AN IMPROVED FRAMEWORK FOR OUTLIER PERIODIC PATTERN DETECTION IN TIME SERIES USING WALMAT TRANSACTION DATA

Sulochana M. Gagare
Department of Computer Enginnering Savitribai Phule Pune University
Vishwabharti College Of Engg.Ahmednagar
* sulochanagagare1234@gmail.com

Prof.S.B.Natikar
Department of Computer Enginnering Savitribai Phule Pune University
Vishwabharti College Of Engg.Ahmednagar
* sbnatikar5@gmail.com

ABSTRACT

Periodic pattern detection in time-series is one of the most important data mining task. The periodicity detection of an outlier patterns might be more important than the periodicity of regular, more frequent patterns means periodic pattern .periodic patterns means Patterns which repeat over a period of time. Pattern those which occur unusually or surprisingly called as Outlier Pattern. In this paper ,we present the development of a enhanced spatio-temporal algorithm capable of detecting the periodicity of outlier patterns in a time series using Walmart transaction data and MAD (Median Absolute Deviation) is presented. mean valuesis used in existing algorithm which is not efficient. We have to use MAD which increases the output of these algorithms and gives more accurate information.

INTRODUCTION

Data mining is a powerful knowledge discovery tool useful for modeling relation-ships and discovering hidden patterns in large databases [1]. Among four typical data mining tasks, predictive modeling, cluster analysis and association analysis outlier detection is the closest to the initial motivation behind data mining. [2].Outlier detection can be consider as widely researched problem in several knowledge disciplines, including statistics, data mining and machine learning. Outlier detection can be also known as anomaly detection, deviation detection, novelty detection and exception mining in some literature [3]Though all these definitions are different but all are identifying instances of unusual behavior when compared to the majority of observations. There are various definitions for an outlier,but no universally accepted definition exists. Two classical definitions of an outlier include Hawkins [4]and Barnett and Lewis [5]. According to the former, an outlier is an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism", where as the latter definition of an outlier is an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data". The term"outlier" is nothing but an observation that is significantly different from the other values in a data set. Outliers often occur due to following causes.

1.Error-The outliers occure due to error are also known as anomalies, discordant observations, exceptions, faults, defects, aberrations, noise, damage or contaminants. Human errors, instrument errors, mechanical faults or change in the environment are also causes of occurence of Outlier. Due to the fact that such outliers reduce the quality of data analysis and so may lead to erroneous results, they need to be identified and immediately discarded.

2. Event- outliers may be generated by a different mechanism, which indicates that outliers belong to unexpected patterns that do not show normal behavior and may include interesting and useful information about rarely occurring events within numerous application domains. Because of this reason such outliers would be identified for further investigation.

There are many applications of outlier detection such as :

1. Fraud detection -The purchasing behavior of people who steal credit cards may be different from that of the owners of the cards. The identification of such buying pattern could effectively prevent thieves from a long period of fraud activity. This approaches can also be used for other kinds of commercial fraud such as in mobile phones, insurance claim, financial transactions etc .

2.Intrusion detection [6]. System can be disabled due to frequent attacks on computer, even completely collapsing. The identification of such intrusions could find out malicious programs in computer operating system and also detects unauthorized access with malicious to computer network systems and so effectively keep out hackers.

3.Environmental monitoring-There are many unusual events occur in the natural environment such as a typhoon, flooding, drought they often have an adverse impact on the normal life of human beings. The identification of certain typical behaviors could accurately predict the likelihood of these phenomena and allow people to take effective measures on time.

4.Medical [7]. Patient records with unusual symptoms or test results may indicates potential health problems for a particular patient. The identification of such unusual records could distinguish instrumentation or recording errors from whether the patient really has potential diseases and so take effective medical measures in time.

5.Localization and tracking-Localization refers to the determination of the location of an object or a set of objects. The collection of raw data can be used to calibrate and localize the nodes of a network while simultaneously tracking a moving target. It is a known fact that raw data may contain error, which make localization results not accurate and useful. Filtering such erroneous data could improve the estimation of the location of objects and make tracking easier.

