ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 1, July 2013

Real Time Periodicity Detection in Time-Series Databases

Fathima Kunhi Mohamed, Shaiju Panchikkil
MTech Student, Assistant Professor, MES College of Engineering, Kuttippuram, Kerala, India

***Abstract* — Time series is a set of statistics, usually collected at regular intervals. The goal of analyzing a time series database is to find whether and how frequent a periodic pattern is repeated within the series. Periodic pattern mining is the problem that regards temporal regularity. It has a number of applications, such as prediction, forecasting, detection of unusual activities, etc. Several algorithms have been developed for detecting periodicity in time series databases. A new method is proposed for online data sets which need periodicity detection to be online.**

***Index Terms*— Periodicity detection, suffix array, suffix tree, time series.**

I. INTRODUCTION

A time series is a sequence of recorded values. These values are usually real numbers recorded at regular intervals, such as yearly, monthly, weekly, daily, and hourly [6]. Data recorded irregularly are often interpolated to form values at regular intervals before the time series is analyzed. Time series data appear naturally in almost all fields of natural and social science as well as in numerous other disciplines. People are interested in time series analysis for two reasons:

- Modeling time series - To obtain insights into the mechanism that generates the time series.
- Forecasting time series - To predict future values of the time series variable.

Data reduction is an important data mining concept. Data reduction techniques will reduce the massive data into a manageable synoptic data structure while preserving the characteristic of the data as much as possible. One of the data reduction method is discretization by assigning a letter from a predefined alphabet to each value or range. It is discretized by considering a distinct range such that all values in a range are represented by one symbol. Periodicity detection of a time series is a process for finding temporal regularities within the time series database [7]. In general, there are three types of periodic patterns in a time series. They are: 1) Symbol periodicity 2) Partial periodicity 3) Segment periodicity. Symbol periodicity means only one symbol is periodic. If more than one symbol is periodic and occur partially it is called partial periodicity. Segment periodicity means the whole time series is represented as a periodic pattern. A pattern is said to be periodic if its confidence value is above or equal to the threshold value. The confidence of a periodic pattern X occurring in time series T is the ratio of its actual periodicity to its expected perfect periodicity. Several noise can be present in the time series. Three basic types of noise generally considered in time series analysis are replacement, insertion and deletion noise [2]. In replacement noise, some symbols in the discretized time series are replaced at random with other symbols. In case of insertion and deletion noise, some symbols are inserted or deleted, respectively, randomly at different positions (or time values). Noise can also be a mixture of these three types.

II. LITERATURE SURVEY

Several algorithms have been developed to detect periodicity in time series databases. Some of them are discussed here. One of the earliest best known work has been developed by Elfeky et al. [1]. They proposed two separate algorithms to detect symbol and segment periodicity in time series. Their algorithm (CONV) is based on the convolution technique with reported complexity of O (n log n). Although their algorithm works well with data sets having perfect periodicity, it fails to perform well when the time series contains insertion and deletion noise. Another algorithm for the periodic pattern mining in time series database is periodic pattern mining using suffix tree by Rasheed et al. [2]. STNR (Suffix Tree Noise Resilient algorithm) involves two phases. In the first phase, it builds a tree which is known as suffix tree for the time series database and in the second phase, it uses the suffix tree to calculate the periodicity of various patterns in the time series database. This algorithm is capable of: 1) identifying the three different types of periodic patterns, 2) handling asynchronous periodicity by locating periodic patterns that may drift from their expected positions up to an allowable limit, and 3) investigating periodic patterns in the whole time series. The worst case complexity is O (n^2). Another algorithm by Rasheed et al. [3], which is a modification of STNR have an additional feature of detecting periodic patterns even in a subsection of the time series. The worst case complexity is O (k. n^2), where k is the maximum length of periodic pattern and n is the length of the analyzed portion (whole or subsection) of the time series. Another method has been introduced by Xylogiannopoulos et al. [4] by modifying the data structure used. They utilize Suffix Arrays in data mining instead of the commonly used data structure Suffix Trees to detect repeated patterns in time series with time complexity of O (n log n). This method is also a modification of the work in [3]. A new method of periodicity mining in time series databases is introduced to generate patterns by adding extra flexibilities for the user to facilitate the discovery for those patterns which are generated by skipping intermediate events by Nishi et al. [5]. All the three types of periodicity in one run, that is, the symbol, sequence and segment periodicity can be detected in more flexible and proficient way. Initially, single length patterns are mined and gradually generated larger length patterns by joining interesting and periodic smaller length patterns in each pass. Then in the same pass, the proposed approach calculates occurrence vector for the newly generated patterns. As a next step, the algorithm generates all possible exclusive interesting patterns by allowing event skipping among intermediate events. The number of allowed event skipping within any two events can be determined by the difference vector and user specified maximum event skipping threshold, h. Then the generated patterns are tested for periodicity using periodicity detection algorithm. From the periodic patterns found in this step, the algorithm generates patterns for next phase and this process continues until user specified phase number is reached or no new patterns can be generated. Periodicity detection with skipping events is found to be good only in specific applications such as weather forecast center to predict the humidity for a country with the knowledge in mind that during winter the humidity needs not to be considered. Hence, if the time interval contains any month information which belongs to winter then that information can be neglected.

