ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 11, November 2013

Study Of Various Periodicity Detection Techniques In Time Series Data And Primary Level Plan For New Technique

Prof. Sandeep Khanna and Mr. Swapnil A. Kasurkar

PADM. DR. V B KOLTE COE Malkapur

Abstract—Research on periodic pattern mining has attained a great focus on nowadays. It is the problem that regards temporal regularity. The rapid growth in data and databases increased a need of powerful data mining technique that will guide to analyze, forecast and predict behaviour of events. Periodicity mining needs to give more attention as its increased need in real life applications. The discovery of patterns with periodicity is of great importance and has been rapidly developed in recent years. The problem of discovering periods for time series databases, referred as periodicity detection. These types of periodicities are available such as symbol periodicity, sequence periodicity and segment periodicity and they are identified even in the presence of noise in the time series database. Using pruning strategy some of these patterns are identified and extracted from the given time series database. There are different techniques already exists for periodic pattern mining. Those existing techniques have their own merits and demerits. In this paper, we are going to discuss on various periodicity mining techniques in Time Series Data as well as plan work for developing efficient periodicity mining techniques in time series data with optimum performance parameter.

Index Terms—Data mining, periodicity detection, time series data

I. INTRODUCTION

The explosive growth in data and databases has generated an urgent need for new techniques and tools that can intelligently and automatically transform the processed data into useful information and knowledge. A time series is a collection of data values gathered generally at uniform interval of time to reflect certain behaviour of an entity. A time series database is one that supplied with data evolving over time. The several examples of time series data are weather conditions of a particular location meteorological data, stock market data, power consumption data and computer network data. A time series is mostly characterized by being composed of repeating cycles[1][2]. Data mining is the process of discovering patterns and trends from the large amounts of data using techniques that uses mathematical and statistical concepts. Research in time series data mining has concentrated on discovering different types of patterns. Periodicity mining is a tool that helps in predicting the behaviour of time series data [3]. For example, periodicity mining allows an energy company to analyze power consumption patterns and predict periods of high and low usage so that proper planning may take place.

Research in time series data mining has concentrated on discovering different types of patterns: sequential patterns [4], temporal patterns [5], periodic association rules [6], partial periodic patterns [7], and surprising patterns [8] to name a few. These periodicity mining techniques require the user to specify a period that determines the rate at which the time series is periodic. They assume that users either know the value of the period beforehand or are willing to try various period values until satisfactory periodic patterns emerge. Since the mining process must be executed repeatedly to obtain good results, this trial-and-error scheme is clearly not efficient. Even in the case of time series data with a priori known periods, there may be obscure periods and, consequently, interesting periodic patterns that will not be discovered. The solution to these problems is to devise techniques for discovering potential periods in time series data. Research in this direction has focused either on devising general techniques for discovering potential periods or on devising special techniques for specific periodicity mining problems. Both approaches require multiple phases over the time series in order to output the periodic patterns themselves.

The problem of discovering periods for time series databases, referred as periodicity detection. These types of periodicities are available such as symbol periodicity, sequence periodicity and segment periodicity and they are identified even in the presence of noise in the time series database [9]. There are different techniques available for periodicity detection. They have their own merits and demerits. This paper provides some discussion about some of the techniques available for periodicity detection. It also includes primary level discussion on one efficient technique that has optimum performance parameter.

II. LITERATURE SURVEY

M.G. Elfeky, W.G. Aref, and A.K. Elmagarmid have developed an efficient algorithm for detecting each type of periodicity in O(nlogn).[9] In that paper, they have defined two types of periodicities for time series databases. Whereas symbol periodicity addresses the

periodicity of the symbols in the time series, segment periodicity addresses the periodicity of the entire time series regarding its segments. They have proposed a scalable, computationally efficient algorithm for detecting each type of periodicity in O(nlogn) time, for a time series of length n. An empirical study of the algorithms using real- world and synthetic data sets proves the practicality of the problem, validates the accuracy of the algorithms, and validates the usefulness of their outputs. Moreover, segment periodicity detection takes less execution time whereas symbol periodicity detects more periods. They conclude that in practice, segment periodicity detection could be applied first and if the results are not sufficient, or not appealing, symbol periodicity detection can be applied. They also have studied the integration of their proposed periodicity detection algorithms in the entire process of time series mining, and have proved its effectiveness in the case of partial periodic patterns mining.

