

The Mining Hourly Fuzzy Patterns from Temporal Datasets

Dr. F. A. Mazarbhuiya
Assistant Professor,
Department of Computer Science & IT
Al Baha University, Al Baha,
Kingdom of Saudi Arabia (KSA)

Dr. Yusuf Perwej
Assistant Professor,
Department of Computer Science & Engg.
Al Baha University, Al Baha,
Kingdom of Saudi Arabia (KSA)

Abstract: The retrace patterns from supermarket datasets is an interesting data mining issue. The hourly fuzzy pattern is an example of such patterns where the pattern holds in some fuzzy time interval in each hour. The issue involves first frequent set mining and then deduces association rules from the frequent sets. We call it hourly fuzzy patterns. In the beginning works were mainly concentrated about a user-specified fuzzy pattern mining. In as much as, in some applications, the user may not have previous knowledge about the datasets under opinion, so may not be able to specify fuzzy time interval. Identical may happen due to the boundary of natural language. In this paper, we are intending a method of finding such patterns. The nicety about the method is that the fuzzy time intervals are generated by the method itself. The impact of our method is exhibited with experimental results.

Keywords: Temporal Patterns, Temporal Association Rules, Superimposed Intervals, Fuzzy Set, Right Reference Functions, Left Reference Functions, Membership Functions.

I. INTRODUCTION

The analysis of temporal data has been pondered as an important data mining problem. An example of such is super market dataset mining. In real world supermarket data if a transaction happens its time also recorded. In [1], the authors have addressed such problems in details. In their work, the attribute associated with the dataset has been pondered like any other attribute. The taking into account time attribute aside, pattern can be extracted which are time-dependent and cannot be extracted by [1]. *Ale et al* [2], first studied such problems in detail. They proposed a method of discovering association rules that hold within the life-span an itemset. In this spot life-span of an itemset is the time period between first transactions, including the itemset to the last transaction inclusive the equal in the dataset. The life-span of an itemset is not necessary equal as that of dataset.

The idea of local Patterns has been proposed by *Anjana et al* [3], which are patterns that are frequently in certain time intervals and may or may not be frequent throughout the life-span of the itemset. They proposed an algorithm which can extract locally frequent itemsets along with a list of sequences of time intervals where each locally frequent itemsets is frequent in a sequential time intervals. The considering the time-stamp in the form *year_month_day* i.e. time hierarchy a method is discussed in [4] which can extract *yearly*, *monthly*, periodic or partially periodic

patterns. In [4] a method called set superimposition is used to keep the intervals of the frequency with overlapping. The superimposed intervals turn out to be fuzzy intervals associated with each locally frequent itemsets. Considering time stamp *year_month_day_hour_minute_second*, methods were proposed in [6, 7, 8], for finding yearly, monthly, and daily fuzzy frequent itemsets. Unless if only *minute_second* are pondered in time-hierarchy, then there may exist some patterns which are hourly fuzzy frequent. In this paper, we have discussed hourly fuzzy patterns and devised an algorithm for finding such patterns. Talk about this algorithm similar to that in [6, 7, 8]. The paper is organized as follows. In section-2, we discuss related works. In section-3, we discuss terms, definitions and notations used in the algorithm. In section-4, the algorithm is discussed. In section-5, we discuss about results and analysis. Finally a summary and lines for future works are discussed in section-6.

II. RELATED WORKS

The association rules discovery problem was first proposed by *Agrawal et al* [1]. In their work, if I , is the set of items and D , a large collection of transactions involving the items, the question is to find kinship among the presence of various items in the transactions. The temporal data mining [9] is an important extension of conventional data mining. If time aspect is taken into account, then more interesting patterns that are time dependent can be extracted. The process of association rule discovery is also extended to incorporate temporal aspects. The corresponding problems are to find valid time periods during which association rules hold and the discovery of possible periodicities that association rules have. This author proposed an algorithm for the discovery of temporal association rules [2]. 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. Next itemsets are calculated only during its life-span of itemset. After that each rule has associated with it a time-frame. In [3], the works done in [2] have been extended by considering time gap between two consecutive transactions containing an item set into account. The acceptance the periodicity of patterns into rumination, *Ozden* [10] proposed a method, which is able to find user-specified periodic patterns. In [11], the authors discussed about a

method of discovering temporal association rules with respect to fuzzy match, i.e. association rule holding during “enough” number of intervals given by the corresponding calendar pattern.