III. PROBLEM DEFINITION

The ubiquitousness of sensor devices that generate real-time, append-only and semi-infinite data streams has revived the need for online processing. In order to deal with this, online algorithm detecting all the three types of periodicities is proposed. Online algorithm is one that can process its input piece-by-piece in a serial fashion, i.e., in the order that the input is fed to the algorithm, without having the entire input available from the start. So here the problem is to find out an optimal method for periodicity detection in online time series databases.

IV. PROPOSED WORK

The proposed work is periodicity detection in online databases. As the first step, data reduction is done by a descretization technique. The work has incorporated methods used in [3] and [4] and online periodicity detection is included.

V. IMPLEMENTATION DETAILS

Implementation is done in Java. The following methods were implemented and tested.

Descretization: The maximum and minimum values of input values are found and it is segmented based on range factor. Each segment is separated by val:

val = (max - min) / range factor

After this step, need to map each segment into symbols. For this, val is added to minimum value and this range i.e. from minimum to minimum + val is mapped to the symbol 'a'. This process is repeated on each entry of the data.

Suffix array construction: Suffix array is the sorted list of all suffixes of a string. The algorithm Suffix Array Construction (SAC) is used, in which a for-loop calculates each time the substring from position till the end of the string of the time series. Then the array is sorted by calling a sorting function.

Occurrence vector calculation: Occurrence vector is the list of index positions of a substring in the original string. It is calculated for all substring. In this step many substrings will be removed as they may not be repeating.

Periodicity calculation: Periodicity is calculated based on occurrence vector. For each occurrence vector occurVec of size k for pattern X, find the difference vector which is position $i+1$ minus position i of the repeated pattern and calculates the differences (periodicities) p for each pair. Checks if the modulo of the division between starting position and p is equal to the modulo of position i and p. If equal, it means that the specific position i is a repetition with the specific periodicity p and the algorithm increments the count of the specific periodicity. Then calculate the confidence which is the number that periodicity p has been found valid over the perfect periodicity by the equation:

Confidence (p) = count (p) /P(p, startPos, X),

Where

P(p, startPos, X) = (|T|-stPos+1) / p

|T| is the length of the string representing the time series, P is the period and stPos is the starting position of the substring S in the time series T. A pattern is considered to be periodic only if its confidence is above the threshold value which is initially set between 0 and 1.

Algorithm for the online periodicity detection method is as follows.

Step 1: Read the input value one by one at regular intervals.

Step 2: Apply descretization based on range factor, when the number of data elements is at least 2.

Step 3: Apply SAC algorithm to the output of step 2.

Step 4: Find occurrence vector for the suffix array.

Step5: Apply periodicity detection algorithm for the patterns with occurrence vector.

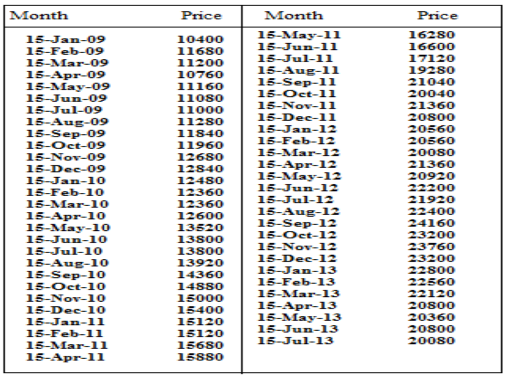
Step 6: Repeat steps 2-5 for each data read at regular interval.

In order to test online periodicity detection, online or real time dataset is needed. Due to the difficulty in getting online database for testing, static databases are simulated to be online. So the above mentioned methods are repeated on each data entry at regular intervals. So periodic pattern found at each interval will be different.

VI. DATA SETS

Real data set is used for experimental evaluation. The data set used is:

1. Monthly gold price in Kerala from 2009 to 2013 [8].



Month	Price	Month	Price
15-Jan-09	10400	15-May-11	16280
15-Feb-09	11680	15-Jun-11	16600
15-Mar-09	11200	15-Jul-11	17120
15-Apr-09	10760	15-Aug-11	19280
15-May-09	11160	15-Sep-11	21040
15-Jun-09	11080	15-Oct-11	20040
15-Jul-09	11000	15-Nov-11	21360
15-Aug-09	11280	15-Dec-11	20800
15-Sep-09	11840	15-Jan-12	20560
15-Oct-09	11960	15-Feb-12	20560
15-Nov-09	12680	15-Mar-12	20080
15-Dec-09	12840	15-Apr-12	21360
15-Jan-10	12480	15-May-12	20920
15-Feb-10	12360	15-Jun-12	22200
15-Mar-10	12360	15-Jul-12	21920
15-Apr-10	12600	15-Aug-12	22400
15-May-10	13520	15-Sep-12	24160
15-Jun-10	13800	15-Oct-12	23200
15-Jul-10	13800	15-Nov-12	23760
15-Aug-10	13920	15-Dec-12	23200
15-Sep-10	14360	15-Jan-13	22800
15-Oct-10	14880	15-Feb-13	22560
15-Nov-10	15000	15-Mar-13	22120
15-Dec-10	15400	15-Apr-13	20800
15-Jan-11	15120	15-May-13	20360
15-Feb-11	15120	15-Jun-13	20800
15-Mar-11	15680	15-Jul-13	20080
15-Apr-11	15880		

