# Mining Interesting Periodicities of Temporal Patterns

**Anjana Kakoti Mahanta**[1]
Gauhati University, India,
anjanagu@yahoo.co.in

**Hung Son Nguyen**
Warsaw University, Poland
son@mimuw.edu.pl

## Abstract

Data mining also known as knowledge discovery from datasets has been recognized as an important area of database research. This area can be defined as efficiently discovering interesting patterns from large data sets. In this paper a generic method has been proposed to extract interesting periodicities of patterns from large datasets where the transactions in the data sets are associated with patterns and time intervals in which the patterns hold. Considering the hierarchy associated with time stamps of the form day-date-hour-minutes-seconds, different types of periodic patterns such as daily, weekly, monthly patterns can be extracted.

**Key words:** Temporal pattern, periodicity mining, interesting periodicity.

## 1. Introduction

Electronic data repositories are growing very fast and they contain data from business, scientific, web-related and many other domains. Much of this data is inherently temporal. By taking into account the time aspect, interesting patterns that are time dependent can be extracted. The event logs monitored in computer networks and the click stream data gathered by web sites are huge collection of data generated daily. Finding patterns from such data sets can help provide insight into behavior, buying habits and preferences of the perspective customers of a site. This can then help the web designers to maintain personalized information and create personalized views of their sites.

Detecting temporal patterns such as "rising interest rates", "panic reversal in sales" in the underlying time series is an important data mining task. The detection of patterns in time series requires an approximate or fuzzy matching process. Once detected, pattern instances may be used to construct high level rules. Approximate pattern detection tasks are found in several fields such as signal processing, speech recognition etc. The problem of discovering words in continuous human speech includes important aspects of pattern detection in time series. The process involves matching of pre-stored word templates against a wave form of continuous speech converted into a discrete time series. Speech recognition researchers have used dynamic programming as the basis for word recognition ([1], [9], [10]). The method proposed in this paper for finding interesting periodicity of temporal patterns uses the technique known as dynamic time warping (DTW). The DTW method uses a dynamic programming approach to align a time series and a specific word template so that some distance measure is minimized ([9], [10]). The algorithm considers the possibility that the time axis may be stretched or compressed to achieve a reasonable fit.

The method proposed in this paper also uses another concept known as interval superimposition introduced in [2]. We redefine the concept in a slightly different way to match our situation and also relax a restriction that the underlying intervals should have non-empty intersection. The details are given in section 3. Considering the hierarchy of the time stamps

[1] The work was done when the first author was at Warsaw University, Poland under a Bilateral Exchange Program of Indian National Science Academy and Polish Academy of Sciences from 23rd August 2007 to 21st November 2007.

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 1757–1764
Torremolinos (Málaga), June 22–27, 2008

associated with the events, we propose methods for the discovery of periodic patterns. Most of the existing periodicity mining algorithms assume that the periods are user specified. This assumption is a considerable limitation because knowing the periods a priori is not always possible. The method proposed in this paper does not require the periods to be given by the user. A measure for measuring the interestingness of patterns and their periodicity has been defined to keep only the interesting periodic patterns. Finally a method has been proposed to group similar patterns and to find rules involving the patterns.

In section 2 we discuss recent works done in the field of temporal data mining and time series analysis. In section 3 for the sake of completeness, we formally define the interval superimposition method first. Then we discuss in short the dynamic time warping algorithm. After this in section 4 we discuss the pattern mining method proposed by us. Possible applications of the pattern mining method has been discussed in section 5 followed by conclusion and lines for future research in section 6.