LITERATURE SURVEY

In this section we present background and related work of this domain. Specifically, There are several algorithms that discover the frequent periodic patterns having (user specified) minimum number of repetitions or with minimum confidence (ratio between number of occurrences found and maximum possible occurrences),e.g., [10], and However, not much work has been done for periodicity detection of outlier patterns. It is important to note that surprising, unusual, or outlier patterns are different from outlier (values) in the data. There are many techniques to find local and global outliers in the data, but outlier or surprising patterns are different from others patterns. For example, in a certain sequence, events a and event b might not be outliers but the pattern aba (a certain combination of the events) might bean outlier pattern. There are few algorithms, e.g., [13] and which discover the surprising patterns in time series.

Keogh et al. [13] presented their suffix tree-based algorithm to mine surprising patterns. Their algorithm requires the user to supply training purpose regular series. Patterns in the test data and the training data are compare and those having different expected values are qualified as surprising patterns. Since the algorithm requires the training data,it might not be possible in many cases to define the regular data; secondly, surprising patterns are discovered by the algorithm.which are not necessarily the periodic patterns.

Yang et al. have presented their so-called InfoMiner algorithm[6] and its variations which discover what they call the surprising periodic patterns. They define the measere of surprise using their notion of information gain which gives more significance to patterns involving lesser frequent events and having more support(matching repetition).

Sheng et al. [7] presented their algorithm which is based on Hans [5] partial periodic patterns algorithm,detect periodic patterns in a section of time series, and utilizes the optimization steps to find the dense periodic areas in the time series. However, their algorithm, being based on Hans algorithm, requires the user to provide the maximum period value. We argue that the maximum period value is difficult to be defined by the user and which may lead toward missing some interesting periodic patterns.

Huang and Chang [10] presented their algorithm for finding asynchronous periodic patterns, where the periodic occurrences can be shifted in an allowable range within the time axis. This
is very similar to how we deal with the noisy data by utilizing the time tolerance window for the periodic occurrences

PROPOSED SYSTEM

A. EXISTING SYSTEM

The existing method presented a novel algorithm for the periodicity detection of outlier, surprising, or unusual patterns. It present a robust and time efficient suffix tree-based algorithm capable of detecting the periodicity of outlier patterns in a time series gives more significance to less frequent yet periodic patterns. Many experiments have been conducted using real and synthetic data. 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. This definition is not limited to the assumption that unusual patterns are not the patterns involving less frequent events as described in and nor it requires the training/testing phase. As definition, they can also identify outlier patterns that may involve some (or all) frequent events, as it check the repetitions of combination of events and not just the individual events.

B.PROPOSED SYSTEM

Researchers have mostly investigated time series to identify repeating patterns and some researchers studied exceptional patterns (outliers) in time series. We concentreate in this paper, on the first case; we need to develop an algorithm capable of detecting in an encoded time series (even in the presence of noise) symbol, sequence, and segment periodicity, which are formally defined next. We start by defining confidence because it is not always possible to achieve perfect periodicity and hence we need to specify the degree of confidence in the reported result.

Perfect Periodicity: Consider a time series T, a pattern X is said to satisfy perfect periodicity in T with period p if starting from the first occurrence of X until the end of T every next occurrence of X exists away from the current occurrence of X. Imperfect periodicity.occure when some of the expected occurrences of X missing.

Confidence: X is the confidence of a periodic pattern occurring in time series T is
actual periodicity /expected perfect periodicity. Formally, the X with periodicity p starting at position stPos is defined as: conf (p; stpos ;X) = Actual Periodicity(p; stPos; X)
Perfect Periodicity(p; stpos; X)
where Perfect Periodicity(p; stpos, X)=(|T|-stpos+1)/p;

I.PATTERN MINING

We have data is in the form of patterns in Time series databases. The patterns can be mined for the prediction the future data. For example, consider the time series containing the hourly number of transactions in a Walmart's store; the discretization process may define the following mapping by considering different possible ranges of transactions; A=0-100,B=101-200,C=201-300,D=301-400. Based on this mapping, the time series T= 243; 267; 355; 511; 120; 0; 0; 197 can be discretized into T= cccdbaab.

II. PERIODICITY MINING

From the generated suffix trees we are performing periodicity mining. By this method, we take the difference between any two successive occurrence vector elements leading another vector called the difference vector. For Mining periodicity we need to remove the noise from the data sequence. Here we need to remove the redundancy also. This supposed give the perfect periodic mining.

