International Journal of Intelligence Science, 2015, 5, 37-43
Published Online January 2015 in SciRes. http://www.scirp.org/journal/ijis
http://dx.ID.org/10.4236/ijis.2015.51004

Discovering Monthly Fuzzy Patterns

M. Shenify, F. A. Mazarbhuiya

College of Computer Science and IT, Albaha University, Albaha, Saudi Arabia
Email: mshenify@yahoo.com, fokrul_2005@yahoo.com

Received 20 October 2014; revised 22 November 2014; accepted 18 December 2014

Academic Editor: Zhongzhi Shi, Institute of Computing Technology, CAS, China

Abstract

Discovering patterns that are fuzzy in nature from temporal datasets is an interesting data mining problems. One of such patterns is monthly fuzzy pattern where the patterns exist in a certain fuzzy time interval of every month. It involves finding frequent sets and then association rules that holds in certain fuzzy time intervals, viz. beginning of every months or middle of every months, etc. In most of the earlier works, the fuzziness was user-specified. However, in some applications, users may not have enough prior knowledge about the datasets under consideration and may miss some fuzziness associated with the problem. It may be the case that the user is unable to specify the same due to limitation of natural language. In this article, we propose a method of finding patterns that holds in certain fuzzy time intervals of every month where fuzziness is generated by the method itself. The efficacy of the method is demonstrated with experimental results.

Keywords

Frequent Item Sets, Superimposition of Time Intervals, Fuzzy Time Intervals, Right Reference Functions, Left Reference Functions, Membership Functions

1. Introduction

Analysis of transactional data has been considered as an important data mining problem. Market basket data is an example of such transactional data. In a market-basket data set, each transaction is a collection of items bought by a customer at one time. The concept proposed in [1] is to find the co-occurrence of items in transactions, given minimum support and minimum confidence thresholds. Temporal Association rule mining is an important extension of above-mentioned problem. When an item from super-market is bought by a customer, this is called transaction and its time is automatically recorded. Ale *et al.* [2] have proposed a method of extracting association rules that hold within the life-span of the corresponding item set.

Mahanta *et al.* [3] have introduced concept of locally frequent item sets as item sets that are frequent in certain time intervals and may or may not be frequent throughout the life-span of the item set. An efficient algorithm is developed by them which is used find such item sets along with a list of sequences of time intervals. Considering the time-stamp as calendar dates, a method is discussed in [4] which can extract yearly, monthly and daily periodic or partially periodic patterns. If the periods are kept in a compact manner using the method discussed in [4], it turns out to be a fuzzy time interval. In this paper, we discuss such patterns and device algorithms for extracting such patterns. Although our algorithm works for extracting monthly fuzzy patterns, it can be modified for daily fuzzy periodic patterns. 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 proposed 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.

2. Related Works

Agrawal *et al.* [1] first formulated association rules mining problems. One important extension of this problem is Temporal Data Mining [5] by taking into account the time aspect, more interesting patterns that are time dependent can be extracted. The problems associated are to find valid time periods during which association rules hold and the discovery of possible periodicities that association rules have. In [2], an algorithm for finding temporal rules is described. There each rule has associated with it 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.

Considering the periodic nature of patterns, Ozden *et al.* [6] proposed a method, which is able to find patterns having periodic nature where the period has to be specified by the user. In [7], Li *et al.* discuss about a method of extracting temporal association rules with respect to fuzzy match, *i.e.* association rule holding during "enough" number of intervals given by the corresponding calendar pattern. Similar works were done in [8] 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 achieved.

Mining fuzzy patterns from datasets have been studied by different authors. In [9], the authors present an algorithm for mining fuzzy temporal patterns from a given process instance. Similar work is done in [10]. In [11] method of extracting fuzzy periodic association rules is discussed.

3. Terms, Definitions and Notations Used

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

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 followings:

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 [12].

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 denoted 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

When we overwrite, the overwritten portion looks darker for obvious reason. The set operation union does not explain this phenomenon. After all

$$A \cup B = (A - B) \cup (A \cap B) \cup (B - A)$$

and in $(A \cap B)$ the elements are represented once only.

