ISSN:0975-9646

An overview of interval encoded temporal mining involving prioritized mining, fuzzy mining, and positive and negative rule mining

#C. Balasubramanian[1], K. Duraiswamy[2], V.Palanisamy[3]

[1] Associate Professor/CSE, P.S.R.Rengasamy College of Engineering for women,Sivakasi-626140,India
[2] Dean (Academic/ CSE), K.S.Rangasamy College of Technology, Tiruchengode-637215, India.
[3] Principal, Info institute of Engineering, Kovilpalayam, Coimbatore, India.

Abstract

Databases and data warehouses have become a vital part of many organizations. So useful information and helpful knowledge have to be mined from transactions. In real life, media information has time attributes either implicitly or explicitly called as temporal data. This paper focuses on an encoding method for the temporal database that reduces the memory utilization during processing. The first approach involves temporal mining applying the conventional algorithms like Apriori, AprioriTid and AprioriHybrid to an encoded temporal database that has a better performance than that when applied over a static database. The second approach involves weighted temporal mining over an encoded temporal database consisting of items which are prioritized by assigning weights. These weights are given according to the importance of the item from the user's perspective. A fuzzy mining approach involving AprioriTid for weighted association rule mining gives better results than quantitative values. Also a method for positive and negative temporal mining extends traditional associations to include association rules of forms A=> ¬B, ¬A => ¬B, A=> ¬B, which indicate negative associations between itemsets. The experimental results are drawn from the complaints database of the telecommunication system which presents the most feasible temporal mining method with reduced time and computational complexities.

Keywords

Encoded temporal database; temporal mining; weighted items; weighted temporal mining; fuzzy mining; positive and negative temporal mining

1. Introduction

Data mining is the discovery of knowledge from databases. It discovers useful information from large collections of data [1].

#Corresponding author .Tel. +91 4562 239085; fax: +91 4562 239084, E-mail addresses:rc.balsubramanian@gmail.com (C.Balasubramanian).

The discovered knowledge can be rules describing properties of the data, frequently occurring patterns, clusterings of the objects in the database, etc. The amount of data stored in database is growing rapidly. Intuitively, these large amounts of stored data contain valuable hidden knowledge. Data mining especially association rule discovery tries to find interesting patterns from databases that represent the meaningful relationships between items in transactions. Because the amount of these transaction data can be very large, efficient algorithms needs to be designed for discovering useful information.

An association rule describes the associations among items in which the presence of some items implies the presence of some other items. In order to find association rules, all large itemsets from a large database of transactions must be discovered. A large itemset is a set of items which appear often enough within the given set of transactions. The frequent itemset and association rules mining problem has received a great deal of attention and many algorithms have been proposed to solve this problem. Discovering association rules in these algorithms are usually done in two phases. In the first phase, the frequent itemset are generated and in the second phase, the interesting rules are extracted from these frequent itemset. If the support and confidence of a rule is above the minimum threshold, the rule will be of interest. A famous algorithm, called Apriori, was proposed in [2], which generates (k+1)-candidates by joining frequent k-itemset. So all subsets of every itemset must be generated for finding superior frequent itemset, although many of them may be not useful for finding association rules because some of them have no interesting antecedent or consequent in the rules. This process takes a long time. And it also requires thousands of times of database scan. The complexity of the calculation increases exponentially. Additionally, the size of database is the main problem of this algorithm. Some modified algorithms of Apriori (AprioriTid and AprioriHybrid) were proposed to solve this problem but these algorithms also have the database size problem. A method to encoding the database and an algorithm, which is called anti-Apriori algorithm, was used. By this, only the frequent itemsets that are of interest and can be converted

into association rules are generated, so it has a lower complexity of time and space. At the meantime, the times of the database scan are also reduced.

In real life, media information has time attributes either implicitly or explicitly. This kind of media data is called temporal data. Temporal data exist extensively in economical, financial, communication, and other areas such as weather forecast. Unlike conventional data mining, temporal data mining has its exclusive characteristics. Temporal data mining can be classified into temporal schema and similarity. Temporal schema mining focuses on time-series schema mining, temporal causal relationship mining, and association rules mining. While similarity study is mainly concentrated on query algorithms, such as the design of similarity query algorithms and the development of similarity query language [3]. A temporal database consists of transactions of the form:

$<TID, CID, I_1 \ldots I_m, t_s, t_e>$

where TID - Transaction ID

CID - Customer ID
$I_1 \ldots I_m$ - items
$[t_s, t_e]$ - the valid time range for

the transaction.

