do not use pre or post processing as mentioned earlier. Weighted utility ARM has slightly higher execution time due to the fact that Weighted Utility ARM initially uses weighted Association Rule Mining approach and then use already generated frequent sets for pruning, which takes computation time.

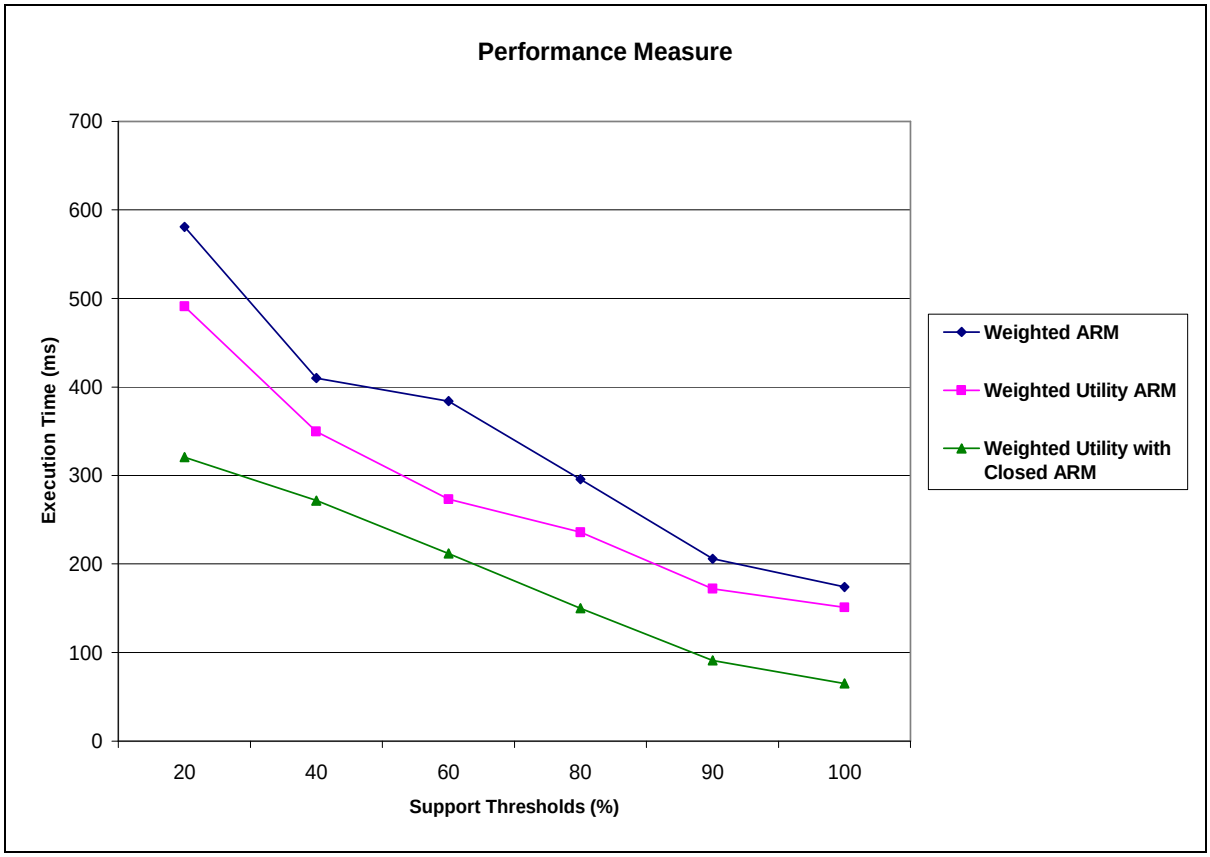

Figure.2 Time Analysis

VI. CONCLUSION

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. The main challenge is weighted and utility framework does not hold anti-monotonic property. This framework produces too many rules, most of which are redundant. The proposed framework is used to generate non-redundant rules using a closed frequent item sets. This itemsets can be obtained from frequent itemsets and are not losing any interesting and significant item sets.

REFERNCES

[1] A. Zubair Rahman and P. Balasubramanie, “Weighted Support Association Rule Mining using Closed Itemsets Lattices in Parallel”, Journal of Computer Science and Network Security, 2009, vol. 9, no. 3, pp. 247-253.

[2] F. Bodon, "A Fast Apriori Implementation," in Proc. Workshop on International Conference onData Mining, 2003, pp. 109-117.

[3] F. Tao and F. Murtagh, "Weighted Association Rule Mining using Weighted Support and Significance Framework", in Proc Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining, 2003, pp. 661-666.

[4] H. Yao, J. Hamilton and C.J Butz, "A Foundational Approach to Mining Item Set Utilities from Databases", in Proc. 4thInt. Conf. on Data Mining, 2004, pp.482-486.

[5] H. Jiawei, W. Jianyong, L. Ying and P. Tzvetkov, "Mining Top-K Frequent Closed Patterns without Minimum Support", in Proc. IEEE Int. Conf. on Data Mining, 2002, pp. 211-218.

[6] H. Jianying, M. Aleksandra, "High-Utility Pattern Mining : A Method for Discovery of High-Utility Item Sets", Association for Computing Machinery Transaction on Pattern Recognition, 2007, vol 40, no 11, pp 3317-3324.

[7] J. Kasthuri, "An Effective Mining Association Rules using Weighted Support with Closed Itemsets", presented at the Int. Conf. on Emerging Trends in Engineering Technologies, Nooral Islam University, Nagarcoil, 2010.

[8] K.M. Sulaiman, M. Muyeba and F. Coenen, "A Weighted Utility Framework for Mining Association Rules", in Proc of 2nd United Kingdom Society of Information Management European Symposium on Computer Modeling and Simulation, 2008, pp. 87-92.

[9] L. Claudio, O. Salvatore, P. Raffaele, "Fast and Memory Efficient Mining of Frequent Closed Itemsets", IEEE Transaction on Knowledge and Data Engineering, 2006, vol. 18, no. 1, pp. 21-36.

[10] R. Chan, Q. Yang and Y.D. Shen, "Mining High Utility Item Sets", in Proc. 3rd IEEE Int. Conf. onData Mining, 2003, pp. 19-25.

[11] W. Wang, J. Yang and P. Yu, "Efficient mining of weighted association rules (WAR)", Proceedings of Association for Computing Machinery Special InterestGroup on Knowledge Discovery and Data Mining, 2000, pp 270-274.

[12] Y. Hong, J. Hamilton, "Mining Itemsets Utilities from Transaction Databases", Association for Computing Machinery Transaction on Data and Knowledge Engineering, 2006, vol. 59, no. 3, pp.603-626.