In [13] an operation called superimposition denoted by (S) was proposed. If A is superimposed over B or B is superimposed over A, we have

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

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 $A \cap B \neq \varnothing$, we would get a superimposed portion. It can be seen from (1)

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

where

$$a_{(1)} = \min(a_1, a_2) \quad a_{(2)} = \max(a_1, a_2)$$
$$b_{(1)} = \min(b_1, b_2) \quad \text{and} \quad b_{(2)} = \max(b_1, b_2)$$

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

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

To explain this we take the fuzzy intervals $[1,5]^{(1/2)}$ and $[3,7]^{(1/2)}$ with constant membership value $(1/2)$ given in **Figure 1** and **Figure 2**. Here $[1,5] \cap [3,7] = [3,5] \neq \varnothing$.

If we apply *superimposition* on the intervals then the *superimposed* interval will be consisting of $[1,3)^{(1/2)}$, $[3,5]^{(1)}$ and $(5,7]^{(1/2)}$. Here the membership of $[3,5]$ is (1) due to double representation and it is shown in **Figure 3**.

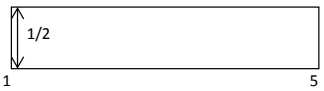

Figure 1. Equi-fuzzy Interval $[1, 5]^{(1/2)}$.

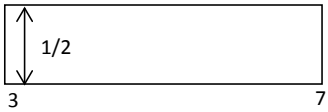

Figure 2. Equi-fuzzy interval $[3, 7]^{(1/2)}$.

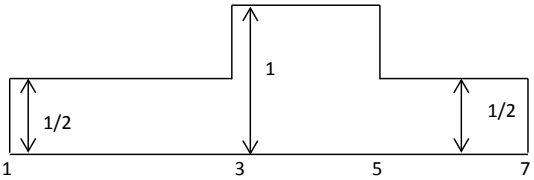

Figure 3. Superimposed interval.

Let $[x_i, y_i]$, $i = 1,2,\cdots,n$, be n real intervals such that $\bigcap_{i=1}^{n}[x_i, y_i] \neq \varnothing$. Generalizing (3) we get

$$\begin{aligned}&[x_1,y_1]^{(1/n)}(S)[x_2,y_2]^{(1/n)}(S)\cdots(S)[x_n,y_n]^{(1/n)}\\&=\left[x_{(1)},x_{(2)}\right)^{(1/n)}(+)\left[x_{(2)},x_{(3)}\right)^{(2/n)}(+)\cdots(+)\left[x_{(r)},x_{(r+1)}\right)^{(r/n)}(+)\cdots\\&\quad(+)\left[x_{(n)},y_{(1)}\right]^{(1)}(+)\left(y_{(1)}y_{(2)}\right]^{((n-1)/n)}\\&\quad(+)\cdots(+)\left(y_{(n-r)},y_{(n-r+1)}\right]^{(r/n)}(+)\cdots(+)\left(y_{(n-2)},y_{(n-1)}\right]^{(2/n)}(+)\left(y_{(n-1)},y_{(n)}\right]^{(1/n)}.\end{aligned}\tag{4}$$

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

Lemma 1. The Glivenko-Cantelli Lemma of Order Statistics

Let $X = (X_1, X_2, \cdots, X_n)$ and $Y = (Y_1, Y_2, \cdots, Y_n)$ be two random vectors, and $(x_1, x_2, \cdots, x_n)$ and $(y_1, y_2, \cdots, y_n)$ be two particular realizations of X and Y respectively. Assume that the sub-σ fields induced by X_k, $k = 1,2,\cdots,n$ are identical and independent. Similarly assume that the sub-σ fields induced by Y_k, $k = 1,2,\cdots,n$ are also identical and independent. Let $x_{(1)}, x_{(2)}, \cdots, x_{(n)}$ be the values of $x_1, x_2, \cdots, x_n$, and $y_{(1)}, y_{(2)}, \cdots, y_{(n)}$ be the values of $y_1, y_2, \cdots, y_n$ arranged in ascending order.