A temporal association rule is a binary form (AR, TimeExp), in which the left-hand side element "AR" is an association rule expressed as,

$X \Rightarrow Y, X \subset I, Y \subset I, X \cap Y = \varnothing$

where "TimeExp" is a time expression [3].

This paper includes a method of encoding the temporal database. It also focuses on a collection of algorithms for an encoded temporal database with relatively better performance using a) the conventional approaches, b) weighted temporal mining, c) fuzzy mining using linguistic terms, d) positive and negative association rule mining which extends traditional associations to include association rules of forms A=> ¬B, ¬A => ¬B, A=> ¬B, which indicate negative associations between itemsets. Large databases can be scaled with a pruning strategy and an interestingness measure [5].

The rest of the paper is organized as follows: Section2 briefs about the research works in related areas. Section3 describes the interval based encoding method. Section4 gives the extension of the conventional methods to encoded temporal mining.Section5 gives the overview of weighted temporal mining. Section6 projects the application of fuzzy mining over an encoded temporal database. Section7 outlines positive and negative temporal mining. Section8 gives the performance evaluation of the various temporal mining methods. Finally, section9 gives the conclusion and the possible extensions that could be done over the current work to gain an increase in the efficiency.

2. Foundation research works for temporal mining

Data mining process identifies the important associations among items. An association is said to be present among the items if the presence of some items also means the presence of some other items [5]. Several mining algorithms have been proposed to find the association rules from the transactions [5][6][7][8]. The large itemsets were identified to find the association rules. First, the itemsets which satisfy the predefined minimum support were identified and these were called the large itemsets. Then, the association rules were identified from these itemsets. The association rules which satisfy the predefined minimum confidence were the association rules produced as the output [6]. Also, in the Graph-Based approach for discovering various types of association rules based on large itemsets, the database was scanned once to construct an association graph and then the graph was traversed to generate all large itemsets [9]. This method avoids making multiple passes over the database.

In addition to the above mentioned method of association rule mining, which overlooks time components that are usually attached to transactions in databases, the concept of temporal mining was proposed giving importance to the time constraints [3]. The concept of valid time was used to find out the time interval during which a transaction is active. Time interval expansion and mergence was performed which gives importance to the time at which a transaction had taken place, before the application of the graph mining algorithm [9] to identify the temporal association rules. For discovering association rules from temporal databases [10], the enumeration operation of the temporal relational algebra was used to prepare the data. The incremental association rule mining technique was applied to a series of datasets obtained over consecutive time intervals to observe the changes in association rules and their statistics over the time. Temporal issues of association rules was addressed with the corresponding algorithms, language and system [11][12] for discovering temporal association rules.

Further, to mine rules based on the priority assigned to the elements, weighted mining was proposed to reflect the varying importance of different items [13]. Each item was attached a weight, which was a number given by users. Weighted support and weighted confidence was then defined to determine interesting association rules.

In general, a fuzzy approach leads to results which are similar to human reasoning. A fuzzy approach involving the enhancement over AprioriTid algorithm was identified, which had the advantage of reduced computational time [14]. Also, in the mining of fuzzy weighted association rules [15], great importance had been given to the fuzzy mining concept. Positive and negative mining include an approach that identifies frequent itemsets and infrequent itemsets of interest using a pruning strategy [4].

This paper proposes algorithms for an encoded temporal mining on the basis of conventional methods, prioritized mining, fuzzy mining, and positive and negative rule mining methods to identify association rules from an encoded temporal database that consist of transactions with their corresponding valid time intervals.

3. Interval based encoding method

The most common approach to interval encoding of temporal databases is to use intervals as codes for one dimensional set of time instants. The choice of this representation is based on the following empirical observation: Sets of time instants describing the validity of a particular fact in the real world can be often described by an interval or a finite union of intervals. For simplicity, a discrete integer-like structure of time is assumed. However, dense time can also be accommodated by introducing open intervals.

Let Interval-based Domain be TI and let TP = (T, <) be a discrete linearly ordered point-based temporal domain. The set I(T) is defined as

$I(T) = \{(a, b) : a \leq b, a \in T \cup \{-\infty\}, b \in T \cup \{\infty\}\}$

where < is the order over TP extended with $\{(-\infty,a), (a,\infty), (-\infty,\infty): a \in T\}$. The elements of I(T) are denoted by [a,b] which is the usual notation for intervals. The four relations on the elements of I(T) denoted by [a,b] are defined as follows:

$([a, b] < -- [a', b']) \Leftrightarrow a < a'$

$([a, b] < +- [a', b']) \Leftrightarrow b < a'$

$([a, b] < -+ [a', b']) \Leftrightarrow a < b'$

$([a, b] < ++ [a', b']) \Leftrightarrow b < b'$

for $[a, b], [a', b'] \in I(T)$.

