International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 11 – No. 8, January 2017 – www.ijais.org

The Weekly Mining of Fuzzy Patterns from Temporal Datasets

Md Husamuddin
College of Computer Science and IT
AlBaha University
AlBaha, KSA

ABSTRACT

The process of extracting fuzzy patterns from temporal datasets is a well known data mining problem. Weekly pattern is one such example where it reflects a pattern with some fuzzy time interval every week. This process involves two steps. Firstly, it finds frequent sets and secondly, it finds the association rules that occur in certain time intervals weekly. Most of the fuzzy patterns are concentrated as user defined. However, the probability of user not having prior knowledge of datasets being used in some applications is more. Thus, resulting in the loss of fuzziness related to the problem. The limitation of the natural language also bounds the user in specifying the same. This paper, proposes a method of extracting patterns that occur weekly in a particular fuzzy time frame and the fuzzy time frame is generated by the method itself. The efficacy of the method is backed by the experimental results

Keywords

Temporal Patterns, Temporal Association rules, Superimposed intervals, Fuzzy set, Right reference functions, left reference functions, Membership functions.

1. INTRODUCTION

The analysis of transactional data is given significant importance among the various types of data mining applications. The most common example of such transactional data is '*market basket'* data. In a *market basket* data set, each transaction is a collection of items bought by a customer at one time. The notion proposed in [1] is to capture the co-occurance of items in transactions, given minimum support and minimum confidence threshold. Adding the time attribute to the above problem is one of the extensions. Whenever a transaction occurs, the time of transaction is automatically recorded. *Ale et al* [2] has proposed a method of extracting association rules that occur within a life span of a corresponding itemset.

The concept of locally frequent itemset has been proposed by *Mahanta et al* [3], where patterns occur frequently in a particular time frame and may or may not occur frequently throughout the life-span of the itemset. In [3] an algorithm has been proposed to extract such itemsets along with a list of sequences of time frames. Here each frequent itemset is linked with a sequence of time frames where it is frequent. In [4], the time stamp is considered as calendar dates and a method is discussed which can extract yearly, monthly and daily periodic or partially periodic patterns. If these periodic patterns are in a compact manner using the method discussed. In [4], it turns out to be a fuzzy time interval. In [5], a superimposition method is used for overlapping the frequency of intervals. Considering the time-stamp as *year_month_day_hour_minute_second*, methods were proposed in [6, 7, 8,9], for extracting yearly, monthly, and daily fuzzy frequent itemsets. This paper discusses weekly fuzzy patterns and devised an algorithm for extracting such patterns**.** The algorithm discussed in this similar to that in [6, 7, 8, 9]**.** The paper is organized as follows. In section-II related works are discussed. Section-III discusses terms, definitions and notations used in the algorithm. In section-IV, the algorithm is discussed. Section-V discusses about results and analysis. Finally a summary and lines for future works are discussed in section-VI.

2. RELATED WORKS

Agrawal el al [1] first formulated the problem of discovering of association rules. Given a set *'S'* of items and a large collection 'C' of transactions containing the items, the query is to find the association among the presence of various items in the transactions. An important extension of conventional data mining is Temporal Data Mining [10]. More interesting patterns can be extracted, if time aspect is taken into account. The association rule discovering process is also extended to incorporate temporal aspects. The query associated is to extract valid time frames during which association rules occur and the discovery of possible periodicities that association rules consists of. In [2], the authors proposed an algorithm for the discovery of temporal association rules. For each item (or itemset), a life-span is defined which is the time gap between the first occurrence and the last occurrence of the item in the transaction in the dataset. Thus each rule is associated with a time frame. In [3], the works done in [2] has been extended by considering time gap between two consecutive transactions containing an item set into account.

Ozden [11] proposed a method, considering the periodic nature of patterns which is able to extract patterns having periodic nature where the time period is specified by the user. In [12], *Li et al* discussed a method to find temporal association rules corresponding to *fuzzy match* i.e. association rule holding during "enough" number of intervals given by the corresponding calendar pattern. Similar works were done in [13] with multiple granularities of time intervals (e.g. first working day of every month) where both cyclic and user defined calendar patterns can be achieved.

