ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Periodic Outlier Detector For Fuzzy Time Series

Sreelakshmy M K.

M.Tech Student, Dept. of CSE, Viswajyothi College of Engineering and Technology, Vazhakulam, India

ABSTRACT: There are several methods and techniques to detect the periodicity of outliers in time series. Time series data are the data in which it contains value verses time in a fixed interval of time. So that there should be a single value for a particular time, and the time interval between the two adjacent data element should be same. In the case of fuzzy time series data, there can be two type of fuzziness. Either the length of the time interval may vary or one particular time slot may contain several numbers of values. So there is fuzziness in the input data when dealing with fuzzy time series data. The system is to detect the periodicity of outliers in fuzzy time series. The suffix tree method is used to create the tree structure for the equivalent string of fuzzy dataset. The suffix tree is one of the best methods of representation for string processing and this advantage is utilised to find the periodicity of outliers by means of pattern detection.

KEYWORDS: Time series data, outliers, fuzziness.

I. INTRODUCTION

A fuzzy time series data features recorded regularly at non-uniform interval of time period. Real life has many examples of time series such as weather data, stock exchange price movement, road traffic or network traffic density pattern, sensory data, transactions data record, etc. Fuzzy time-series database captures such fuzzy time-ordered characteristics, and data mining aimed at discover in the database hidden information, which cannot be found by classical query languages which used in databases like SQL. Often the information is of pattern format, which may lead toward some kind of rules. For instance, association rules mining discovers patterns in which various data items present together and then try to create some rules based on the discovered association. Periodicity detection in fuzzy time-series databases is a data mining problem where periodically repeating patterns are discovered. Periodic patterns are found in weather records, bank transactions history, movement of stock price, road and computer network traffic density, gene expression, etc.

Detection of periodicity in fuzzy time-series identifies the periodical functions to get seasonal behaviour using Fourier transforms. Periodic pattern mining is a very critical task, as it facilitates analysis of data leading to forecast or prediction of future patterns and events. It also helps us in finding the abnormal activities or anomalies in the given data. So it can be say that events may occur at unexpected time. The latter patterns are completed by the suffix tree method proposed here in this paper. Further, by considering the recent economical issues, it should be possible to understand that similar economic situations have been observed in the past years shall the historical data be available and properly studied. It would be interesting to filter out whether the loss in economy has a periodic pattern. Discovering how periodically repeating other characters are arranged with the loss in economy might help in discovering the connection between different characteristics. Although the problem has been handled before in time-series data, it is very clear that decline in the economy is an outlier, unusual, or surprising activity, which does not happen repeatedly; in addition to this, the repetitions are not exactly periodic (say exactly after five years), and the period value might be significantly much larger than the regular frequent patterns (compare five years with regular weekly or monthly periods).

Periodic pattern mining algorithms usually give more significance to patterns that appear more frequently or have higher support in the analyzed sequence. Here the problem is to detecting the periodicity of outlier patterns in a fuzzy time series by giving more significance to less frequent yet periodic patterns. Discovering the outlier patterns is a

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

challenge because periodicity detection algorithms usually report large number of patterns and manual discovery of unusual patterns from such large number of frequent and regular patterns is very difficult, time consuming, and error prone. The periodicity detection is quiet difficult where the data set is a fuzzy time series.

In the proposed system, time series data alone is not used for outlier detection, instead fuzzy time series data is used for detecting outliers. Thus outliers can be detected from both time series data and fuzzy time series data.

II. RELATED WORKS

A time series data may show different types of periodicity; like, a single symbol is periodic called symbol periodicity, a collection of patterns are periodic and called as partial periodic patterns or sequence periodicity, or the entire time series is periodic with the same pattern is called segment or full-cycle periodicity. For instance, in string = abcabcabcabc, beside others, symbol 'a' and pattern 'bc' are periodic has period value 3, while the series itself has full-cycle periodicity for the pattern 'abc'. There are several algorithms that can find out the frequent periodic patterns having minimum number of repetitions or with minimum confidence value. Confidence value is the ratio between number of occurrences found and maximum possible occurrences. There are not much work has been done for detecting the periodicity of outlier patterns. It is very important to note that surprisingly, unusual, or outlier patterns are different from outlier in the data.

There are many techniques to find local and/or global outliers in the data, but outlier or surprising patterns are different from others patterns. For example, in a certain sequence, events 'a' and 'b' might not be outliers but the pattern 'aba' (a certain combination of the events) might be an outlier pattern. There are few algorithms, which discover the surprising patterns in time series.