In common with works were done in [12] incorporating multiple granularities of time intervals (e.g. first working day of every month) from which both cyclic and user defined calendar patterns can be cognizable. Mining fuzzy patterns from datasets have been studied by many researchers. In [13], the authors presented a method for mining fuzzy temporal patterns from a given process instance. In common with work is done in [14]. Next the method of extracting fuzzy periodic association [15] rules is discussed. In [6], authors have presented a method of finding yearly fuzzy patterns. In [7], the works were done in mining monthly fuzzy patterns. In [8], methods were discussed for finding daily fuzzy patterns.

III. TERMS, DEFINITIONS AND NOTATIONS

Let us scrutiny some definitions and notations used in this paper. Let E be the universe of discourse. A fuzzy set A in E is characterized by a membership function $A(x)$ lying in $[0, 1]$.

$A(x)$ for $x \in E$ represents the grade of membership of x in A . Thus a fuzzy set A is defined as

$$A = \{ (x, A(x)), x \in E \}$$

A fuzzy set A is said to be normal if $A(x) = 1$ for at least one $x \in E$. An α -cut of a fuzzy set is an ordinary set of elements with membership grade greater than or equal to a threshold α , $0 \leq \alpha \leq 1$. Thus an α -cut A_α of a fuzzy set A is characterized by $A_\alpha = \{ x \in E; A(x) \geq \alpha \}$ [see e.g. [16]]

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 A defined on the real line R such that

1. there exists an $x_0 \in R$ such that $A(x_0) = 1$, and
2. $A(x)$ 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 $[a, b] \subset R$ such that $A(x_0) = 1$ for all $x_0 \in [a, b]$, and
2. $A(x)$ is piecewise continuous.

A fuzzy interval can be thought of as a fuzzy number with a flat region. A fuzzy interval A is denoted by $A = [a, b, c, d]$ with $a < b < c < d$ where $A(a) = A(d) = 0$ and $A(x) = 1$ for all $x \in [b, c]$. $A(x)$ for all $x \in [a, b]$ is known as *left reference function* and $A(x)$ for $x \in [c, d]$ 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 A within a universal set E is the crisp set that contains all the elements of E that have non-zero membership grades in A and is notified by $S(A)$. Thus

$$S(A) = \{ x \in E; A(x) > 0 \}$$

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

Set Superimposition

In [17] an operation called superimposition (S) was proposed. If A is superimposed over B or B is superimposed over A , we have

$$A(S)B = (A-B) (+) (A \cap B)^{(2)} (+) (B-A) \quad \dots (1)$$

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

To explain this, an example has been taken.

If $A = [a_1, b_1]$ and $B = [a_2, b_2]$ are two real intervals such that $A \cap B \neq \emptyset$, we would get a superimposed portion. It can be seen from (1)

$$[a_1, b_1](S)[a_2, b_2] = [a_{(1)}, a_{(2)}]^{(2)} (+) [a_{(2)}, b_{(1)}]^{(2)} (+) (b_{(1)}, b_{(2)}) \quad \dots (2)$$

Where $a_{(1)} = \min(a_1, a_2)$ $a_{(2)} = \max(a_1, a_2)$

$b_{(1)} = \min(b_1, b_2)$, and $b_{(2)} = \max(b_1, b_2)$

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

Let now, $[a_1, b_1]^{(1/2)}$ and $[a_2, b_2]^{(1/2)}$ be two fuzzy sets with constant membership value $1/2$ everywhere (i.e. equi-fuzzy intervals with membership value $1/2$). If $[a_1, b_1] \cap [a_2, b_2] \neq \emptyset$ then applying (2) on the two equi-fuzzy intervals we can write

$$[a_1, b_1]^{(1/2)}(S)[a_2, b_2]^{(1/2)} = [a_{(1)}, a_{(2)}]^{(1/2)} (+) [a_{(2)}, b_{(1)}]^{(1/2)} (+) (b_{(1)}, b_{(2)})^{(1/2)} \quad \dots (3)$$