M.G. Elfeky, W.G. Aref, and A.K. Elmagarmid, mentioned the concept of time warping for periodicity algorithm.[10] In that paper, they have proposed a time warping algorithm, named WARP, for periodicity detection in the presence of noise. To handle efficiently all types of noise, WARP extends or shrinks the time axis at various locations to optimally remove the noise. Furthermore, he have proposed an online version of WARP that fits the data stream model. An empirical study using synthetic data shows that there is a tradeoff between noise resiliency and time performance. WARP is more noise resilient, yet requires more processing time, than the previous periodicity detection algorithms. Moreover, Online WARP is shown empirically to be reasonably accurate, even under low memory resources.

David Lo et al., [11] put forth a novel method, frame work, and tool for mining inter-object scenario-based specifications in the form of a UML2-compliant variant of Damm and Harel's live sequence charts (LSC). LSC as a specification language extends the partial order semantics of sequence diagram with temporal liveness and symbolic class level lifeliness to generate compact specifications. The output of this algorithm is satisfying the given thresholds of support and confidence, mined from an input program execution race. The author uses search pruning strategy, specifically adapted to LSCs, which provides efficient mining of scenarios of arbitrary size.

Live sequence charts (LSC), a visual model, scenario-based, inter-object language is proposed by David Lo et al., [12] to investigate the problem of mining scenario-based triggers and effects from program execution tracers. The author uses data mining methods to provide significant and complete results of modulo user-defined thresholds. The input trigger and effect scenarios and the resulting candidate modal scenarios are represented and visualized using a UML2- complaint variant of LSC.

Jinlin Chen [13] presented an updown directed acyclic graph approach for sequential pattern mining. Sequential pattern mining is an important data mining problem that detects frequent subsequences in a sequence database. The author proposed an UDDAG for fast pattern growth. It is a new novel data structure, which supports bidirectional pattern growth from both ends of detected patterns. With UDDAG, at level i recursion, we may grow the length of patterns by 2i-1 at most. Thus, a length-k pattern can be detected in [log2 k+1] levels of recursion at best and that will give result in fewer levels of recursion and faster pattern growth.

A suffix tree based noise resilent algorithm for periodicity detection in time series database is proposed by Faraz Rasheed et al., [14]. They present a noise resilent algorithm using suffix tree as an underlying data structure. This algorithm not only calculates symbol and segment periodicity, but also detects the partial periodicity in time series. It also efficiently detects periodicity in the presence of noise compared with existing algorithm. It detects periodicity in the presence of replacement, insertion, deletion or a mixture of any of this type of noise. The authors improve their previous algorithm by incorporating the time tolerance window so as to make it more silent to insertion and deletion noise.

Efficient periodicity mining in time series databases using suffix tree is proposed by Faraz Rasheed et al., [15].Time series database is a collection of data values stored at uniform interval of time to show the behavior of an entity. Periodicity detection is a method for detecting temporal regularities within the time series and the goal of analyzing this database is to find whether and how frequent a periodic pattern is repeated within the series. Here, the data to be analyzed are mostly noisy and there of different periodicity types. The author used STNR as a suffix-tree based algorithm for periodicity detection in time series data. This algorithm is noise-resilient and run in O (kn2) in the worst case. This method also found symbol, sequence and segment periodicity in the time series.

David Lo et al. [16] provides mining iterative generators and representative rules for the specification of software. It is best if the software is developed with clear, precise and documented specifications. But the software products are often come with poor, incomplete and even without any documented specifications. These factors are contributed to high software maintenance cost. This is mainly due to the effort put in comprehending or understanding the software code base. So, to improve program understanding, author introduces iterative pattern mining that outputs pattern that are occurred frequently within a program trace. Frequent program behaviors that in turn represents software specifications. So, author introduces mining closed iterative patterns (ie) maximal patterns without any superpattern having the same support. These generators can be joined with the closed patterns to produce a set of rules called

representative rules for forward, backward in-between temporal conditions among events in one general representation.

Avrilia Floratou et al. [17] give a technique for efficient and accurate discovery of patterns in sequence datasets. The main aim of sequential data mining applications is to discover frequently occurring patterns. The challenge behind this frequent pattern is allowing some noise in the matching process. The main thing is the definition of a pattern and the definition of similarity between two patterns. This definition of similarity can vary from one application to another. The Author presents a new algorithm called FLAME (Flexible and Accurate Motif Detector) is a flexible suffix tree based algorithm that can be used to find frequent patterns with a variety of definition of motif (pattern) models. FLAME is accurate, fast and scalable one.