1. Efficient Periodicity Mining in Time Series Databases Using Suffix Trees

Periodicity detection or periodic pattern mining has a number of variety applications, like future prediction, event forecasting, unusual activity detection, etc. The issue is not major because the data to be checked are mostly have noisy content and different types of periodicity like symbol periodicity, sequence periodicity, and segment periodicity. These kinds of periodicities are to be investigated and analysed. Accordingly, there is a need for a comprehensive method capable of checking the complete time series data or in a subsection of that to efficiently and effectively handle different types of noise to a degree and at the same time it is able to detect different kinds of periodic patterns; and combining these under one area is by itself a challenge. An algorithm which can detect both partial and full cycle periodicity like symbol, sequence, and segment periodicity in time series is proposed. The used algorithm is suffix tree algorithm and used a suffix tree as the underlying data structure; this allows us to make the algorithm such that its worst-case complexity is O (l, n2), where l is the maximum length of periodic pattern and n is the length of the analyzed portion of the time series. It can be whole section or subsection. The algorithm is noise resilient; it has been successfully demonstrated to work with replacement, insertion, deletion, or a mixture of these types of noise.

2. Pattern Recognition and Classification for Multivariate Time Series

Currently we are faced with lots of data which are fast growing and permanently evolving in nature. Such data including data from social networks and sensory data recorded from moving objects, smart phones or moving vehicles. In the case of temporally evolving data, which can brings a lot of new adventures to the data mining area and machine learning community. This paper is really concerned with the identification of recurring patterns within several multivariate time series records, which records the evolution of multiple parameters over a fixed period of time. This method initially separates a time series record into many segments that can be considered as several situations, and then clusters the identified data segments into number of groups of similar content. The time series data segmentation is published in a bottom-up way according the connection of the individual signals. Recognized data segments are clustered in terms of statistical characteristics using agglomerative hierarchical clustering. The proposed method is analyzed and evaluated on the basis of real sensor data from many vehicles captured during car drives. According to the analysis or evaluation it is able to recognize recurring patterns in multivariate time series by means of bottom-up data segmentation and hierarchical clustering.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

III. ARCHITECTURE

For implementation of the proposed work uses python language. Technical details on how the system is implemented have been discussed in detail in the sections that follow. The system PODF (Periodic Outlier Detector for Fuzzy time series) is modularized as follows,

Modules

- Data extraction
- String processing
- Build suffix tree
- Build PFT
- Outlier Periodicity Detection

Data extraction:

Here the dataset used is fuzzy time series data. It contains missing time slots and this consists of non-unique time intervals. First it is needed to make this fuzzy data into non fuzzy time series. The data extraction module perfectly did this requirement. The missing time slot data is extracted and automatically fill the slots with sequential time data and also fill the value for the newly added slots by null value appending method of data processing.

String processing:

The string processing module is for make the data string first, by extracted data. Then the data string is converted to an equivalent string of alphabets. So the alphabet string is better for string processing. Also the efficiency of string processing is more in this kind of alphabet strings because, several ranges of values can be represented by a single alphabet. So when this module is executed the data string is processed and gets an equivalent alphabet string for further processing. Finally put a $ symbol at the end of the alphabet string in order to fix the end of string. This will be useful when rest of the processing of that particular string. The architecture of the proposed system is given in figure 1.