Let $[x_i, y_i]$, $i=1, 2, \dots, n$, be n real intervals such that

$\bigcap_{i=1}^n [x_i, y_i] \neq \emptyset$. Generalizing (3) we get

$$[x_1, y_1]^{(1/n)}(S)[x_2, y_2]^{(1/n)}(S) \dots (S) [x_n, y_n]^{(1/n)} = [x_{(1)}, x_{(2)}]^{(1/n)} (+) [x_{(2)}, x_{(3)}]^{(2/n)} (+) \dots (+) [x_{(r)}, x_{(r+1)}]^{(r/n)} (+) \dots (+) [x_{(n)}, y_{(1)}]^{(1)} (+) (y_{(1)}, y_{(2)})^{((n-1)/n)} (+) \dots (+) (y_{(n-r)}, y_{(n-r+1)})^{(r/n)} (+) \dots (+) (y_{(n-2)}, y_{(n-1)})^{(2/n)} (+) (y_{(n-1)}, y_{(n)})^{(1/n)} \quad \dots (4)$$

In (4), the sequence $\{x_{(i)}\}$ is formed by sorting the sequence $\{x_i\}$ in ascending order of magnitude for $i=1, 2, \dots, n$ and similarly $\{y_{(i)}\}$ is formed by sorting the sequence $\{y_i\}$ in ascending order. In as much as 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.

LEMMA 1. (THE GLIVENKO-CANTELLI LEMMA OF ORDER STATISTICS)

Let $X = (X_1, X_2, \dots, X_n)$ and $Y = (Y_1, Y_2, \dots, Y_n)$ be two random vectors, and (x_1, x_2, \dots, x_n) and (y_1, y_2, \dots, y_n) be two distinctive realizations of X and Y respectively. Assume that the sub-fields induced by X_k , $k = 1, 2, \dots, n$ are

IV. ALGORITHM

identical and independent. Accordingly assume that the sub- fields induced by Y_k , $k = 1, 2, \dots, n$ are also identical and independent. Let $x_{(1)}, x_{(2)}, \dots, x_{(n)}$ be the values of x_1, x_2, \dots, x_n , and $y_{(1)}, y_{(2)}, \dots, y_{(n)}$ be the values of y_1, y_2, \dots, y_n arranged in ascending order.

For X and Y if the experiential probability distribution functions $\phi_1(x)$ and $\phi_2(y)$ 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.

$$\phi_1(x) = \begin{cases} 0 & x < x_{(1)} \\ (r-1)/n & x_{(r-1)} \leq x \leq x_{(r)} \\ \dots & (5) \\ 1 & x > x_{(n)} \end{cases}$$

$$\phi_2(y) = \begin{cases} 0 & y < y_{(1)} \\ (r-1)/n & y_{(r-1)} \leq y \leq y_{(r)} \\ \dots & (6) \\ 1 & y > y_{(n)} \end{cases}$$

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

$$\left. \begin{aligned} E[\phi_1(x)] &= P_1(a, x), a \leq x \leq b, \text{ and} \\ E[\phi_2(y)] &= P_1(b, y), b \leq y \leq c \end{aligned} \right\} \dots (7)$$

It can be take a notion that in equation (4) the membership values of $[x_{(r)}, x_{(r+1)}]^{(r/n)}$, $r = 1, 2, \dots, n-1$ look like experiential probability distribution function $P_1(x)$ and the membership values of $[y_{(n-r)}, y_{(n-r+1)}]^{(r/n)}$, $r=1,2,\dots,n-1$ look like the values of experiential complementary probability distribution function or experiential survival function $[1 - P_2(y)]$.

Therefore, if $A(x)$ is the membership function of an L-R fuzzy number $A=[a, b, c]$. We get from (4)

$$A(x) = \begin{cases} P_1(a, x), & a \leq x \leq b \\ 1 - P_2(b, x), & b \leq x \leq c \end{cases} \quad (8)$$

It remains to be seen $P_1(x)$ can indeed be the *Dubois-Prade left reference function* and $(1 - P_2(x))$ can be the *Dubois-Prade right reference function* [19]. *Baruah* [17] has shown that if a possibility distribution is viewed in this way, two probability laws can, indeed, give rise to a possibility law.