## 2. Recent Works in the Field

Applying data mining techniques to time-series data has recently been emerged as a major area of research. The underlying problem is to discover different types of patterns and in some cases to find the periodicity of patterns. For finding patterns in time-series data, time series analysis has been done in [4]. The problem of mining frequent event temporal patterns has been discussed in [3]. Considering periodic nature of patterns, Ozden [8] proposed a method which can find patterns having periodic nature where the period has to be specified by the user. The problem of mining partial periodic patterns in time-series database is studied by Han et al [6]. The problem of mining frequently occurring periodic patterns with a gap requirement from sequences has been studied in [12]. Given a character sequence S of length L and a pattern P of length l, they consider P a frequently occurring pattern in S if the probability of observing P, given a randomly picked length –l subsequence of S exceeds a certain threshold. Algorithms were proposed for extracting such patterns and the algorithm works well for DNA sequences.

In [5] the problem of detecting the periodicity rate of a time-series data base is addressed. Algorithms were proposed to discover periodic patterns of unknown periods. In that paper the time stamps are not explicitly considered and the input to the algorithm proposed is a sequence of feature values of a feature. Ma and Hellerstein in [7] have developed a linear distance based algorithm for discovering the potential periods regarding the symbols of a time-series. In [11] an algorithm similar to this has been proposed where some pruning techniques have been used. However since both the algorithms consider the distance between adjacent intervals they miss some valid periods. The algorithm proposed in our paper extracts all the periods with certainty values associated with the periodic patterns, which gives an estimate of how much regular the periodic pattern is and the highest value of this certainty function is 1. The method uses a concept called interval superimposition introduced in [2] to extract the periods. Using the certainty values we can compare or rank the periodic patterns. The method uses the hierarchy associated with any calendar date but the same procedure is applicable to any time hierarchy other than calendar dates also. Again all periodic patterns that exist in a time series may not be equally interesting or some of them may be very trivial. What types of periods are actually interesting depends in most cases on the application at hand. We discuss certain common features which can be used to measure interestingness of periods and propose procedures for extracting these using a method known as dynamic time warping to match a given time series against a given time template.

## 3. Works Related to the Paper

In this section we discuss in short the interval superimposition method that we are going to use in our proposed periodicity mining procedure. Then we discuss the dynamic time warping algorithm used to align a time series against a given time template.

### 3.1. Superimposition of Intervals

The concept of superimposition of intervals was introduced in [2]. We define the superimposition process here in a slightly different way keeping the underlying concept same and at the same time relaxing a restriction imposed in [2] that

the participating intervals should have non-empty intersection.

Suppose we consider two intervals [2,5] and [4,7] on the real line (Figure 1). When the interval [4,7] is superimposed on [2,5] then the interval [4,5] will have double representation in the superimposed interval. This can be demonstrated diagrammatically as shown below.

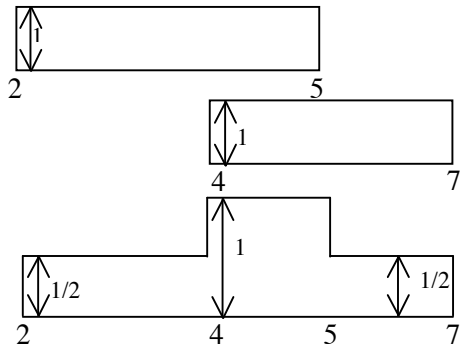


Figure 1. Example of superimposition of two intervals

The new interval formed after superimposition is [2,7] in which the elements in the interval [4,5] will be represented twice. To represent this phenomenon we define a function which we may call membership function, which takes double value for the elements in [4,5] than the elements in [2,4) and (5,7]. Before applying the superimposition process, we associate membership functions to the intervals [2,5] and [4,7] which take unit value for the elements in the respective intervals and zero elsewhere. These membership functions are nothing but the characteristic functions defined on the intervals [2,5] and [4,7] on the real line R. Let us denote these as $\chi^1_{[2,5]}$ and $\chi^2_{[4,7]}$. Then the membership function for the superimposed interval [2,7], which we have denoted as $f$ is defined as