Jae-Gil Lee et al. [18] proposed a technique for mining discriminative patterns for classifying trajectories on road networks. Feature-based classification is used in the field of data mining. Using this method, features are extracted from the data points and that points are transformed into feature vector. Each vector represents the existence of features in its corresponding data point. For effective classification, we require the discovery of discriminative features. This method uses frequent pattern for classification. To know the usefulness of frequent pattern, in the classification first analyze the behavior of trajectory data on road networks. By analyzing it, what they have observed means, in addition to the location where vehicles have visited, the order of these locations is important one for improving classification accuracy. Based on the author's analysis, he assured that frequent sequential patterns are compressed with previous method that uses only individual good feature candidates since they maintain this order information. This pattern also improves classification accuracy by 10-15%.

Obules u et al., [19] suggests a pruning strategy to remove redundant data in spatiotemporal database. The spatiotemporal data movements obey periodic patterns. (ie) the objects follow the same route over regular time intervals. Author presented the pattern matching technique to find the patterns that were repeated in the time-series database. Three kinds of patterns such as symbols, sequence and segment periodicity are also discovered. Using pruning strategy redundant data are deduced in order to reduce the memory usage and complexities.

III. PROPOSED ALGORITHAM

A. *Drawbacks of existing algorithm*

Many techniques, algorithm were developed to achieve periodicity detection. They are having some disadvantages which are discussed below:

1. **CONV:** Based on convolution Technique with reported complexity of O(nlogn) Fails to perform well when the time series contain insertion and deletion noise.
2. **WARP:** Based on time warping technique reporting $O(n^2)$ Only detect segment periodicity.
3. **PARPER:** Requires the user to provide the expected period values and runs in linear O(n) time Complexity would increases to $O(n^2)$ for all possible period Only detect partial periodic pattern.

B. *Plan for proposed algorithm*

1. Problem definition

To develop a algorithm that could tackle the problem well by being capable.

1) To identify the three different types of periodic patterns.

2) To investigate periodic patterns in the whole time series as well as in a subsection of the time series.

3) To handle asynchronous periodicity by locating periodic patterns that may drift from their expected positions up to an allowable limit.

4) Algorithm should be noise-resilient.

2. Reason for selecting problem

1) A need for a comprehensive approach capable of analyzing the whole time series or in a subsection of it to effectively handle different types of noise (to a certain degree) and at the same time is able to detect different types of periodic patterns

2) It has number of application

3. Research component

1) Accurate and efficient mining of periodicity

2) Try to optimize suffix tree complexity

3) Datasets are Temperature and Weather data and Electricity load data.

4) MATLAB 7.10 will be the execution environment

C. *Project plan*

1) Study of various periodicity detection algorithms.

2) Analyzing existing periodicity detection algorithms.

3) Implementation of efficient periodicity detection algorithm using suffix tree as underlying data structure.

4) Use redundant period pruning, so that the algorithm does not waste time to investigate a period which has already been identified as redundant

5) Comparison with other existing algorithms

D. System Design & Architecture

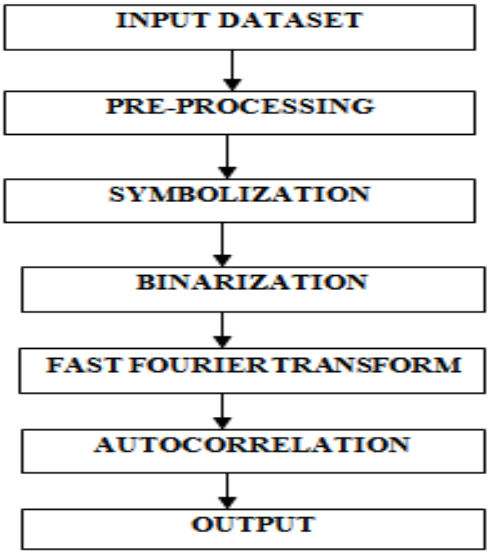

Fig 3.1 Design & Architecture

E. System Architecture

1) **Input Dataset:** In our research work, the following input parameters along with time series data will be considered

- INPUT PATTERN (STRING)
- CONFIDENCE MEASURE
- MINIMUM SUPPORT
- PERIOD OF INTEREST

2) Pre-processing: In this step of processing concerned will be with proper categorization of data stream provided as input.

3) Symbolization: The time series database is a large volume of data, non-finite, noise interference forms [7].It is infeasible to analyze large data manually. So automatic or semi automatic tools are used for data analysis. The time series database should be symbolized in order to improve analysis that is complex

4) Fast Fourier Transformation: FFT is applicable for further processing generated in above step.

5) Autocorrelation: Correlation is a mathematical function related with making data stream with desired application.