Many researchers have studied about extracting fuzzy patterns from datasets. In [14], the authors_proposed a method for extracting fuzzy temporal patterns from a given process instance. Similar work is done in [15]. In [16], a method of mining fuzzy periodic association rules is discussed. In [6], authors have discussed a method for finding yearly fuzzy patterns. In [7], methods for mining monthly fuzzy patterns are discussed. In [8], methods are discussed for finding daily fuzzy patterns. In [9], similar methods are discussed for extracting hourly fuzzy patterns.

3. TERMS, DEFINITIONS AND NOTATIONS

Let us review some definitions and notations used in this paper.

Let U be the universe of discourse. A fuzzy set X in U is characterized by a membership function $X(a)$ lying in [0, 1]. $X(a)$ for $a \in U$ represents the grade of membership of a in X. Thus a fuzzy set X is defined as

$$X = \{(a, X(a)), a \in U\}$$

A fuzzy set X is said to be normal if $X(a) = 1$ for at least one $a \in U$. An α-cut of a fuzzy set is an ordinary set of elements with membership grade greater than or equal to a thresholdα, $0 \leq \alpha \leq 1$. Thus an α-cut X_α of a fuzzy set X is characterized by $X_\alpha = \{a \in U; X(a) \geq \alpha\}$ [see e.g. [17]]

A fuzzy set is said to be convex if all its α-cuts are convex sets.

A fuzzy number is a convex normalized fuzzy set X defined on the real line R such that

1. there exists an $a_0 \in R$ such that $X(a_0) = 1$, and
2. $X(a)$ is piecewise continuous.

Thus a fuzzy number can be thought of as containing the real numbers within some interval to varying degrees.

Fuzzy intervals are special fuzzy numbers satisfying the following.

1. there exists an interval $[x, y] \subset R$ such that $X(a_0)$=1for all $a_0 \in [x, y]$, and
2. $X(a)$ is piecewise continuous.

A fuzzy interval can be thought of as a fuzzy number with a flat region. A fuzzy interval X is denoted by $X = [x, y, z, w]$ with $x < y < z < w$ where $X(x) = X(w) = 0$ and $X(a) = 1$ for all $a \in [y, z]$. $X(a)$ for all $a \in [x, y]$ is known as *left reference function* and $X(a)$ for $a \in [z, w]$ is known as the *right reference function*. The *left reference function* is non-decreasing and the *right reference function* is non-increasing

The *support* of a fuzzy set X within a universal set U is the crisp set that contains all the elements of U that have non-zero membership grades in X and is denoted by $S(X)$. Thus

$$S(X) = \{ a \in U; X(a) > 0\}$$

The *core* of a fuzzy set X within a universal set U is the crisp set that contains all the elements of U having membership grades 1 in X.

3.1 Set Superimposition

In [18] an operation called *superimposition* denoted by (S) was proposed. If X is superimposed over Y or Y is superimposed over X, then

$$X \ (S) \ Y = (X-Y) \ (+) \ (X \cap Y)^{(2)} \ (+) \ (Y-X) \quad \ldots. \ (1)$$

Where $(X \cap Y)^{(2)}$ are the elements of $(X \cap Y)$ represented twice, and (+) represents union of disjoint sets.

To explain this, an example has been taken.

If $X = [x_1, y_1]$ and $Y = [x_2, y_2]$ are two real intervals such that $X \cap Y \neq \phi$, it would get a superimposed portion. It can be seen from (1)

$$[x_1, y_1] \ (S) \ [x_2, y_2] = [x_{(1)}, y_{(2)}) \ (+) \ [x_{(2)}, y_{(1)}]^{(2)} \ (+) \ (y_{(1)}, y_{(2)}] \quad \ldots.. \ (2)$$

where $x_{(1)} = min(x_1, x_2)$ $x_{(2)} = max(x_1, x_2)$

$y_{(1)} = min(y_1, y_2)$, and $y_{(2)} = max(y_1, y_2)$

(2) Explains why if two line segments are *superimposed*, the common portion looks doubly dark [5]. The identity (2) is called *fundamental identity of superimposition* of intervals.