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

$$\phi_1(x) = \begin{cases} 0 & x < x_{(1)} \\ (r-1)/n & x_{(r-1)} \le x \le x_{(r)} \\ 1 & x \ge x_{(n)} \end{cases}\tag{5}$$

$$\phi_2(x) = \begin{cases} 0 & y < y_{(1)} \\ (r-1)/n & y_{(r-1)} \le y \le y_{(r)} \\ 1 & y \ge y_{(n)} \end{cases}\tag{6}$$

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

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

It can be observed that in Equation (4) the membership values of $\left[x_{(r)}, x_{(r+1)}\right]^{(r/n)}$, $r = 1,2,\cdots,n-1$ look like empirical probability distribution function $\phi_1(x)$ and the membership values of $\left[y_{(n-r)}, y_{(n-r+1)}\right]^{(r/n)}$, $r = 1,2,\cdots,n-1$ look like the values of empirical complementary probability distribution function or empirical

survival function $\left[1-\phi_2(y)\right]$.

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

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

Thus it can be seen that $P_1(x)$ can indeed be the Dubois-Prade left reference function and $\left(1-P_2(x)\right)$ can be the Dubois-Prade right reference function [13]. Baruah [14] 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 Proposed

If the time-stamps stored in the transactions of temporal data are the time hierarchy of the type *hour_day_month_year*, then we do not consider *month_year* in time hierarchy and only consider day. We extract frequent item sets using method discussed in [3]. Each frequent item set will have a sequence of time intervals of the type (day 1, day 2) associated with it where it is frequent. Using the sequence of time intervals we can find the set of superimposed intervals (Definition of superimposed intervals is given in Section 3) and each superimposed intervals will be a fuzzy intervals. The method is as follows: for a frequent item set the set of superimposed intervals is initially empty, algorithm visits each intervals associated with the frequent item set 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 else a new superimposed intervals will be started with the this interval. This process will be continued till the end of the sequence of time intervals. The process will be repeated for all the frequent item sets. Finally each frequent item sets will have one or more superimposed time intervals. As the superimposed time intervals are used to generate fuzzy intervals, each frequent item set 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 3.

For representing each *superimposed* interval of the form

$$\left[t^{(1)},t^{(2)}\right]^{1/n}\left[t^{(2)},t^{(3)}\right]^{2/n}\left[t^{(3)},t^{(4)}\right]^{3/n}\cdots\left[t^{(r)},t^{(r+1)}\right]^{r/n}\cdots$$

$$\left[t^{(n)},t'^{(1)}\right]^{1}\left[t'^{(1)},t'^{(2)}\right]^{(n-1)/n}\cdots\left[t'^{(n-2)},t'^{(n-1)}\right]^{2/n}\left[t'^{(n-1)},t'^{(n)}\right]^{1/n}$$

we keep two arrays of real numbers, one for storing the values $t^{(1)},t^{(2)},t^{(3)},\cdots,t^{(n)}$ and the other for storing the values $t'^{(1)},t'^{(2)},t'^{(3)},\cdots,t'^{(n)}$ each of which is a sorted array. Now 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 remains sorted. Similarly t' is added to the second array.

Data structure used for representing a *superimposed* interval is

```
struct superinterval
        { int arsize, 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 are two pointer pointing to the two associated arrays.

Algorithm 4.1

```
for each locally frequent item sets 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 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 actually carries on the superimposition process by updating the two lists associated as described earlier. The function returns 1 if *lt* has been superimposed on the *lst* 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 experimentation purpose we have used retail market basket dataset from an anonymous Belgian retail store. The dataset contains 88,162 transactions and 17,000 items. This dataset does not have attribute, so time was incorporated on it. The domain of the time attribute was set to the calendar dates from 1-1-2001 to 30-2-2003. For the said purpose, a program was written using C++ which takes as input a starting date and two values for the minimum and maximum number of transactions per day. A number between these two limits are selected at random and that many consecutive transactions are marked with the same date so that many transactions have taken place on that day. This process starts from the first transaction to the end by marking the transactions with consecutive dates (assuming that the market remains open on all week days). This means that the transactions in the dataset are happened in between the specified dates. A partial view of the generated monthly fuzzy frequent item sets from retail dataset is shown in **Table 1**.