6) Output: We will use Temperature and Weather data as input. And the input data will be processed using proposed algorithm in order to mine periodicity efficiently.The system will be operated by interactive GUI as a front end. Results will be displayed in tabular and graphical form.

F. Requirement Analysis

Requirement can be designed on following Basis

1) Study of various periodicity detection algorithms.

2) Analyzing existing periodicity detection algorithms.

3) Implementation of efficient periodicity detection algorithm using suffix tree as underlying data structure.

4) Use redundant period pruning, so that the algorithm does not waste time to investigate a period which has already been identified as redundant

5) Comparison with other existing algorithms

System requirement can be given as follow:

- Accurate and efficient mining of periodicity
- Try to optimize suffix tree complexity
- Datasets are Temperature and Weather data and Electricity load data.
- MATLAB 7.10 will be the execution environment.

By observing various periodicity mining techniques and finding their merits and demerits, we are trying to design and implement an Efficient Periodicity Mining technique in time series data. The main aspect for designing this algorithm is to achieve optimum performance parameter. In this research paper we are planning for an efficient

algorithm that gives accurate result with optimum performance parameter.

A time series is a collection of data values gathered generally at uniform interval of time to reflect certain behaviour of an entity. Real life has several examples of time series such as weather conditions of a particular location, spending patterns, stock growth, transactions in a superstore, network delays, power consumption, computer network fault analysis and security breach detection, earthquake prediction, gene expression data analysis, etc. A time series is mostly characterized by being composed of repeating cycles[13][14].

In this research work, the focus will be on an efficient algorithm for periodicity mining in time series. The algorithm can detect symbol, sequence (partial) and segment (full cycle) in time series. The problem is not trivial because the data to be analyzed are mostly noisy and different periodicity types (namely symbol, sequence, and segment) are to be investigated. Accordingly, we can say that there is a need for a comprehensive approach capable of analyzing the whole time series or in a subsection of it to effectively handle different types of noise and at the same time is able to detect different types of periodic patterns.

Many algorithms, techniques have already been developed for periodicity detection, some techniques find the various patterns like discovering different types of patterns: sequential patterns [15], temporal patterns [16], periodic association rules [17], partial periodic patterns [18], and surprising patterns [19] also some of techniques finds periodicity like segment periodicity, symbol periodicity and sequence periodicity. Each of these techniques finds specific patterns for finding periodicity. Each algorithm has performance factor. In this paper, we will cover all these periodic patterns and periodicity, and finding techniques that has better or optimal performance factor. Again the output may be less accurate due to noise so this technique will completely Resilience to Noise. All three types of noise replacement, insertion, deletion noise and the mixture will be observe and the detected. By finding their behaviour this algorithm will reduces near about all issues created by noise. Time complexity is very important for each and every algorithm by using this technique the time complexity will be reduce as compare to existing periodicity mining techniques in time series data.

IV. PROPOSED SOFTWARE DESCRIPTION

The scope of the proposed work is associating Data mining under the field of Computer Science & Engineering. MATLAB 7.10 can be used to implement and evaluate the performance of the algorithm. Datasets can be used for evaluation are Temperature and Weather data and the parameters for evaluation are

- Accuracy of periodicity mining
- Time Complexity

In this research work, the following input parameters along with time series data will be considered

- INPUT PATTERN (STRING)
- CONFIDENCE MEASURE
- MINIMUM SUPPORT
- PERIOD OF INTEREST

And the result of experiment on time series data set will be presented and evaluated on the basis of key factors ACCURACY and TIME PERFORMANCE.

Here, we will use Temperature and Weather data as input. And the input data will be processed using proposed algorithm in order to mine periodicity efficiently.The system will be operated by interactive GUI as a front end. Results will be displayed in tabular and graphical form.

A. Advantages of system

1) Specific: Matlab, is quite generic

2) Speedy: Matlab is just way too slow. Matlab itself was built upon Java. Also Java was built upon C. So when we run a Matlab program, our computer gets busy trying to interpret and compile all that complicated Matlab code. Then it is turned into Java, and finally executes the code.

3) Efficient: Matlab uses just way too much system resources.

B. Limitations

1) Hardware requirements

Though the system runs fine on higher configurations, when a system has an inferior configuration, the system may not be smooth and drowsiness detection will be slow.

2) Varied weather

Boundaries come when there is change in temperature within less period of time.

V. APPLICATION

Periodic pattern mining or periodicity detection has a number of applications;

Some of them are listed below:

- This technique is useful in earthquake prediction.
- Weather forecasting.
- Power consumption.

- Fraud detection applications.
- It is also applicable in the areas of biological and DNA sequences.