Let now, $[x_1, y_1]^{(1/2)}$ and $[x_2, y_2]^{(1/2)}$ be two fuzzy sets with constant membership value ½ everywhere (i.e. equi-fuzzy intervals with membership value ½). If $[x_1, y_1] \cap [x_2, y_2] \neq \phi$ then applying (2) on the two equi-fuzzy intervals it can be written as

$$[x_1, y_1]^{(1/2)} (S) [x_2, y_2]^{(1/2)} = [x_{(1)}, x_{(2)})^{(1/2)} (+) [x_{(2)}, y_{(1)}]^{(1)} (+) (y_{(1)}, y_{(2)}]^{(1/2)} \quad \ldots (3)$$

Let $[a_i, b_i]$, i=1,2,…,n, be n real intervals such that $\bigcap_{i=1}^{n} [a_i, b_i] \neq \phi$. Generalizing (3) gives

$$[a_1, b_1]^{(1/n)} \ (S) \ [a_2, b_2]^{(1/n)} \ (S) \ldots (S) \ [a_n, b_n]^{(1/n)} = [a_{(1)}, a_{(2)})^{(1/n)} \ (+) \ [a_{(2)}, a_{(3)})^{(2/n)} \ (+) \ldots (+) \ [a_{(r)}, a_{(r+1)})^{(r/n)} (+) \ldots (+) \ [a_{(n)}, b_{(1)}]^{(1)} (+) (b_{(1)}, b_{(2)}]^{((n-1)/n)} (+) \ldots (+) \ (b_{(n-r)}, b_{(n-r+1)}]^{(r/n)} (+) \ldots (+) (b_{(n-2)}, b_{(n-1)}]^{(2/n)} (+) (b_{(n-1)}, b_{(n)}]^{(1/n)} \quad \ldots \ (4)$$

In (4), the sequence $\{a_{(i)}\}$ is formed by sorting the sequence $\{a_i\}$ in ascending order of magnitude for i=1,2,…n and similarly $\{b_{(i)}\}$ is formed by sorting the sequence $\{b_i\}$ in ascending order.

Although the set superimposition is operated on the closed intervals, it can be extended to operate on the open and the half-open intervals in the trivial way.

3.1 Lemma 1. (The Glivenko-Cantelli Lemma Of Order Statistics)

Let $A = (A_1, A_2, \ldots, A_n)$ and $B = (B_1, B_2, \ldots, B_n)$ be two random vectors, and $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ be two particular realizations of A and B respectively. Assume that the sub-fields induced by A_k, k = 1, 2, …, n are identical and independent. Similarly assume that the sub-fields induced by B_k, k = 1, 2, …, n are also identical and independent. Let $a_{(1)}$, $a_{(2)}$, …, $a_{(n)}$ be the values of a_1, a_2, …, a_n, and $b_{(1)}$, $b_{(2)}$, …, $b_{(n)}$ be the values of b_1, b_2, …, b_n arranged in ascending order.

For A and B, if the empirical probability distribution functions $\phi_1(a)$ and $\phi_2(b)$ are defined as in (5) and (6) respectively. Then, the Glivenko-Cantelli Lemma of order statistics states that the mathematical expectation of the empirical probability distributions would be given by the respective theoretical probability distributions.

$$\phi_1(a) = \begin{cases} 0 & a < a_{(1)} \\ (r-1)/n & a_{(r-1)} \leq a \leq a_{(r)} \\ 1 & a > a_{(n)} \end{cases} \quad \ldots \ (5)$$

$$\phi_2(b) = \begin{cases} 0 & b < b_{(1)} \\ (r-1)/n & b_{(r-1)} \le b \le b_{(r)} \\ 1 & b > b_{(n)} \end{cases} \quad \ldots \quad (6)$$

Now, let A_k is random in the interval [x, y] and B_k is random in the interval [y, z] so that $P_1(x, a)$ and $P_2(y, b)$ are the probability distribution functions followed by A_k and B_k respectively. Then in this case Glivenko-Cantelli Lemma gives

$$\left.\begin{array}{l} U[\phi_1(a)] = P_1(x,a),\ x \le a \le y,\ and \\ U[\phi_2(b)] = P_1(y,b),\ y \le b \le z \end{array}\right\} \ldots (7)$$

It can be observed that in equation (4) the membership values of $[a_{(r)}, a_{(r+1)}]^{(r/n)}$, r = 1, 2, …, n-1 look like empirical probability distribution function $P_1(a)$ and the membership values of $[b_{(n-r)}, b_{(n-r+1)}]^{(r/n)}$, r=1,2,….,$n$-1 look like the values of empirical complementary probability distribution function or empirical survival function [1- $P_2(b)$].