6. Conclusions and Lines for Future Work

An algorithm for finding monthly fuzzy patterns is discussed in this paper. The method takes input as a list of time intervals associated with a frequent item set. The frequent item set is generated using a method similar to the method discussed [4]. However, in our work we do not consider the *month_year* in the time hierarchy and only consider day. So each frequent item set will be associated with a sequence of time intervals of the form (day 1, day 2) where it is frequent. The algorithm visits each interval in the sequence one by one and stores the intervals in the superimposed form. This way each frequent item set is associated with one or more superimposed time intervals. Each superimposed interval will generate a fuzzy time interval. In this way each frequent item set is associated with one or more fuzzy time intervals. The nicety about the method is that the algorithm is less user-dependent, *i.e.* fuzzy time intervals are extracted by algorithm automatically.

Future work may be possible in the following ways.

- Daily patterns can be extracted.

Table 1. Monthly fuzzy frequent item sets for different set of transactions.

Data Size (No. of Transactions)	10000	20000	30000	40000	50000	60000	70000	Whole Dataset
No. fuzzy time intervals	1	2	2	3	3	4	4	4

- Clustering of patterns can be done based on their fuzzy time interval associated with yearly patterns using some statistical measure.

References

[1] Agrawal, R., Imielinski, T. and Swami, A.N. (1993) Mining Association Rules between Sets of Items in Large Databases. *Proceedings of the* 1993 *ACM SIGMOD International Conference on Management of Data*, **22**, 207-216. http://dx.doi.org/10.1145/170035.170072

[2] Ale, J.M. and Rossi, G.H. (2000) An Approach to Discovering Temporal Association Rules. *Proceedings of* 2000 *ACM Symposium on Applied Computing*, Como, 19-21 March 2000, 294-300.

[3] Mahanta, A.K., Mazarbhuiya, F.A. and Baruah, H.K. (2005) Finding Locally and Periodically Frequent Sets and Periodic Association Rules. *Pattern Recognition and Machine Intelligence*, **3776**, 576-582.

[4] Mahanta, A.K., Mazarbhuiya, F.A. and Baruah, H.K. (2008) Finding Calendar-Based Periodic Patterns. *Pattern Recognition Letters*, **29**, 1274-1284.

[5] Antunes, C.M. and Oliviera, A.L. (2001) Temporal Data Mining: An Overview. *Workshop on Temporal Data Mining—7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, 26-29 August 2001, 1-13.

[6] Ozden, B., Ramaswamy, S. and Silberschatz, A. (1998) Cyclic Association Rules. *Proceedings of the* 14*th International Conference on Data Engineering*, Orlando, 23-27 February 1998, 412-421. http://dx.doi.org/10.1109/ICDE.1998.655804

[7] Li, Y., Ning, P., Wang, X.S. and Jajodia, S. (2001) Discovering Calendar-Based Temporal Association Rules. Elsevier Science, Amsterdam.

[8] Zimbrado, G., de Souza, J.M., de Almeida, V.T. and de Silva, W.A. (2002) An Algorithm to Discover Calendar-Based Temporal Association Rules with Item's Lifespan Restriction. *Proceedings of the* 8*th ACM SIGKDD*, Alberta, 23 July 2002.

[9] Subramanyam, R.B.V., Goswami, A. and Prasad, B. (2008) Mining Fuzzy Temporal Patterns from Process Instances with Weighted Temporal Graphs. *International Journal of Data Analysis Techniques and Strategies*, **1**, 60-77. http://dx.doi.org/10.1504/IJDATS.2008.020023

[10] Jain, S., Jain, S. and Jain, A. (2013) An assessment of Fuzzy Temporal Rule Mining. *International Journal of Application or Innovation in Engineering and Management* (*IJAIEM*), **2**, 42-45.

[11] Lee, W.-J., Jiang, J.-Y. and Lee, S.-J. (2008) Mining Fuzzy Periodic Association Rules. *Data & Knowledge Engineering*, **65**, 442-462.

[12] Klir, J. and Yuan, B. (2002) Fuzzy Sets and Logic Theory and Application. Prentice Hill Pvt. Ltd., Upper Saddle River.

[13] Dubois, D. and Prade, H. (1983) Ranking Fuzzy Numbers in the Setting of Possibility Theory. *Information Sciences*, **30**, 183-224. http://dx.doi.org/10.1016/0020-0255(83)90025-7

[14] Baruah, H.K. (1999) Set Superimposition and Its Application to the Theory of Fuzzy Sets. *Journal of Assam Science Society*, **10**, 25-31.

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or Online Submission Portal.