In our proposed algorithm the time-stamps stored in the transactions of temporal data are the time hierarchy of the type *second_minute_hour_day_month_year*, then we do not contemplate *year, month, day, and hour* in time hierarchy and only contemplate *minute* and *second*. Now we extract frequent itemsets using method discussed in [3]. Each frequent itemset will have a sequence of time intervals of the type $[min_second, min_second]$ associated with it where it is frequent [18]. Using the sequence of time intervals, we can discover the set of superimposed intervals [Definition of superimposed intervals is given in section-3] and each superimposed intervals will be a fuzzy intervals. The algorithm is alike that of [6, 7, 8]. The method is as follows for a frequent itemset the set of superimposed intervals is initially empty, algorithm visits each interval associated with the frequent itemset sequentially, if an interval is intersecting with the core of any existing superimposed intervals [Definition of core is given in section-3] in the set it will be superimposed on it and membership values will be adjusted otherwise a new superimposed intervals will be started with the this interval. This procedure will be continued till the end of the sequence of time intervals. The proposed procedure will be repeated for all the frequent itemsets as well as possible each frequent itemsets will have one or more superimposed time interval. Additionally 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. In addition to this each superimposed intervals is represented in a compact manner discussed in section-3.

For representing each superimposed interval of the form $[t^{(1)}, t^{(2)}]^{1/n} [t^{(2)}, t^{(3)}]^{2/n} [t^{(3)}, t^{(4)}]^{3/n} \dots [t^{(r)}, t^{(r+1)}]^{r/n} \dots$

$$[t^{(n)}, t^{(1)}]^{1/n} [t^{(1)}, t^{(2)}]^{n-1} \dots [t^{(n-2)}, t^{(n-1)}]^{2/n} [t^{(n-1)}, t^{(n)}]^{1/n}$$

In this algorithm, we lay two arrays of real numbers, one for storing the values $t^{(1)}, t^{(2)}, t^{(3)}, \dots, t^{(n)}$ and the other for storing the values $t^{(1)}, t^{(2)}, \dots, t^{(n)}$ each of which is a sorted array. At the moment if a new interval $[t, t']$ is to be superimposed on this interval we add t to the first array by finding its position (using binary search) in the first array so that it remnant sorted. In like manner t' is added to the second array.

Data structure used for representing a superimposed interval is

```
structsuperinterval
{
    intarsize, count;
    short *l, *r;
}
```

Here *arsize* represents the maximum size of the array used, *count* represents the number of intervals superimposed, and *l* and *r* is two pointers pointing to the two associated arrays.

Algorithm 4.1

```

for each locally frequent item set s do
    {L← sequence of time intervals associated with s
    Ls← set of superimposed intervals initially set to null
    lt = L.get();
    // lt is now pointing to the first interval in L
    Ls.append(lt);
    While ((lt = L.get()) != null)
        {flag = 0;
    While ((lst = Ls.get()) != null)
        if(compsuperimp(lt, lst))
            flag = 1;
        if (flag == 0) Ls.append(lt);
        }
    }
compsuperimp(lt, lst)
{ if( | intersect(lst, lt) | != null)
    { superimp(lt, lst);
    return 1;
    }
return 0;
}
    
```

The correspondingly function `compsuperimp(lt, lst)` first computes the intersection of `lt` with the core of `lst`. If the intersection non-empty it superimposes `lt` by calling the function `superimp(lt, lst)` which virtually carries on the superimposition process by updating the two lists associated as described earlier. The function return 1 if `lt` has been superimposed on the `lst` else returns 0. Obtain and tag on are functions operating on lists to get a pointer to the next element in a list and to tag on an element into a list.

V. RESULTS PROCURED

In this paper experimental result purpose, we have used a synthetic dataset T10I4D100K, available from FIMI¹ website. A summarized view of the dataset is presented in table 1. The dataset mentioned in table 1 and some procured results are presented in table 2.

<http://fimi.cs.helsinki.fi/data/>

Table 1. T10I4D100K dataset characteristics

Dataset	#Items	#Transactions	Min T	max T	Avg T
T10I4D100K	942	100 000	4	77	39

Table 2. The daily 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	2	3	3	3	4	4	4	4	4	4