Monthly gold price in Kerala from 2009 to 2013

VII. EXPERIMENTAL RESULTS

In order to track the periodicity of patterns, parameters are set. Optimal parameter settings will be different for various applications. The two parameters are:

1. Range factor : It determines the number of segments in which the input values need to be divided. As the first step, range factor segments the entire range of input values by descretization. So it affects the accuracy of the result. On increasing range factor, accuracy of the result increases. In this method range factor is limited to 26.

2. Threshold: It is the lower bound on the value of confidence. When threshold is set nearly 1, only those patterns which are perfectly periodic will be shown as the result.

Perfect- Periodicity (P, stPos, S) =(|T|-stPos+1) / p

and

conf (P, stPos, S) = Actual Periodicity(P, stPos, S) / Perfect Periodicity(P, stPos, S)

Where, |T| is the length of the string representing the time series, P is the period and stPos is the starting position of the substring S in the time series T.

Table 1. Results of the method with different threshold values with range factor 6

Threshold	No. of patterns	Max length Pattern	Range
0.9	0	-	

0.5	6	ee, fffe	e=19573.32-21866.5
			f=21866.5-24160

Table 2. Results of the method with different range factor values with threshold=0.5

Range factor	No. of patterns	Max. length Pattern	Range
4	5	ddddd	d=20720-24160

5	6	dd, eeed	d=18656-21408
			e=21408-24160

Table 3. Results of the method after reading first 12 input values and threshold=0.5

Range Factor	No. of Patterns	Pattern	Range

5	2	a, b	a=10400-10488
			b=10488-10976

6	2	a, b	a=10400-10806.67
			b=10806.67-11213.34

From the Table 1 table, it is inferred that when threshold is increased, number of periodic pattern decreases since the result is becoming more accurate. From the Table 2, it is inferred that for constant threshold, number of periodic pattern depends on range factor. When range factor is increased the range of values represented by a symbol is smaller and the result will be more accurate. From the Table 2 and Table 3, periodic pattern will be different at different intervals and also the range will be different depending on minimum and maximum values of the input. The online construction of a data structure to represent the suffixes is not possible, since the input values are entered at regular intervals in time series. So the string may be different at each stage. As a result, the periodic patterns also vary at different time interval and also the range represented by each symbol.

VIII. CONCLUSION AND FUTURE WORK

The data streams and some other applications have revived the need for online processing which is not possible by offline algorithm. This purpose is fulfilled by online periodicity detection. the proposed method can be extended by using enhanced suffix array construction to reduce time complexity of the method.

REFERENCES

[1] Elfeky, Mohamed G., Walid G. Aref, and Ahmed K. Elmagarmid. "Periodicity detection in time series databases, “Knowledge and Data Engineering, IEEE Transactions, pp. 875-887, 2005.

[2] Rasheed, Faraz, and Reda Alhajj. "STNR: A suffix tree based noise resilient algorithm for periodicity detection in time series databases, “Applied Intelligence, Springer, pp. 267-278, 2010.

[3] Rasheed, Faras, Mohammed Alshalalfa, and Reda Alhajj. "Efficient periodicity mining in time series databases using suffix trees,” Knowledge and Data Engineering, IEEE Transactions , pp. 79-94, 2011.

[4] Xylogiannopoulos, Konstantinos F., Panagiotis Karampelas, and Reda Alhajj. “Periodicity data mining in time series using Suffix Arrays,” Intelligent Systems (IS), 2012 6th IEEE International Conference, IEEE, 2012.

[5] Manziba Akanda Nishi, Chowdhury Farhan Ahmed, Md. Samiullah, Byeong-Soo Jeong. “Effective Periodic Pattern Mining in Time Series Databases, “Expert Systems with Applications. Elsevier, 2012.

[6] Ratanamahatana, Chotirat Ann and Lin, Jessica and Gunopulos, Dimitrios and Keogh, Eamonn and Vlachos, Michail and Das, Gautam. "Mining time series data," Data Mining and Knowledge Discovery Handbook, Springer, pp. 1049-1077, 2010.

[7] Fu, Tak-chung. "A review on time series data mining," Engineering Applications of Artificial Intelligence, Elsevier, pp. 164-181, 2011.

[8] "KeralaGold.com,"Available:http://www.keralagold.com/monthly-gold-prices.htm, (Accessed on 26/07/2013).

AUTHOR BIOGRAPHY

Fathima Kunhi Mohamed, received the BTech degree in Computer Science and Engineering from Government Engineering College, Thrissur in 2009 and doing MTech in MES College of Engineering, Kuttippuram, Kerala, India.

Shaiju Panchikkil, received the BTech and ME degree in Computer Science and Engineering. He is now working as Assistant Professor in MES College of Engineering, Kuttippuram, Kerala, India.