A. PERIODITICITY DETECTION

Our algorithm involves two cases. In the first case, we build the suffix tree for the time series and in the second case, we use the suffix tree to calculate the periodicity of various patterns in the time series. One important aspect of our algorithm is redundant period pruning. The algorithm does not waste time to investigate a period which has already been identified as redundant due to redundant pruninig. This saves considerable time and also results in reporting fewer but more useful periods. This is the primary reason why our algorithm, intentionally, reports significantly fewer number of periods without missing any existing periods during the pruning process.

III. SUFFIX TREE BASED REPRESENTATION

Suffix tree is a data structure that has been proven to be very useful in string processing. It can be 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; there is a distinguished path from the root to a corresponding leaf node for each suffix of the string in the suffix tree. Given that a time series is encoded as a string, the most important aspect of the suffix tree, related to our work, is its capability capture and highlight the repetitions of substrings within a string.

Suffix for the string is the path from the root to any leaf .A string of length n can have exactly n suffixes, the suffix tree for a string also contains exactly n leaves. Each edge contain a label of the string that it represents. Each leaf node consists a number that represents the starting position of the suffix yield when traversing from the root to that leaf. Each intermediate node can have a number which is the length of the substring read when traversing from the root to that intermediate node.The string which is repeated at least twice in the original string is read by Each intermediate edge from root to that edge.

When the tree is constructed, we traverse the tree in bottom-up order to construct occurrence vector for each edge connecting an internal node to its parent.The nodes which having only leaf nodes as children we start from that node, this each node passes the values of its leaf node to the edge connecting it to its parent node. The latter edge used this values to create its occurrence vector. Then we consider leaf and non leaf nodes as children. Finally, we recursively consider each node u having only non leaf children until we reach all direct children of the root. Applying this bottom-up traversal process on the suffix tree shown in fig1 it will produce the occurrence vectors reported in Fig. 2

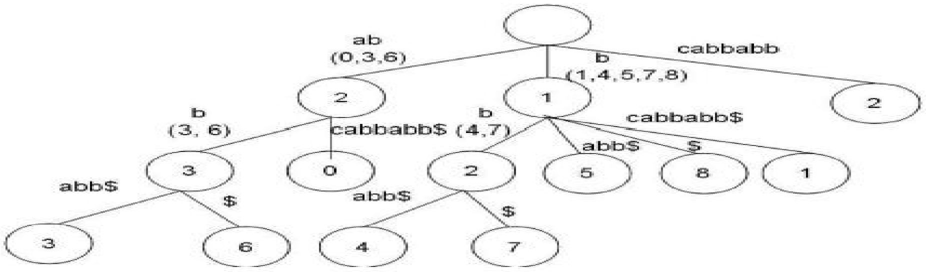

Fig. 2. Suffix tree for string *abcabbabb$* after bottom-up traversal.

To check whether the string represented by the edge is periodic the periodicity detection algorithm is use the occurrence vector of each intermediate edge (an edge that leads to a nonleaf node).The non recursive explicit stack-based algorithm is used to implement the tree traversal process which prevents the program from the stack-overflow-exception.

Spatio Temporal Algorithm

1. Create two time series from the sequence of previous visits: (the time series of the visit daily start times C and the time series of the visit durations D defined as follows:)

C = (c_1 to c_n)

D = (d_1 to d_n)

where ci is the time of the day in seconds corresponding to the time instant ti (i.e., c_i is in the range [0; 86400]);

2.Search in the time series C sequences of consecutive values ($c_{(i)) [m + 1]}$; : : : ; $c_{(i)}$) that are closely similar to the last m values ($c_{(n)[m + 1]}$; : : : ; $c_{(n)}$);

3.Estimate the next value of time series C by averaging all the values $c_{(i+1)}$ that follow each found sequence;

4. Elect Corresponding sequences D ($d_{(i) [m + 1]}$; : : : ; $d_{(i)}$) and Locate the sequences exactly at the same indexes in C.

5. The next value of time series D is then estimated by averaging all the values $d_{(i+1)}$ that follow these sequences.

We can process this algorithm to predict not only the next visit to a location, but also successive visits in the future: in fact, we can choose average together not only the next values of each sub sequences but also values that are 2 or more steps ahead.