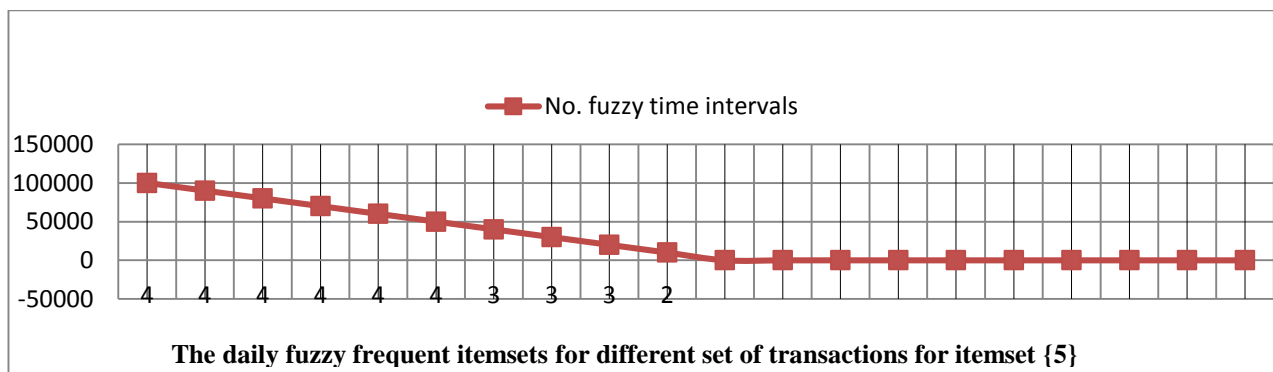


Figure 1. The daily fuzzy frequent itemsets for different set of transactions for itemset {5}

So far as dataset is non-temporal, we incorporate temporal features into it. To this spot we keep the life-span of our datasets as 1 year. Firstly, we take 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, in the end from 60,000-100,000 transactions, we get {5} is frequent in four fuzzy time intervals figure 1 shown.

VI. CONCLUSION AND LINE FOR FUTURE WORK

In this paper, we are discussing methods for discovering hourly fuzzy patterns. The method receive as input a list of time intervals associated with a frequent itemset generated using a method discussed [4]. In spite of, our work we do not contemplate the year, month, day and hour at the time hierarchy and only contemplate minute, second. Finally, every frequent itemset will be associated with a sequence of time intervals of the form [minute_second, minute_second] where it is frequent. The algorithm sees each other interval in the sequence one by one and stores the intervals in the superimposed form. Next every frequent itemset is associated with one or more superimposed time intervals. Each superimposed interval gives fuzzy time intervals. In this case we will have each frequent itemset is associated with one or more fuzzy time intervals. The nicety about the method is that the algorithm is fewer user-reliant on i.e. fuzzy time intervals are citation by algorithm automatically. Future work may be possible in the following ways.

- A further type of fuzzy patterns, namely weekly, quarterly, half yearly patterns can be extracted.
- Clustering of patterns can be done based on their fuzzy time interval associated with yearly patterns using some statistical measure.

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. 13, No. 10, October 2015* (communicated).
- [9] C. M. Antunes, and A. L. Oliveira; Temporal Data Mining an overview, *Workshop on Temporal Data Mining-7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, (2001).
- [10] 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.
- [11] Y. Li, P. Ning, X. S. Wang, and S. Jajodia; Discovering Calendar-based Temporal Association Rules, *Elsevier Science*, (2001).
- [12] G. Zimbrado, J. Moreira de Souza, V. Teixeira de Almeida, and W. Araujo de Silva; An Algorithm to Discover Calendar-based Temporal Association Rules with Item's Lifespan Restriction, *In Proc. of the 8th ACM SIGKDD 2002*.
- [13] R.B.V. Subramanyam, A. Goswami, Bhanu Prasad; Mining fuzzy temporal patterns from process instances with weighted temporal graphs *Int. J. of Data Analysis Techniques and Strategies*, 2008 Vol.1, No.1, pp.60 – 77.
- [14] 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.
- [15] 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.
- [16] Klir, J. and Yuan, B.; Fuzzy Sets and Logic Theory and Application, *Prentice Hill Pvt. Ltd.* (2002).
- [17] 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.
- [18] F. A. Mazarbhuiya, Yusuf Perwej, "An Efficient Method for Generating Local Association Rules", *International Journal of Applied Information Systems* 9(2):1-5, June 2015. Published by Foundation of Computer Science, New York, USA, DOI: 10.5120/ijais15-451368
- [19] D. Dubois and H. Prade; Ranking fuzzy numbers in the setting of possibility theory, *Inf. Sc.*30, (1983), pp. 183-224.