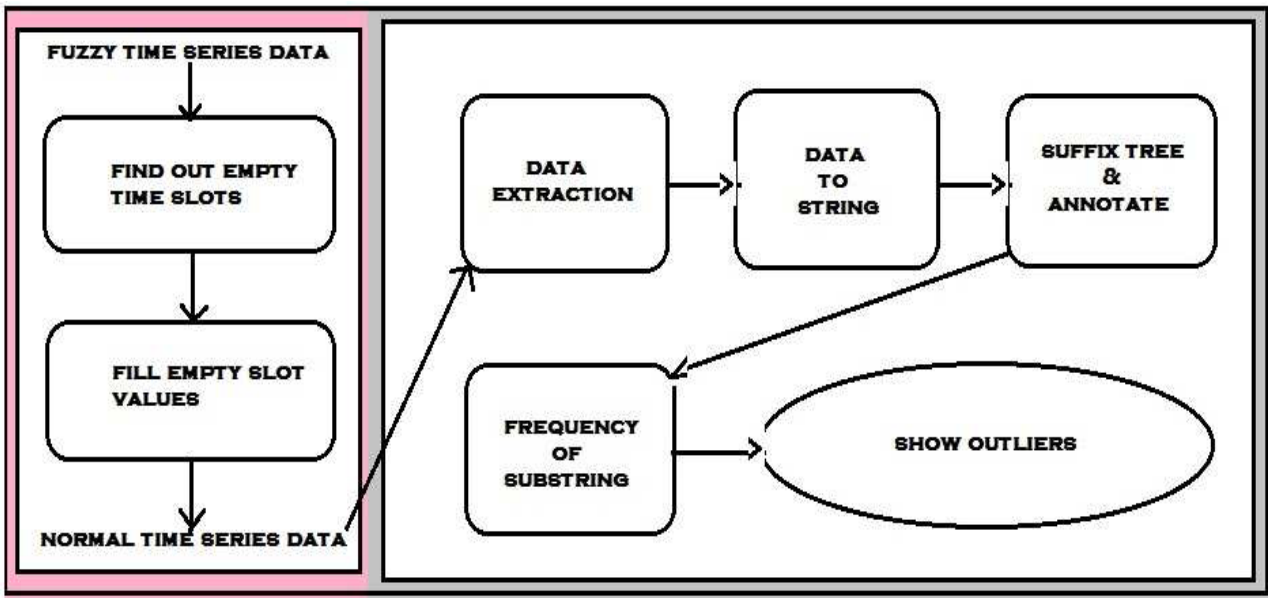

Figure 1: Architecture

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Build suffix tree:

Suffix tree is a data structure which is very helpful in string processing and commonly used data structure that has been proved very useful in string processing. It can be efficiently used to find a substring in the original string, to find the frequent substring and other string matching problems. A suffix tree for a string represents all its suffixes. It contains a distinguished path from the root for each of the suffixes of the string. The most important aspect of the suffix tree related to this work is that it very efficiently captures and highlights the repetitions of substrings within a string. The $ symbol denotes the end marker for the string, a unique symbol that does not appear anywhere in the string (also called a sentinel). The path from the root to any leaf represents a suffix for the string. Since a string of length n can have exactly n suffixes, the suffix tree for a string also contains exactly n leaves. Each edge is labelled with the string that it represents. Each leaf node holds a number that represents the starting position of the suffix yield when traversing from the root to that leaf. Each intermediate node contains a number which is the length of the substring read when traversing from the root to that intermediate node. Each intermediate edge reads a string (from the root to that edge), which is repeated at least twice in the original string. A suffix tree can be generated in linear time using Ukkonens algorithm; it is online algorithm, i.e., it allows extending a suffix tree as new symbols are added to the string. A suffix tree for a string of length n can have at most 2n nodes, and the average depth of the suffix tree is of the order log (n). Here the suffix tree for the alphabet string is constructed and traverse the tree for number of occurrence of each substring of the giver string of alphabets.

IV. PERFORMANCE EVALUATION

The performance evaluation of the proposed system in terms of execution time and response time versus length of the string are presented. Here present the results of the experiments performed to test various aspects of the algorithm employed in outlier patterns detection. The algorithm is compared with STNR. We test the algorithm using stock exchange data. The data contain the stock exchange data of the number of days. All the experiments are compared with STNR to highlight areas where FUZZY performs better than existing approaches. We also demonstrate how the time performance of the algorithm remains almost unchanged and is not affected by the number of outlier patterns detected, provided the overall series size remains the same. The results also show that the time taken by FUZZY is not affected by the number of periods.

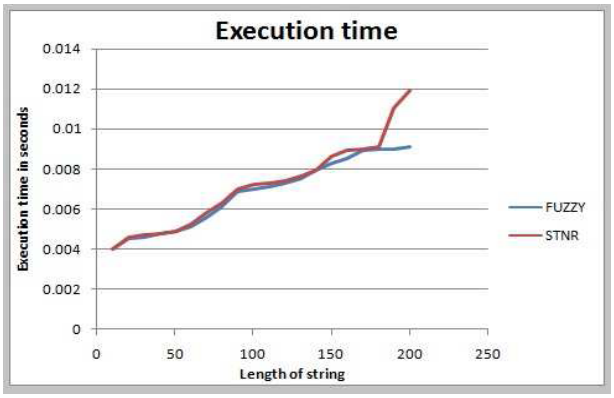

Figure 2: Execution time v/s Length of string.

Execution Time v/s Length of String:

In this section compare the time for execution of both algorithms (STNR and FUZZY) by increasing the number of outlier patterns by keeping the overall series size constant. We used the series of size 100 000 produced using ten alphabets having the regular periodicity of 10. Outlier patterns of length 5 are embedded at the end of the series. The results are presented in Fig.2. The response time of FUZZY is not affected by the number of embedded outlier patterns. Similarly, the response time of STNR also remains constant once the outliers are introduced. STNR did take more time to calculate surprising patterns compared with when there were no outliers in the series but afterward it steadies. For

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

FUZZY data, even increased the number of outlier patterns to 2000 and 4000, but the time performance remained almost the same. This is because no matter how many occurrences of the outlier patterns are there, they would be detected by just a single traversal of the occurrence vector.