However, the prediction of time series can become inaccurate when adopted to calculate further values in the future. Since we can predict when the future visits to all significant locations will start and for how long they will last.To predict the location where the user will be at a given time in the future we can design a simple method. Let us suppose that at time T we want to predict in which significant location user i will be after T seconds. Then, we have to perform below steps:

1. The sequence of the next k visits (starting with k = 1) are predicted for each location and a global sequence of all predicted visits (loc$_1$; t$_1$; d$_1$); : : : ; (loc$_n$; t$_n$; d$_n$) is created, with t$_1$.......t$_n$;

2. if there is a prediction (loc$_i$; t$_i$; di) which satisfies t$_i$,T + T, ti + di, then loc$_i$ is returned as predicted location (in case several predictions exist which satisfy the predicate, we choose at random between them);

3. there are two cases if no prediction satisfies the condition stated above: if the minimum start time t$_1$ of the current predicted visits is smaller than T + T, then prediction needs to be extended further in the future in order to find a suitable visit, thus the parameter k is doubled and the algorithm is repeated considering new predicted visits. Otherwise, visits which start after T + T provides extending the prediction and which cannot be exploited for prediction: thus, the algorithm terminates returning that the user will not be in any significant location.

One thing that we have to note- it is realistic for a user to be predicted as being outside the set of significant places (e.g., maybe transitioning from one to another) and that our technique is also able to predict this state.

MATHEMATICAL MODELLING

1.USE OF MEDIAN ABSOLUTE DEVIATION

The existing algorithm uses mean value to find out outliers. But this method is not efficient for various reasons, this reasons are explained in the paper //

1.MEAN

Mean of data set can be calculated using below formula:

$$\mu = \sum_{i=1}^{n} y_i \ / \ n$$

It is the average value of any given data set. The reasons why it is considered as robust estimator are as follows: 1)Mean value is highly biased even if there is a single outlier and 2) a mean value can be changed in a large data set seven though an outlier is removed. So, when we used a mean value to detect an outlier an outlier can be considered as a normal data point. This reduces the efficiency of the method and makes it a non robust estimator.

2.The Median Absolute Deviation (MAD)

The MAD overcomes these problems. To calculate the median,we have to sort observation in ascending order. Let us consider the previous series: 1, 3, 3, 6, 8, 10, 10. The average rank = (n + 1) / 2 . The median is between the fourth and the fifth , that is, between six and eight means it is seven. The MAD involves calculating the median of absolute deviations from the median. the Median Absolute Deviation is defined as follows :

$$MAD = b \ M_i\left(\left|x_i - M_j\left(x_j\right)\right|\right) \quad (2)$$

Where the xj = n original observations and Mi =median of the series. b = 1.4826, assumption of normality of the data, disregarding the abnormality induced by outliers .

5. Data Set And Result

The algorithm is applied on the time series data set. The data values in data set are used to calculate MAD values. This calculated MAD value is used to determine surprising values from the given data set by comparing them with the MAD. Values which are 3 times away from the Median values are considered as Outliers.

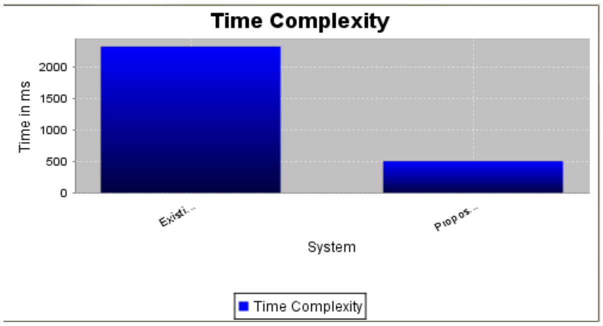

Fig3.Time Complexity of Existing System and Proposed System

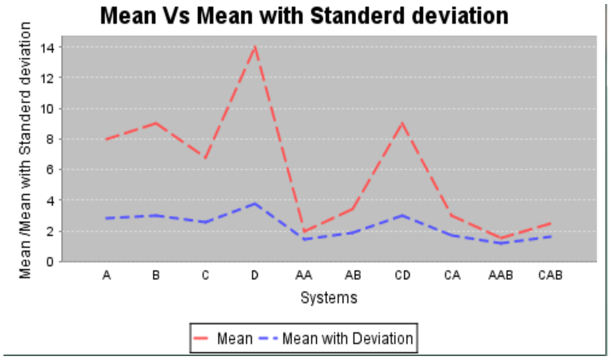

Fig4.Accuacy

CONCLUSION AND FUTURE WORK

With this definition, it can also identify outlier patterns that may involve some (or all) frequent events, as it check 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 InfoMiner. Additionally a novel algorithm for the periodicity detection of outlier, surprising, or unusual patterns is shown. It makes use of the MAD value to compare relative frequency of the outlier pattern instead of mean value which was previously
used in the existing algorithm. As the mean method is not robust and do not give the accurate results. It can easily be affected with the presence of outlier. A new measure known as Median Absolute Deviation is used to detect outlier instead of mean, as it is more efficient compare to mean. It increases the accuracy of the existing algorithm. In the carried out experiments outliers detected by MAD are more accurate.