The structure TI = (I(T), <--, < +- , <-+, <++) is the interval based temporal domain corresponding to TP.

A concrete (timestamp) temporal database is defined analogously to the abstract (timestamp) temporal database. The only difference is that the temporal attributes range over intervals (TI) rather than over the individual time instants (TP). Based on this concept of interval based encoding the temporal database is encoded using the valid time.

4. Temporal Mining using conventional approaches

4.1 Time based extension of Apriori

Temporal association rules can be viewed as a factor of derivative consideration time, when mining association rules. Recently, the Apriori algorithm is adapted to temporal association rules mining. Time-interval expansion, and mergence, is combined with the Apriori algorithm to association rules mining on datasets that have valid-time constraints. The key implementation is to add a valid-time attribute on association patterns. If a given tuple owns attribute A, the problem can be decomposed into two sub problems. (1) Check if the valid-time attribute of a given tuple matches another attributes. (2) Check whether the tuple that does not attach any valid-time attribute has attribute "A". The logical linkage between two sub problems is that if the sub problem (1) holds, sub problem (2) must hold; otherwise; if sub problem (2) doesn't sub problem (1) mustn't hold. According to the first case, database can be scanned to solve these two sub problems simultaneously. Time complexity and space complexity of this case are almost the same as those of Apriori algorithm.

Let min_s and min_c represent minimum support threshold and minimum confidence threshold respectively. If and only if support ≥ min_s, and confidence ≥ min_c, during $[t_s, t_e]$, rule $X \Rightarrow Y$ is a temporal association rule, which could be described as,

$X \Rightarrow Y$ (support, confidence, $[t_s, t_e]$)

The support of the association rule $X \Rightarrow Y$ is the probability that $X \cup Y$ exists in a transaction in the database D. The confidence of the association rule $X \Rightarrow Y$ is the probability that Y exists given that a transaction contains X , represented by eqn (1) as follows,

$$Pr(Y/X) = \frac{Pr(X \cup Y)}{Pr(X)} \quad (1)$$

The Apriori and AprioriTid algorithms generate the candidate itemsets to be counted in a pass by using only the itemsets found large in the previous pass, without considering the transactions in the database. The basic intuition is that any subset of a large itemset must be large. Therefore the candidate itemsets having k items can be generated by joining large itemsets having k-1 items, and deleting those that contain any subset that is not large. This procedure results in generation of a much smaller number of candidate itemsets. The AprioriTid algorithm has the additional property that the database is not used at all for counting the support of candidate itemsets after the first pass. Rather, an encoding of the candidate itemsets used in the previous pass is employed for this purpose. In later passes, the size of this encoding can become much smaller than the database, thus saving much reading effort. Based on the observations of Apriori and AprioriTid, a hybrid algorithm which is called as AprioriHybrid uses Apriori in the initial passes and switches to AprioriTid when the candidate itemset at the end of the pass will fit into the memory.

4.2 Anti-Apriori for a temporal database

All Apriori-like algorithms for itemset mining start from finding frequent 1-itemset. In these algorithms, finding frequent itemset is done in bottom up manner. Different from

these algorithms, anti-Apriori discovers frequent itemsets in up to down style. It means that the large frequent itemsets are found at first and then all of their subsets (that are certainly frequent) are extracted. In this technique, it is supposed that any frequent itemset must occur at least one time in the transactions lonely (without any other items that are not members of that itemset).

5. Weighted Temporal Mining

The temporal database consists of items which are prioritized by assigning weights. These weights are given according to the importance of the item from the user's perspective. Given a set of items $I = \{i_1, i_2, \dots, i_n\}$, a weight w_j for each item i_j, with $0 \le w_j \le 1$ is assigned where $j = \{1, 2, \dots, n\}$, to show the importance of the item. The weighted support for the weighted association rules can be defined by eqn (2) as

$$\left(\sum_{i_j \in (XUY)} W_j \right) X \, (Support\ (XUY)) \qquad (2)$$

The confidence is given by eqn (3) as follows

$$Confidence = \frac{Weighted\ Support(X \cup Y)}{Weighted\ Support(X)} \qquad (3)$$

A support threshold and a confidence threshold will be assigned to measure the strength of the association rules.