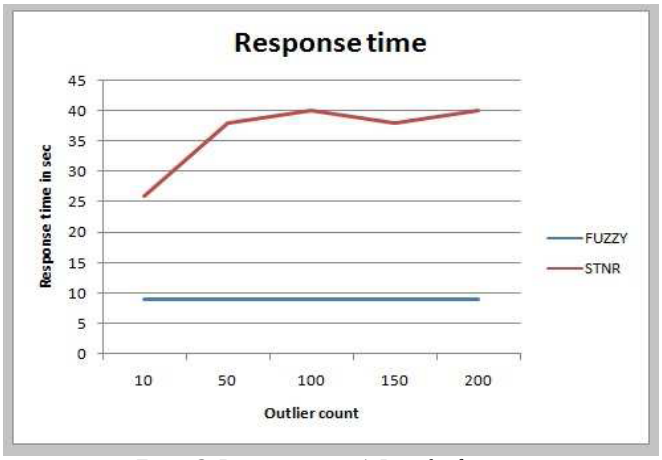

Figure 3: Response time v/s Length of string.

Response Time v/s Length of String

In this section, the response time of STNR as the increasing series length is compared with the response time of FUZZY time series system. The results are shown in figure 3. Here run the algorithm on series with same periods, same number of alphabets, and same number of outlier periodic patterns, while changing the series size and recorded the response time of the algorithm. The result is presented in fig. The response time changes in an increased fashion in the case of STNR but which is stable in the case of this fuzzy system. Both algorithms are run on the same series. It is checked for all periodic patterns with period p. The same is done for STNR; the results are presented as a graph. Although FUZZY takes significantly less time than STNR, the time performance of FUZZY gets steady.

V. CONCLUSION AND FUTURE WORK

In this paper, a novel algorithm for the detection of periodic outliers, surprising, or unusual patterns in a fuzzy time series is presented. The notion of a surprising or unusual pattern in a fuzzy time series takes into account the relative frequency of a pattern with patterns of similar length. The algorithm also takes into account the coverage area of the pattern and the likelihood of pattern occurrence to classify it as an outlier pattern. The algorithm can easily identify the fuzziness of the dataset and defuzzify in the second stage. It can also identify outlier patterns that may involve some frequent events, as it checks the repetitions of combination of events and not just the individual events. The experimental results show that the proposed algorithm consistently outperforms the existing approach STNR. Additionally, the outlier detection algorithm, being an extension of the STNR periodicity detection framework, can achieve things like detection of periodic patterns in subsections and works with noisy series containing any of insertion, deletion, and replacement noise. It is also shown that our STNR is not only time efficient but also space efficient. Finally, here currently working on the following aspect. The definition of surprising patterns can be further improved; some possibilities include the exclusion of user-specified minimum surprise value and integration of standard deviation in the definition of candidate outlier patterns. With this, the candidate outlier patterns might be defined as those which have less than repetitions.

REFERENCES

[1] Faraz Rasheed, Reda Alhajj, "A Framework for Periodic Outlier Pattern Detection in Time-Series Sequences "IEEE Trans on Cybernetics, VOL. 44, NO. 5, MAY 2014..

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

[2] Faraz Rasheed, Mohammed Alshalalfa, and Reda Alhajj, "Efficient Periodicity Mining in Time Series Databases Using Suffix Trees "IEEE Trans on knowledge and data engineering, VOL. 23, NO. 1, JANUARY 2011.
[3] Stephan Spiegel, Julia Gaebler, Andreas Lommatzsch, Ernesto De Luca, Sahin Albayrak, "Pattern Recognition and Classification for Multivariate Time Series"ACM Trans on Sensor KDD'11, AUGUST 21, 2011.
[4] Yueguo Chen, Ke Chen, and Mario A. Nascimento, "Effective and Efficient Shape-Based Pattern Detection over Streaming Time Series"IEEE Trans on knowledge and data engineering, VOL. 24, NO. 2, FEBRUARY 2012.
[5] Varun Chandola, Arindam Banerjee, Vipin Kumar,"Anomaly Detection for Discrete Sequences: A Survey "IEEE Trans on knowledge and data engineering,VOL. 24, NO. 5, MAY 2012.

BIOGRAPHY

Sreelakshmy M K was born in Kerala, India on November 28, 1990. She did her Bachelor of Technology in Computer Science and Engineering from Federal Institute of Science and Technology, Angamaly, Kerala in the year 2013. She is currently pursuing her Master of Technology in Computer Science and Engineering from Viswajyothi College of Engineering and Technology, Vazhakulam, Kerala.

Her field of interest lies in areas of data mining Right from the time she had been doing her bachelor's degree her focus was on building applications for data mining making use of the python language.