$$f(x) = \frac{1}{2}\chi^1_{[2,5]}(x) + \frac{1}{2}\chi^2_{[4,7]}(x)$$

The superimposition process can be extended to any n, n > 0 number of intervals. Suppose we are superimposing n intervals $[t_1, t_1'], [t_2, t_2'], \ldots\ldots [t_n, t_n']$ and the characteristic functions associated with the intervals are $\chi^1_{[t_1,t_1']}, \chi^2_{[t_2,t_2']}, \ldots\ldots\ldots \chi^n_{[t_n,t_n']}$, then the membership function associated with the superimposed interval $[t, t']$ is

$$f(x) = \frac{\chi^1_{[t_1,t_1']}(x) + \chi^2_{[t_2,t_2']}(x) \ldots + \chi^n_{[t_n,t_n']}(x)}{N} \quad (1)$$

where $N = \max_{x \in R}\left(\chi^1_{[t_1,t_1']}(x) + \ldots + \chi^n_{[t_n,t_n']}(x)\right)$ is the normalized factor, $t$ is the minimum of $t_1, t_2, \ldots t_n$ and $t'$ is the maximum of $t_1', t_2', \ldots t_n'$.

From the definition of the function $f$ it is clear that it takes values between 0 and 1 both inclusive in the interval $[t, t']$ and value 0 outside $[t, t']$. Thus the function $f$ can also be thought of as a fuzzy membership function defined on the real line R. The result will hold good for any continuous or discrete but ordered domain.

## 3.2. Dynamic Time Warping

Extracting periodic patterns from time series data such as "rising interest rates", "lowering down of sales", "panic reversal" is an important task in temporal data mining. One way to detect such patterns is to visualize the patterns pictorially. We human beings are good in visually detecting patterns but automatic detection mechanism i.e. by using Computers to do so is not an easy matter. One way to do so is by matching the time series pattern with some given template pattern. For this we need an appropriate matching or distance function. Also the function should be able to capture the notion of fuzziness as we are interested only in the approximate shape of the pattern which we are expecting. Normally used distance functions such as Euclidean function, Manhattan distance etc, which aligns the i-th point in one time series with the i-th point in another time series produce a poor similarity score. For example consider the following pattern from technical analysis of a stock market where we have the interest rates plotted against time.

We see that sufficient interest rising rates (peaks) are evident at two places. But these peaks may appear anywhere in the time series and we should be able to detect these. A function which can perform non-linear alignment (elastic) will give rise to a more meaningful similarity measure which will allow similar shapes to match even if these are out of phase in the time series.

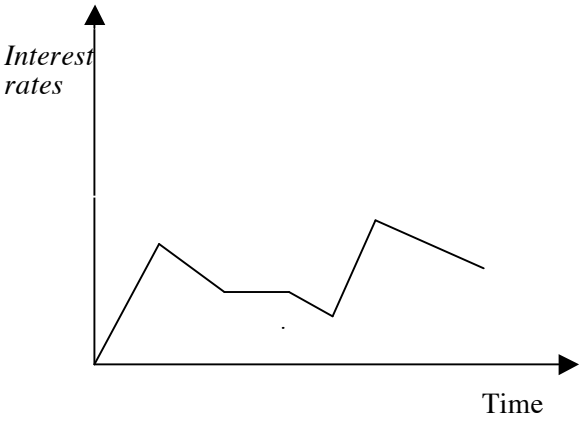


Figure 2. Example of time series data.

The technique of dynamic time warping (DTW) uses a dynamic programming approach ([9], [10]) to align a time series and a specified word template so that some distance measure is minimized. Since the time series is stretched (or compressed) to achieve a reasonable fit, the i-th point in the time series may be aligned with some j-th point in the template (i may or may not be equal to j) to compute the distance between the two. Specifically the pattern detection task involves searching a time series S for instances of a template T, where

$$S = s_1, s_2, s_3, \ldots\ldots\ldots\ldots, s_n$$

$$T = t_1, t_2, t_3, \ldots\ldots\ldots\ldots\ldots, t_m$$

The sequences S and T can be arranged to form a n-by-m grid where each grid point (i, j) corresponds to an alignment between elements $s_i$ and $t_j$ (Figure 3).