Given a database with T transactions belonging to a specified duration $[t_s, t_e]$, the bounded support (BS) of a large k-itemset X is defined to be the transaction number containing X within the specified valid time, and it must satisfy eqn(4) given by:

$$BS(X) \ge \frac{T \, x \, wminsp}{\sum_{\forall i \in X} W_i} \qquad (4)$$

where W_i is the summation of the weights of all the items in large k-itemset X in a specific duration $[t_s, t_e]$ and wminsp is the weighted minimum support. The bounded support value according to equ(4) is calculated for each large k-itemset so that the itemset which are not necessary for further calculations can be avoided. The bounded support calculations are done so that the necessary pruning may be performed and time may be saved.

6. Fuzzy Mining for an encoded temporal database

Fuzzy set theory is being used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning. Therefore to use fuzzy sets in data mining, a mining approach that integrates fuzzy set concepts with the AprioriTid mining algorithm has been identified. It finds interesting itemsets and fuzzy association rules in transaction data with quantitative values. The role of fuzzy sets helps transform quantitative values into linguistic terms, which reduces possible itemsets in the mining process. They are used in the AprioriTid data mining algorithm to discover useful association rules from quantitative values.

The fuzzy mining algorithm first transforms each quantitative value into a fuzzy set with linguistic terms using membership functions. The algorithm then calculates the scalar cardinality of each linguistic term on all transaction data using the temporary set. Each attribute uses only the linguistic term with the maximum cardinality in later mining processes, which keeps the number of items the same as that of the original attributes. The mining process based on fuzzy counts is then performed to find fuzzy association rules. The fuzzy mining approach involving AprioriTid has the following important points

- The fuzzy AprioriTid algorithm is applied on an encoded temporal database
- The temporal database has weighted items
- The weighted minimum support is used to calculate the support.

The use of triangular membership function in fuzzy mining is shown in fig1 as follows:

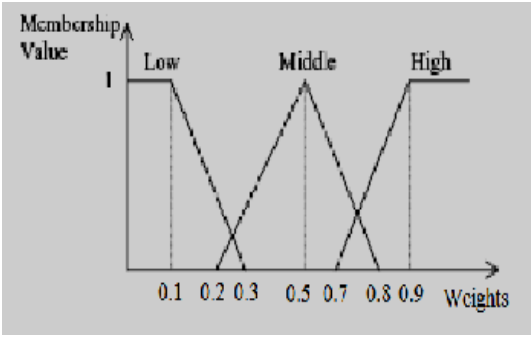

Fig.1. Triangular membership function.

Thus fuzzy mining leads to identifying association rules in terms of linguistic terms rather than quantitative values.

7. Positive and negative temporal rule mining

This is another approach for the generation of itemsets in the case of a temporal database which is more efficient than the Apriori. In this approach frequent itemsets and infrequent itemsets of interest are identified using a pruning strategy.

By definition, an association rule is an implication of the form $A \Rightarrow B$, where *A,* B are frequent itemsets in a transaction database and $A \cap B = \varnothing$. A frequent itemset is an itemset that meets the user-specified minimum support. An infrequent itemset is an itemset that does not meet the user specified minimum support. The negation of an itemset A is indicated by $\neg A$. The support of $\neg A$ is $supp(\neg A) = 1 - supp(A)$. In the pruning strategy, a rule is not interesting if its antecedent and consequent are approximately independent. An interestingness function $interest(X,Y) = |supp(X \cup Y) \leftarrow supp(X)supp(Y)|$ and a

threshold *mi* is defined. Integrating the interest(X, Y) mechanism, I is a frequent itemset of potential interest if:

$$f_{ipi}(I) = supp(I) \leftarrow ms \wedge \exists X,Y : X \cup Y = I \wedge f_{ipis}(X,Y) \quad (5)$$

where

$$f_{ipis}(X,Y) = X \cap Y = \varnothing \wedge f(X,Y,ms,mc,mi) = 1 \quad (6)$$

$$f(X,Y,ms,mc,mi) = \frac{supp(X \cup Y) + conf(X \Rightarrow Y) + interest(X,Y) \leftarrow (ms + mc + mi) + 1}{|supp(X \cup Y) \leftarrow ms| + |conf(X \Rightarrow Y) \leftarrow mc| + |interest(X,Y) \leftarrow mi| + 1}$$

where f() is a constraint function concerning the support, confidence, and interestingness of X⇒Y. Similarly, J is an infrequent itemset of potential interest if:

$$iipis(J) = supp(J) < ms \wedge \exists X,Y : X \cup Y = J \wedge iipis(X,Y) \quad (7)$$

where

$$iipis(X,Y) = X \cap Y = \varnothing \wedge g(X, \neg Y, ms, mc, mi) = 2 \quad (8)$$

$$g(X, \neg Y, ms, mc, mi) = f(X, \neg Y, ms, mc, mi) + \frac{supp(X) + supp(Y) \leftarrow 2ms + 1}{|supp(X) \leftarrow ms| + |supp(Y) \leftarrow ms| + 1}$$

where *g*() is a constraint function concerning *f*( ) and the support, confidence, and interestingness of X⇒Y. Infrequent itemsets of potential interest for rules of the forms of ¬X ⇒Y and ¬X ⇒¬Y can also be defined. Then search for frequent, infrequent itemsets and positive and negative association rule mining are done.

8. Performance Evaluation

Encoding a temporal database based on the time interval or valid time has been found to be more effective and reduced the memory requirements. This is depicted by fig2 given below

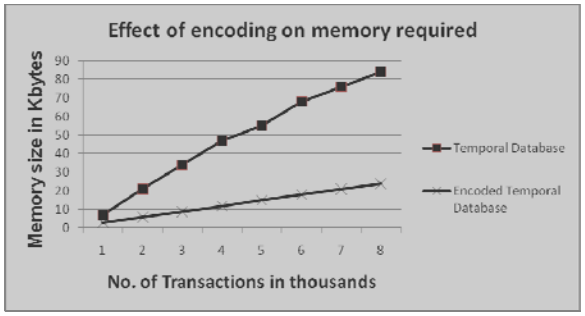

Fig.2. Effect of encoding on memory requirements

The performance comparison of the Apriori family of algorithms is as shown by fig3 below. The Anti-Apriori algorithm is found to have a better performance than the others

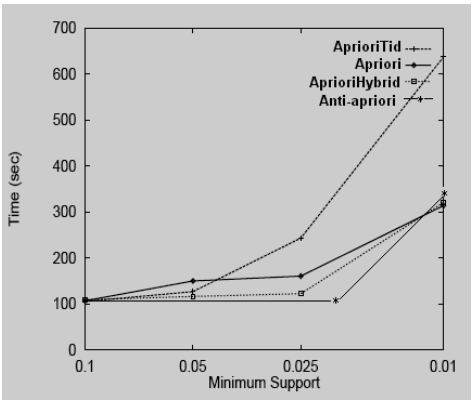

Fig.3. Performance of Apriori family and Anti-Apriori algorithms

The performance chart given below by fig4 shows that the weighted temporal mining is more effective than the conventional approaches

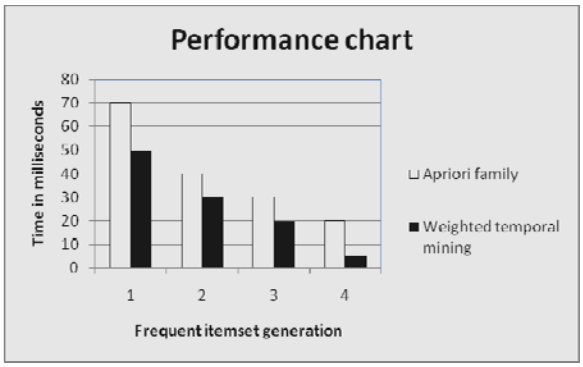

Fig.4. Performance chart for Apriori and Weighted temporal mining

The fuzzy mining approach produced temporal rules using linguistic terms which are found to be similar to human reasoning. The performance of fuzzy mining is compared with that of Apriori in fig5 as follows:

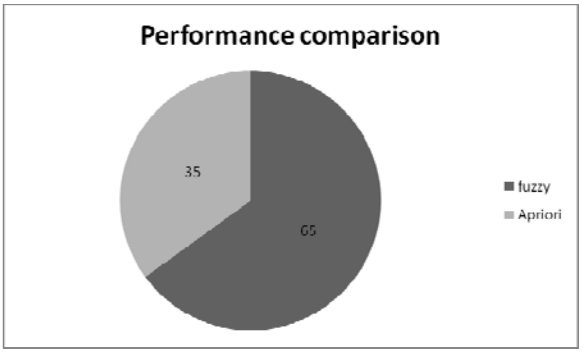

Fig.5. Performance comparison of fuzzy and Apriori

Positive and negative rule mining produced rules giving importance to both frequent and infrequent itemsets so that none of the important rules which are practically possible are omitted. Positive and negative rule mining employs a pruning strategy whereas the conventional approach does not. The performance of both these approaches is compared in fig6 as follows:

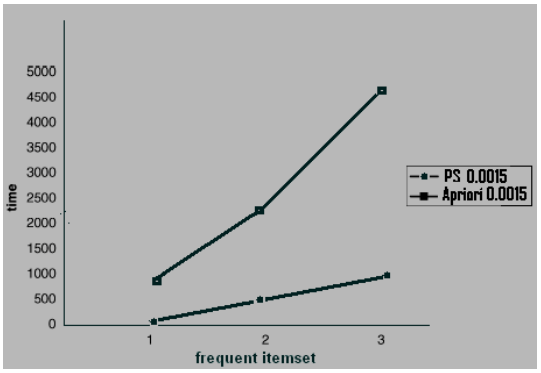

Fig.6. Performance comparison of PS(Pruning Strategy) and Apriori

9. Conclusion and future enhancements

This paper gave an overview of the various techniques that may be applied on an encoded temporal database. Each of the algorithms of the Apriori family had a different impact and produced effective results. The effect of Apriori and anti-Apriori algorithms on a temporal data base which has been encoded by an encoding method has more openings to provide advantageous results in terms of lower complexities of time and space. Weighted temporal mining which prioritizes the items had a better performance than the conventional approaches. Fuzzy mining had the advantage of producing rules in a more understandable manner. Applying the algorithm with pruning strategy to identify all itemsets of interest both frequent and infrequent has better efficiency than the Apriori-like algorithm which does the same with no pruning strategy.

Future developments may involve overcoming the problems involved in deciding what the threshold values of support and confidence should be. This problem may be overcome by using adjusted difference analysis to identify interesting associations among attributes, which does not require any user supplied thresholds. Also weights could be assigned automatically to prioritize the items so that human intervention can be minimized and starvation could be avoided.

References

[1] Abdullah Uz Tansel, Susan P. Imberman, Discovery of Association Rules in Temporal Databases, International Conference on Information Technology, 2007.

[2] Margaret H Dunham, Data Mining Introductory and Advanced Topics, Tsinghua University Press, Beijing, 2003.

[3] Hui Ning, Haifeng Yuan, Shugang Chen, Temporal Association Rules in Mining Method, Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences,2006.

[4] Xindong Wu, Chengqi Zhang, Shichao Zhang, Efficient mining of both positive and negative association rules, ACM Transactions on Information Systems, Vol. 22, No. 3, July 2004, Pages 381–405.

[5] R. Agrawal, T. Imielinksi, A. Swami, Mining association rules between sets of items in large database, The 1993 ACM, SIGMOD Conference, Washington DC, USA, 1993.

[6] R. Agrawal, R.Srikant, Fast algorithm for mining association rules, The International Conference on Very Large Data Bases, 1994, pp. 487–499.

[7] R. Agrawal, T. Imielinksi, A. Swami, Database mining: a performance perspective, IEEE Trans. Knowledge Data Eng. 5 (6) (1993) 914–925.

[8] R. Agrawal, R. Srikant, Q. Vu, Mining association rules with item constraints, The Third International Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, California, August 1997.

[9] S.J. Yen, A.L.P. Chen, A Graph Based Approach for Discovering Various Types of Association Rules, IEEE Transactions on Knowledge and data Engineering, Vol.13, No.5, September/October 2001.

[10] A.U.Tansel, S.P.Imberman, Discovery of Association Rules in Temporal Databases, International Conference on Information Technology,2007.

[11] X. Chen, I. Petrounias, H. Heathfield, Discovering Temporal Association Rules in Temporal Databases, Proc.of IADT'98,Berlin, Germany, pp.312-319.

[12] X. Chen, I. Petrounias, Mining Temporal Features in Association Rules, Proc. of PKDD'99,Prague,Czech Republic, pp.295-300.

[13] C.H. Cai, W.C. Fu, C.H. Cheng, W.W. Kwong, Mining association rules with weighted items, The International Database Engineering and Applications Symposium, 1998, pp. 68–77.

[14] T.P. Hong, C.S. Kuo, S.L. Wang, A Fuzzy AprioriTid mining algorithm with reduced computational time, Tenth International Conference on Fuzzy Systems, 2004.

[15] D.L.Olson, Y.Li, Mining Fuzzy Weighted Association Rules,Proceedings of the 40th Hawaii International Conference on System Sciences, 2007.