ACKNOWLEDGMENT

This paper work is completed successfully only because support from each and every one including teachers, colleague, parents and friends. Especially, I am very thankful to those
who provide me guidance and make this work reachable.
My senior, my teachers and some experienced personalities helped me to complete this paper continusaly.My acknowledgment of gratitude toward my project guide Prof.S.B.Natikar
who make this work reachable. I express my gratitude to Mr. Prabhudev, ME[Computer Engg]Co-ordinator Mrs. Sarika Joshi, Head of Computer Department, for their constant encouragement, co-operation and support. I would like to thank all faculty members, Prof.M.C.Kshirsagar ,Mrs. Jagtap Madam of Computer Engineering Department.I would also like to thank my husband Mr. Prasad Kadam and my family members to their support.

REFERENCES

[1] E. Keogh, J. Lin, and A. Fu, "HOT SAX: Efficiently finding the most unusual time series sub-sequence," in Proc. IEEE Int. Conf. Data Mining, Houston, TX, USA, Nov. 2005, pp. 226–233.
[2] N. Kumar, N. Lolla, E. Keogh, S. Lonardi, C. A. Ratanamahatana, and L. Wei, "Time- series bitmaps: A practical visualization tool for working with large time series databases," in Proc. SIAM Int. Conf. Data Mining, Newport Beach, CA, USA, 2005, pp. 531–535.
[3] M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid, "Periodicity detection in time series databases," IEEE Trans. Knowl. Data Eng., vol. 17, no. 7,pp. 875–887, Jul. 2005.
[4] M. G. Elfeky,W. G. Aref, and A. K. Elmagarmid, "WARP: Time warping for periodicity detection," in Proc. IEEE Int. Conf. Data Mining, Nov. 2005, pp. 8–15.
[5] J. Han, W. Gong, and Y. Yin, "Mining segment-wise periodic patterns in time related databases," in Proc. ACM Int. Conf. Knowl. Discov. Data Mining, vol. 8, no. 1, pp. 53–87, Aug. 1998.
[6] J.Yang,W.Wang, and P.Yu, "InfoMiner+: Mining partial periodic patterns with gap penalties," in Proc. IEEE Int. Conf. Data Mining, Dec. 2002, pp. 725–728.
[7] C. Sheng,W. Hsu, andM.-L. Lee, "Mining dense periodic patterns in time series data," in Proc. IEEE Int. Conf. Data Eng., 2005, p. 115.
[8] C. Sheng, W. Hsu, and M.-L. Lee, "Efficient mining of dense periodic patterns in time series," Nat. Univ. Singapore, Singapore, Tech. Rep. 1, 2005.

[9] J. Han, Y. Yin, and G. Dong, "Efficient mining of partial periodic patterns in time series database," in Proc. IEEE Int. Conf.Data Eng., 1999, pp. 106–115.

[10] K.-Y. Huang and C.-H. Chang, "SMCA: Ageneral model for mining asynchronous periodic patterns in temporal databases," IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 774–785, Jun. 2005.

[11] F. Rasheed, M. Alshalalfa, and R. Alhajj, "Efficient periodicity mining in time series databases using suffix trees," IEEE Trans. Knowl. Data Eng., vol. 23, no. 1, pp. 79–94, Jan. 2011.

[12] F. Rasheed, M. Alshalalfa, and R. Alhajj, "Adaptive machine learning technique for periodicity detection in biological sequences," Int. J. Neural Syst., vol. 19, no. 1, pp. 11–24, 2009.

[13] E. Keogh, S. Lonardi, and B. Y.-C. Chiu, "Finding surprising patterns in a time series database in linear time and space," in Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2002, pp. 550–556.

[14] J. Yang, W. Wang, and P. S. Yu, "Infominer: Mining surprising periodic patterns," in Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2001, pp. 395–400.

[15] J. Yang, W. Wang, and P. S. Yu, "STAMP: On discovery of statistically important pattern repeats in long sequential data," in Proc. SIAM Int. Conf. Data Mining, 2003, pp. 224–238.