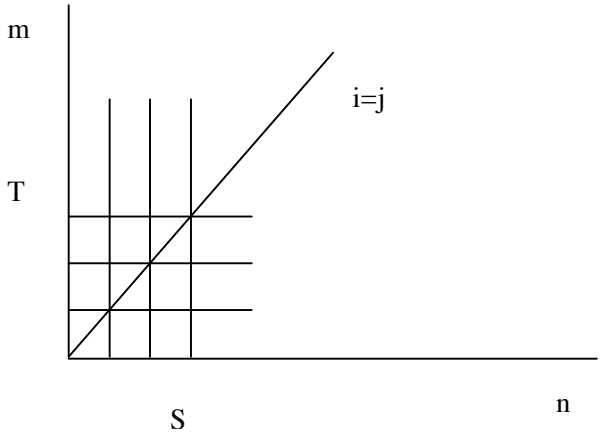


Figure 3. Illustration of dynamic time warping technique.

A warping path W, aligns the elements of S and T such that the distance between them is minimized. W can be written as

$$W = w_1\ w_2 \ldots\ldots\ldots w_p$$

where each $w_k$ corresponds to a point $(i,j)_k$ in the grid. When there is no timing difference, the warping path coincides with the diagonal line i = j. There are a number of distance measures that can be used to find the distance between two elements such as magnitude of the difference, square of the difference etc.

$$\delta(i, j) = |s_i - t_j| \text{ or } \delta(i,j) = (s_i - t_j)^2$$

To find the best match or alignment between these two sequences one need to find a path through the grid which minimizes the total distance between them. The procedure for computing this involves finding all possible routes through the grid and for each one computes the overall distance. Now the dynamic time warping problem can be formally defined as a minimization over potential warping paths based on the cumulative distance for each path.

$$\text{DTW}(S,T) = \min_W \{\sum_{k=1}^{p} \delta(w_k)\}\ .$$

Searching through all possible warping paths is very expensive. To reduce the search space several types of restrictions can be used. Some of these are

(i) *Monotonicity*: the path will not turn back on itself, i.e. for consecutive pairs $w_{k-1}$ and $w_k$ in W, $i_{k-1} \le i_k$ and $j_{k-1} \le j_k$.

(ii) *Continuity*: The path advances one step at a time i.e. $i_k - i_{k-1} \le 1$ and $j_k - j_{k-1} \le 1$.

(iii) *Boundary condition*: the path starts at the bottom left and ends at the top right.

(iv) *Warping window condition:* a good path is unlikely to wander very far from the diagonal. The distance that the path is allowed to wander is the window width: $|i_k - j_k| \le w$, where w is the size of the warping window which is a positive integer.

(v) *Slope constraint condition:* The path should not be too steep or too shallow thereby avoiding excessively large movements in a single direction.

To formulate a dynamic programming problem we need a recurrence relation. Following is a recurrence relation which defines the cumulative distance $\gamma(i, j)$ for each point (i, j) in the grid.

$$\gamma(i, j) = \delta(i, j) + \\ + \min [\gamma(i-1, j), \gamma(i-1, j-1), \gamma(i, j-1)].$$

Other similar recurrence functions can also be used but the above relation is symmetric and it has been seen that symmetric formulations give better result in the speech recognition field. Asymmetric relations could be like

$$\gamma(i, j) = \delta(i, j) + \min [\gamma(i-1, j), \gamma(i, j - 1)].$$

The dynamic programming algorithm fills in a table of cumulative distances as the computation proceeds. Upon completion the optimal warping path can be found by tracing backward in the table.

## 4. Proposed Method of Pattern Mining

First we discuss the general periodic pattern mining procedure in 4.1.1. Not all periodic patterns may be interesting. We next define a measure of interestingness of periodicities and propose methods for finding the interesting periodic patterns in 4.1.2. In 4.1.3 we propose a method for clustering of similar periodic patterns.