Therefore, if X(a) is the membership function of an L-R fuzzy number X=[x, y, z]. It gives from (ix)

$$X(a) = \begin{cases} P_1(x,a), & x \le a \le y \\ 1 - P_2(y,a), & y \le a \le z \end{cases} \quad .. \quad (8)$$

Thus it can be seen that P_1(a) can indeed be the *Dubois-Prade* left reference function and (1 - P_2(a)) can be the *Dubois-Prade* right reference function [19]**.** *Baruah* [18] has shown that if a possibility distribution is viewed in this way, two probability laws can, indeed, give rise to a possibility law.

4. ALGORITHM

Suppose the time-stamps stored in the transactions of temporal data are the time hierarchy of the type *second_minute_hour_day_week_month_year,* then it does not consider *year, month, week* in time hierarchy and only consider *day,hour,second*. Using the method discussed in [3], it extract frequent itemsets. Each frequent itemset will have a sequence of time intervals of the type [*day1, day2*] associated with it where it is frequent. The sequence of time intervals are used to find the set of *superimposed* intervals [Definition of *superimposed* intervals is given in section-III] and each *superimposed* intervals will be a fuzzy intervals. The algorithm is similar to that of [6, 7]. The method is as follows: The set of *superimposed* intervals is initially empty for a frequent itemset, Each interval associated with the frequent itemset is sequentially visited by the algorithm, If the *core* of any existing superimposed intervals is intersected by an interval [Definition of *core* is given in section-III] in the set it will be *superimposed* on it and membership values will be adjusted else a new *superimposed* intervals will be started with the this interval. This process continues till the end of the sequence of time intervals. The process is repeated for all the frequent itemsets. Finally each frequent itemsets will have one or more *superimposed* time intervals. As the *superimposed* time intervals are used to generate fuzzy intervals, each frequent itemset will be associated with one or more fuzzy time intervals where it is frequent. Each *superimposed* intervals is represented in a compact manner discussed in section-III.

For representing each *superimposed* interval of the form

$$[i^{(1)}, i^{(2)}]^{1/n}[i^{(2)}, i^{(3)}]^{2/n}[i^{(3)}, i^{(4)}]^{3/n}\ldots$$

$$[i^{(r)}, i^{(r+1)}]^{r/n}\ldots\ldots\ldots\ldots$$

$$[i^{(n)}, i^{'(1)}]^{1}[i^{'(1)}, i^{'(2)}]^{\frac{n-1}{n}}\ldots\ldots\ldots[i^{'(n-2)},$$

$$i^{'(n-1)}]^{\frac{2}{n}}[i^{'(n-1)}, i^{'(n)}]^{1/n}$$

Let’s take two arrays of real numbers, one for storing the values $i^{(1)}$, $i^{(2)}$, $i^{(3)}$,….$^{(n)}$ and the other for storing the values $i^{'(1)}$, $i^{'(2)}$,…. $i^{'(n)}$ each of which is a sorted array. Now if a new interval $[i, i^{'}]$ is to be *superimposed* on this interval it adds t to the first array by finding its position (using binary search) in the first array so that it remains sorted. Similarly $i^{'}$ is added to the second array.

Data structure used for representing a *superimposed* interval is

```
            struct superinterval
                                   { int arysize, count;
                                     short *p, *q;
                                   }
```

Here *arysize* represents the maximum size of the array used, *count* represents the number of intervals *superimposed*, and *p* and *q* are two pointer pointing to the two associated arrays.

Algorithm 4.1

```
 for each locally frequent item set s do
 {T← sequence of time intervals associated with s
   Ts ←set of superimposed intervals initially set to null
   pi = T.get();
                // ‘pi’ is now pointing to the first interval in T

               Ts.append(pi);

             Do while ((pi = T.get()) != null)

                        {flag = 0;

            Do while ((psi = Ts.get()) != null)
                   if(compsuperimp(pi, psi))
                          flag = 1;
                 if (flag == 0) Ts.append(pi);
             }
  }

compsuperimp(pi, psi)
{ if(│intersect(psi, pi)│!= null)
             { superimp(pi, psi);
                return 1;
             }
   return 0;
}
```

The function *compsuperimp(pi, psi)* first computes the intersection of *pi* with the *core* of *psi*. If the intersection nonempty it superimposes *pi* by calling the function *superimp(pi,psi)* which actually carries on the *superimposition* process by updating the two lists associated as described earlier. The function returns 1 if *pi* has been *superimposed* on the *psi* otherwise returns 0. *get* and *append* are functions operating on lists to get a pointer to the next element in a list and to append an element into a list.