VI. CONCLUSION

This paper includes study of various periodicity mining techniques. It also consisting of planning for the better techniques that give accurate output with less time complexity and output is free of noise. We can design and implement this algorithm on any data sample like temperature, stock etc. In the next paper the exact output of the algorithm with detail discussion will be presented. Here, we proposed an algorithm which can detect all the three types of periodicity in single run, look for all possible period in the time series and perform well when the time series contain insertion and deletion noise.

REFERENCES

[1] M. Ahdesmki, H. Lhdesmki, R. Pearson, H. Huttunen and O. Yli-Harja, "Robust detection of periodic time series measured from biological systems", BMC Bioinformatics, 6:117, 2005.

[2] E. F. Glynn, J. Chen and A. R. Mushegian, "Detecting periodic patterns in unevenly spaced gene expression time series using Lomb–Scargle periodograms." Bioinformatics, Vol.22, No.3, pp.310-316, Feb. 2006.

[3] A. Weigend and N. Gershenfeld. Time Series Prediction: Forecast- ing the Future and Understanding the Past. Addison-Wesley, Read- ing, Massachusetts, 1994.

[4] R. Agrawal and R. Srikant, "Mining Sequential Patterns," Proc. 11th Int'l Conf. Data Eng., Mar. 1995.

[5] C. Bettini, X. Wang, S. Jajodia, and J. Lin, "Discovering Frequent Event Patterns with Multiple Granularities in Time Sequences," IEEE Trans. Knowledge and Data Eng., vol. 10, no. 2, pp. 222-237, Mar./Apr. 1998.

[6] B. Ozden, S. Ramaswamy, and A. Silberschatz, "Cyclic Associa- tion Rules," Proc. 14th Int'l Conf. Data Eng., Feb. 1998.

[7] W. Aref, M. Elfeky, and A. Elmagarmid, "Incremental, Online, and Merge Mining of Partial Periodic Patterns in Time-Series Databases," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 3, pp. 332-342, Mar. 2004.

[8] E. Keogh, S. Lonardi, and B. Chiu, "Finding Surprising Patterns in a Time Series Database in Linear Time and Space," Proc. Eighth Int'l Conf. Knowledge Discovery and Data Mining, July 2002.

[9] M.G. Elfeky, W.G. Aref, and A.K. Elmagarmid, "Periodicity Detection in Time Series Databases," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 7, pp. 875-887, July 2005.

[10] M.G. Elfeky, W.G. Aref, and A.K. Elmagarmid, "WARP: Time Warping for Periodicity Detection," Proc. Fifth IEEE Int'l Conf. Data Mining, Nov. 2005.

[11] D. Lo, S. Maoz, and S.-C. Khoo, "Mining Modal Scenario- Based Specifications from Execution Traces of Reactive Systems," Proc. ACM/IEEE Int'l Conf. Automated Software Eng., 2007.

[12] D. Lo and S. Maoz, "Mining Scenario-Based Triggers and Effects," Proc. ACM/IEEE Int'l Conf. Automated Software Eng., 2008.

[13] Jinlin Chen et. Al., "An UpDown Directed Acyclic Graph Approach for Sequential Pattern Mining", IEEE Transactions On Knowledge And Data Engineering, Vol. 22, No. 7, July 2010

[14] F. Rasheed and R. Alhajj, "STNR: A Suffix Tree Based Noise Resilient Algorithm for Periodicity Detection in Time Series Databases," Applied Intelligence, vol. 32, no. 3, pp. 267-278, 2010.

[15] Faraz Rasheed et. Al., "Efficient Periodicity Mining in Time Series Databases Using Suffix Trees", IEEE Transactions On Knowledge And Data Engineering, Vol. 23, No. 1, January 2011

[16] David Lo et. Al., "Mining Iterative Generators and Representative Rules for Software Specification Discovery", IEEE Transactions On Knowledge And Data Engineering, Vol. 23, No. 2, Febrauary 2011

[17] Avrilia Floratou et. al., "Efficient and Accurate Discovery of Patterns in Sequence Datasets", IEEE Transactions on Knowledge and Data Engineering, 2011

[18] Jae-Gil Lee et. Al., "Mining Discriminative Patterns for Classifying Trajectories on Road Networks", IEEE Transactions On Knowledge And Data Engineering, Vol. 23, No. 5, May 2011

[19] O.Obulesu et. Al., " Finding Maximal Periodic Patterns and Pruning Strategy in Spatiotemporal Databases", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4, April 2012 ISSN: 2277 128X