### 4.1.1. Mining Periodic Patterns

Suppose we have a pattern database in which each pattern has associated with it a time interval in which the pattern has occurred. Each interval is represented as two time stamps, the starting time stamp and the ending time stamp. We assume that the time stamps are given in a hierarchy such as day-date-minutes-hours-seconds. If the times are the Unix dates then these can easily be converted to the above format. To see if a pattern has any daily or 24 hour periodicity, we shall remove all information about the day and date and keep only the hour-minutes-seconds information. For each pattern we can then carry out superimposition of the time intervals as discussed in the previous section. Before superimposing, each interval will have membership value 1. The membership function f of the superimposed interval can be computed using (1) where n is the total number of days covered by the database. While using the interval superimposition method in extracting periodic patterns we shall use the term certainty function instead of membership function because it sounds more appropriate.

After carrying out the superimposition operation, some time periods in the 24 hour duration will have more certainty values than the others. The highest certainty value that a period can have is 1. So 24 hour duration will be divided into several sub-intervals with different certainty values. The certainty function above can also be thought of as a fuzzy membership function defined over the duration of a day. We can also say that the daily behavior of the pattern is represented by a fuzzy interval. To find weekly periodicity, we shall superimpose the intervals day wise i.e time intervals in Monday will superimpose on time intervals in Monday and so for other days. In this case we shall remove information regarding dates and keep only day-hour-minutes-seconds information. To find behavioral pattern for a particular day, say Monday, we shall superimpose all time periods for which the day = Monday. The same idea can be used to find monthly pattern by removing the year information from the time stamps. This approach could be used if the time stamps are given in any hierarchy not just the hierarchy day-date-hours-minutes-seconds.

### 4.1.2. Extracting Interesting Periodic Patterns

Not all extracted periodic patterns will be useful or interesting. For example if a pattern holds almost daily, that may not be interesting for us. To find interesting periodic patterns we propose a method to study the curves associated with the periodic patterns. After carrying out the superimposition of intervals for finding periodicity of patterns as discussed in 4.1.1, we obtain a curve describing the pattern (we call it time curve). Along the x-axis we have the time stamps and along the y-axis we plot the certainty values of the periodicity. Now we consider patterns having peaks of sufficient height as interesting. This means that sufficient number of intervals in which a pattern holds have non-empty intersection. Since the maximum height of a peak can be 1 (as it is kept normalized), a suitable minimum threshold may be used (say .7 or .8) for the height of the peak. If the pattern has only small peaks then moving one level higher in the time hierarchy may lead to higher peaks. For example suppose some pattern holds on almost every Monday of a week and we are finding daily patterns (24 hour periodicity), then obviously we will have small peaks. Now if we move to weekly patterns then all the time slots on Monday will overlap and we shall obtain a peak of sufficient height. In the process it may

be necessary to move to higher level in the time-hierarchy also to obtain interesting patterns.

To see whether there are peaks in the time curve, the method we propose uses the Dynamic Time Warping (DTW) approach ([9], [10]). By considering a template consisting of a peak and using offset values (offset values allow specifying a non zero value which enables sliding the time series against each other along the time axis to perform matching) the template is to be moved through the entire time period under consideration using DTW technique to find the minimum distance between the template and the associated time curve. But without doing any normalization it is not possible to compare the distance values. Normalization for the path length can be done by using the following formula to compute cumulative distance

$$\gamma(i,j) = \min\{\delta(i,j) + \gamma(i,j-1), \\ 2\delta(i-1,j-1) + \gamma(i-1,j-1), \\ \delta(i,j) + \gamma(i-1,j)\}$$

The length normalized distance function in this case will be

$$\mathrm{D(A,B)} = \frac{\gamma(n,m)}{n+m}$$

Where D is the distance function, A and B are the curves (template and time curve) and n and m are the number of time stamps in the template and the time curve. To make the search robust, only the starting point could be used as the anchor point (anchor point provides the option to explicitly fix together a pair of points from two time series) without mentioning the end point. To measure the quality of the matching produced, baseline score could be used. One way to do this is to create a boundary around the time series.