5. RESULTS OBTAINED

For experimental purpose, a synthetic dataset T10I4D100K is used, available from FIMI[1] website. The number of items here are 942 with the transactions involved are 100000. The minimum number of items in a transaction is 4. The maximum number of items in a transaction is 77. The average number of items in a transaction is 39. The results obtained are presented in table 1 and figure 1.

Table 1. Weekly fuzzy frequent itemsets for different set of transactions for itemset {5}

Data Size (No of Transactions)	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000
No. fuzzy time intervals	3	4	4	4	5	5	5	5	5	5

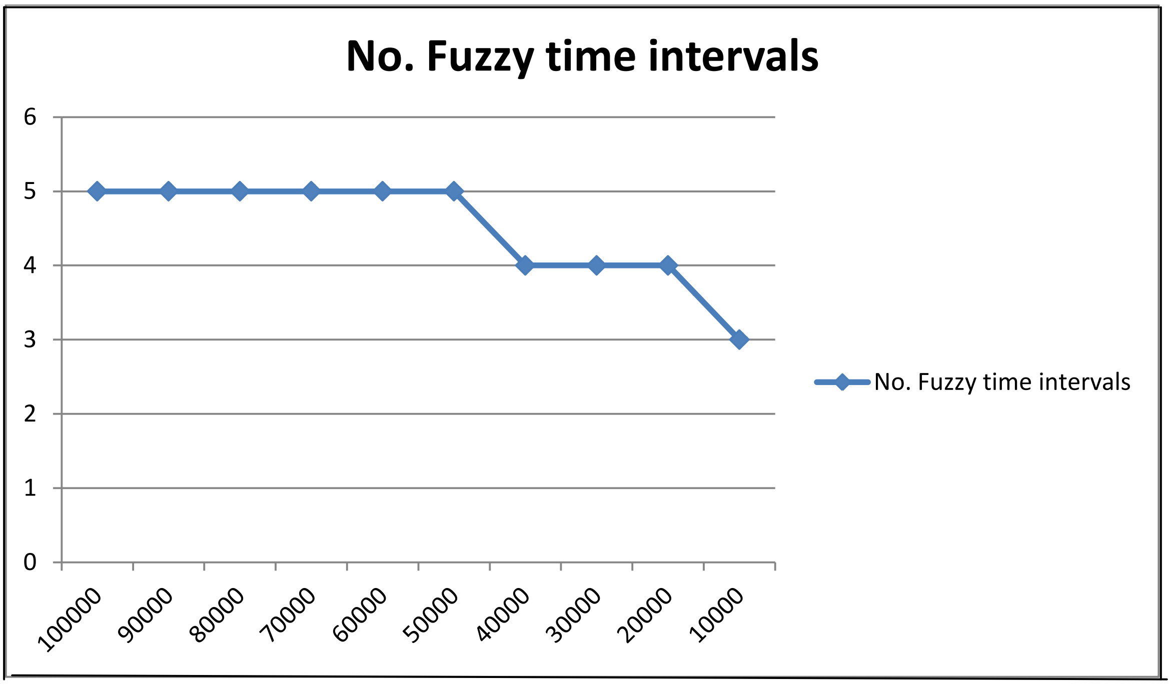

Figure 1. The weekly fuzzy frequent itemsets for different set of transactions for itemset {5}

Since dataset is non-temporal, it incorporates temporal features into it. Here it keeps the life-span of the datasets as 1 year. Firstly, it takes only 10,000 transactions and found that the itemset {5} has a superimposed intervals superimposed on one place and hence it has one fuzzy time interval where it is frequent. For 20,000 and 30,000 transactions the same itemset has two superimposed intervals and so two fuzzy intervals, finally from 60,000-100,000 transactions, it gets {5} is frequent in four fuzzy time intervals.

6. CONCLUSION AND LINES FOR FUTURE WORK

This paper discusses a method for extracting weekly fuzzy patterns. The input is a list of time intervals associated with a frequent itemset generated using a method discussed [4]. Here it does not consider the *year, month, week* in the time hierarchy and only consider day,*minute, second*. The sequence of time intervals of the form [day_minute_second, day_minute_second] will be associated with each frequent itemset where it is frequent. Each interval in the sequence is visited by the algorithm one by one and stores the intervals in the *superimposed* form. In this way each frequent itemset is associated with one or more *superimposed* time intervals. Each *superimposed* interval delivers fuzzy time intervals. So, each frequent itemset is associated with one or more fuzzy time intervals. The best thing about the method is that the algorithm is not user-dependent i.e. fuzzy time intervals are extracted by algorithm automatically. Future work may be on the extraction of fuzzy patterns namely quarterly, half yearly patterns and bi-yearly.

7. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami; Mining association rules between sets of items in large databases, *In Proc. of 1993 ACM SIGMOD Int'l Conf on Management of Data*, Vol. 22(2) of SIGMOD Records, ACM Press, (1993), pp 207-216.

[2] J. M. Ale, and G. H. Rossi; An Approach to Discovering Temporal Association Rules, *In Proc. of 2000 ACM symposium on Applied Computing* (2000).

[3] A. K. Mahanta, F. A. Mazarbhuiya, and H. K. Baruah; Finding Locally and Periodically Frequent Sets and Periodic Association Rules, *In Proc. of 1st Int'l Conf. on Pattern Recognition and Machine Intelligence, LNCS* 3776 (2005), pp. 576-582.

[4] A. K. Mahanta, F. A. Mazarbhuiya, and H. K. Baruah (2008). Finding Calendar-based Periodic Patterns, *Pattern Recognition Letters, Vol. 29(9), Elsevier publication, USA*, pp. 1274-1284.

[5] H. K. Baruah (2010); The Randomness-Fuzziness consistency principle, International Journal of Energy, Information and Communications, Vol 1(1), Nov 2010, Japan.

[6] F. A. Mazarbhuiya (2014); Discovering Yearly Fuzzy Patterns, International Journal of Computer Science and Information security (IJCSIS) *Vol. 12, No. 9, September 2014.*

[7] M. Shenify and F. A. Mazarbhuiya (2015); Discovering Monthly Fuzzy Patterns, International Journal of Intelligence Science (IJIS), 37-43, USA.

[8] F. A. Mazarbhuiya (2015); Extracting daily fuzzy patterns, International Journal of Computer Science and Information security (IJCSIS) *Vol. 44, No. 1, January 2016* .

[9] F. A. Mazarbhuiya and Yusuf Perwej (2015); Mining Hourly Fuzzy Patterns from Temporal Datasets**,** International Journal of Engineering Research and Technology (IJERT), vol 4, issue 10, October 2015.

[10] C. M. Antunes, and A. L. Oliviera; Temporal Data Mining an overview, *Workshop on Temporal Data Mining-7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, (2001).

[11] B. Ozden, S. Ramaswamy, and A. Silberschatz; Cyclic Association Rules, *In Proc. of the 14th Int'l Conf. on Data Engineering*, USA (1998), pp. 412-421.

[12] Y. Li, P. Ning, X. S. Wang, and S. Jajodia; Discovering Calendar-based Temporal Association Rules, *Elsevier Science*, (2001).

[13] G. Zimbrado, J. Moreira de Souza, V. Teixeira de Almeida, and W. Arauja de Silva; An Algorithm to Discover Calendar-based Temporal Association Rules with Item's Lifespan Restriction, *In Proc. of the 8th ACM SIGKDD* 2002.

[14] R.B.V. Subramanyam, A. Goswami, Bhanu Prasad; Mining fuzzy temporal patterns from process instances with weighted temporal graphsInt. J. of Data Analysis Techniques and Strategies, 2008 Vol.1, No.1, pp.60 – 77.

[15] S. Jain, S. Jain, and A. Jain; An assessment of Fuzzy Temporal Rule Mining, International Journal of Application or Innovation in Engineering and Management (IJAIEM), Vol. 2, 1, January 2013, pp. 42-45.

[16] Wan-Ju Lee, Jung-Yi Jiang and Shie-Jue Lee; Mining fuzzy periodic association rules, Data & Knowledge Engineering, Vol. 65, Issue 3, June 2008, pp. 442-462.

[17] Klir, J. and Yuan, B.; Fuzzy Sets and Logic Theory and Application, *Prentice Hill Pvt. Ltd*. (2002).

[18] H. K. Baruah; Set Superimposition and its application to the Theory of Fuzzy Sets, *Journal of Assam Science Society*, Vol. 10 No. 1 and 2, (1999), pp. 25-31.

[19] D. Dubois and H. Prade; Ranking fuzzy numbers in the setting of possibility theory, *Inf. Sc.*30, (1983), pp. 183-224.