### 4.1.3. Clustering of Patterns

Patterns having similar periodicities can be clustered to form interesting groups of patterns. The periodicities of two patterns could be compared by comparing the associated time curves. For this we need a normalized similarity measure. By using the DTW technique two time curves could be aligned by using as anchor points the starting and ending points. After finding the matching pairs of points in the time curves, the similarity measure proposed is defined as follows.

Suppose the matching points are

A : $a_1, a_2, \ldots\ldots a_l$ and B: $b_1, b_2, \ldots\ldots b_l$

where A and B are the time curves, the similarity function is

$$sim(A,B) = \frac{\sum_{i=1}^{l} \min\{a_i, b_i\}}{\sum_{i=1}^{l} \max\{a_i, b_i\}}$$

If the curves are exactly same then the value of the similarity function is 1 and also the value lies between 0 and 1. Now any known clustering technique could be used together with a suitable similarity threshold. After finding the clusters, if frequent item set mining algorithm is used within the elements of each cluster then this may lead to interesting rules such as patterns having similar periodicity share some other feature also.

Representation of curves can be done in the usual way by storing a sufficient number of points in the time curve. The points may be uniformly distributed between the starting and ending point of time. The number of points to be taken depends on the application in hand. The representation of a one point cluster can be taken as the representation of the curve itself. When two one point clusters are to be merged, first the similarity value between the two associated curves will have to be computed. If the similarity value is greater than the minimum similarity threshold value then the clusters will be merged. In the process the matching procedure will compute a sequence of pairs of matching points for the two time curves. Suppose the sequence is

$$(x_1, y_1),\ (x_2, y_2)\ \ldots\ldots\ldots\ldots\ldots\ldots (x_l, y_l)$$

$$(x_1', y_1'),\ (x_2', y_2')\ \ldots\ldots\ldots\ldots\ldots\ldots (x_l', y_l')$$

where the point $(x_i, y_i)$ in the first curve is aligned with the point $(x_i', y_i')$ in the second curve for $i = 1,2,\ldots\ldots l$. Then the following sequence of points

$$\left(\frac{x_1 + x_1'}{2}, \frac{y_1 + y_1'}{2}\right), \ldots, \left(\frac{x_l + x_l'}{2}, \frac{y_l + y_l'}{2}\right)$$

will be considered as the representation of the newly formed cluster obtained after merging. These points together represent a curve lying between the two time curves. If the clusters are represented in this way the similarity value between the two clusters can be computed by using the same formula used for computing the similarity value between two time curves. Also in the process of computing the similarity value we get the representation of the merged cluster. Since while matching two time curves we use the starting and ending points as anchor points, the above procedure will work fine.

## 5. Proposed Area of Application

Although there are many, we propose two possible areas of application of the techniques proposed in this paper. These are

1. **Finding temporal behavior patterns of Internet users**: Many Internet market research agencies keep information about web traffic data together with Internet users behavior and their demographic profile. Suppose the visit periods of the users are known, where each visit period is of the from [log_in time, log_out time]. A user may have several visit periods during a day. Using methods proposed in this paper, one can group users as daily morning users, daily evening users, weekly users etc. This information may be useful for communicating with the users. It may also be possible to extract rules saying that the behavioral patterns of the users are related to their job type.

2. **Determining the periodicity**: In an event database, where the events are associated with time intervals in which the events have taken place: Applying techniques proposed in this paper, it is possible to determine periodicity of the events if exists. Also events with similar periodicities may be grouped to find even more interesting patterns.

The study carried out in this paper was motivated by the "Discovery Challenge" held with ECML/PKDD 2007. The ECML/PKDD 2007 held at Warsaw University, Poland, organized a "Discovery Challenge" which was devoted to three problems:

a. User behavior prediction from Web traffic logs
b. HTTP traffic classification
c. Sumerian Literature understanding.

The users behavior challenge was co-organized by Gemius SA, the leading Internet market research agency in Central and Eastern Europe. The underlying objective is to predict users behavior by characterizing nature of users visit i.e. the list of categories of the visited Internet portal and the number of page views in each category. The challenge is accomplished with use of web traffic data from Polish web sites employing Gemius traffic study, grouped by appropriate categories. This kind of study helps in adjusting an offer to needs of given target groups and achieving increased profits from various web-related business activities. Gemius SA provided data in two text files. The exact formats of the files are as follows:

- Users table: This table consists of the following fields:
  *user_id, country_id, region_id, city_id, system_id, system_sub_id, browser_id, browser_ver_id*. The meaning of the fields follows from their names.
- Visit Paths table: This table consists of the following fields:

  *path_id, user_id, timestamp, path {category_id, pageviews_number}..{ }..{ }*

Solutions to the above mentioned problems were requested by the Conference organizers. Training and test data sets were provided, where in the test data set the information about the visit paths were removed. These were later on used to test the quality of the solutions provided.

## 6. Conclusion and Future Research

In this paper we have proposed methods for extracting periodicities of patterns, to find interesting periodicities and also to group patterns according to their periodic nature. Two possible areas of application have been mentioned. Future works include actual implementation of the proposed methods and experimenting with real life data sets. We plan to carry out this part at an earliest.

## References:


[1] Ainsworth, W. A.: Speech Recognition by Machine, London: Peter Peregrinus Ltd. (1988)

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

[3] Bettini, C., Wang, X., Jajodia, S., and Lin, J.: Discovering Frequent Event Patterns with Multiple Granularities in Time Sequences. IEEE Transaction on Knowledge and Data Engineering, Vol – 10, No – 2, pp 222-237 (Mar-Apr 1998)

[4] Clifford, J., Croker, A.:The Historical Relational Data Model (HRDM) and Algebra Based on Lifespans. Proceedings of the International Conference on Data Emgineering, 528-537, Los Angeles, California: IEEE Computer Society Press (1987).

[5] Elfeky, Mohamed G., Aref, Walid G., Elmagarmid, Ahmed K.: Periodicity detection in time series databases. IEEE Transactions on Knowledge and Data Engineering, Vol – 17, No – 7 (July 2005)

[6] Han, J., Dong, G., and Yin, Y.: Efficient Mining of Partial Periodic Patterns in Time Series Database. Proceedings of the 15th Int'l Conf. on Data Engineering, pp. 106-115 (1999)

[7] Ma, S., and Hellestein, J.,: Mining Partially Periodic Event Patterns with Unknown Periods. Proc of 17th International Conference on Data Engineering (April 2001)

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

[9] Rabiner, L. R., and Levinson, S. E. : Isolated and Connected Word Recognition – Theory and selected Applications. Readings in Speech Recognition, eds. Waibel, A. and Lee, K., 115-165. San Mateo, California : Morgan Kaufmann Publishers, Inc. (1990)

[10] Sakoe, H., and Chiba, S.: Dynamic Programming Algorithm Optimization for Spoken Word Recognition. In Readings in Speech Recognition, eds. Waibel, A. and Lee, K., 159-165. San Mateo, California : Morgan Kauffmann Publishers, Inc. (1990)

[11] Yang, J., Wang, W., and Yu, P., :Mining Asynchronous Periodic Patterns in Time Series Data. Proceedings of 6th International Conference on Knowledge Discovery and Data Mining. (August 2000)

[12] Zhang, Minghu., Kao Ben, Cheung W. David, Yip, Kevin Y.: Mining periodic patterns with gap requirement from sequences. ACM Transactions on Knowledge Discovery from Data (TKDD), Vol- 1, Issue – 2 (August 2007)