THIS IS MY FIRST TEST TITLE

A Thesis
Submitted to the Faculty

of

National Sun Yat-sen University

by

Cheng-An Fu

In Partial Fulfillment of the
Requirements for the Degree

of

Master of Science

September 2017

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

Introduction

Data mining is the process of discovering new patterns from large datasets. It finds useful information from data in databases [6]. Periodicity mining is used for predicting trends in time series data [12, 9, 5]. Applications of periodicity mining in time series data include temperature, stock prices depicted in the financial market, gene expression data analysis, etc [7, 10]. In general, there are three types of periodic patterns which can be detected in the time series data [6]: (1) symbol periodicity, (2) sequence periodicity or partial periodic patterns, and (3) segment or full-cycle periodicity. (Note that the time series data is mostly discretized before it is analyzed.)

CHAPTER II

Related Work

Some applications require that periodic patterns must satisfy two extra conditions including *Min_count* and *Max_distance* [3]. Parameter *Min_count* is used to limit that a pattern needs to repeat itself at least a certain number of times to demonstrate its significance and periodicity. Parameter *Max_distance* is to limit that the distance between the same two patterns has to be within some reasonable bound [1, 8, 4].

2.1 Ch2 Ch2

Usually, *Min_count* is set to be a value greater than or equal to 2, and *Max_distance* is set to be $N/2$, where N is the length of the time series data. To provide accurate period patterns for some applications, the length of the period pattern may be limited [9]. Moreover, to provide efficient processing, some pruning techniques are preferred [8].

2.1.1 Related

For the example of the time series data *abcabdabe*, both patterns *ab* and a start in positions 0, 3, 6. In this case, we usually prefer only a pattern *ab*. That is, a pattern *a* is pruned since it is the subset of a patter *ab* in the same starting position. Similarly, a pattern *b* is pruned since a pattern *b* is the subset of patter *ab* in the same ending positions.

2.1.1.1 Work

In the past of the structure, the suffix tree is a compact version of the suffix tries [12]. A suffix tree of length m is a tree with the following properties: (1) Each tree edge is labelled by a substring of S. (2) Each internal node has at least 2 children. (3) The number of leaves is m. (4) Each suffix has its unique leaf. Furthermore, it is well established in exact string matching and a good introduction to use in biology (for the example, in constructing and searching the DNA sequence) [8].

2.2 Related Work

A suffix tree indexes a string of length N which means that it has N leaf nodes. When the size of the sequence database increases, the storage space of a suffix tree also increases. To reduce the storage space, the suffix array is proposed which is basically a sorting list of all the suffixes of strings and only the sorting list is stored [11]. The main advantage of the suffix array over the suffix tree is that the suffix array uses three to five times less space.

CHAPTER III

Method

In this section, we present the time-position join (TPJ) method which improves the suffix tree method in the first phase of Rasheed *et al.*'s approach.

3.1 Ch3 Ch3

In Phase 1 of Rasheed *et al.*'s approach for periodicity mining in time series databases, it uses a revised version of the suffix tree as the data structure to generate candidate patterns.

3.2 Method

For the input *abcabbabb$* as shown in Figure 3.1, it needs to construct a suffix tree of 14 nodes as shown in Figure 3.2. However, in fact, patterns ab(0, 3, 6), abb(3, 6), b(1, 4, 5, 7, 8), bb(4, 7) are candidate patterns.

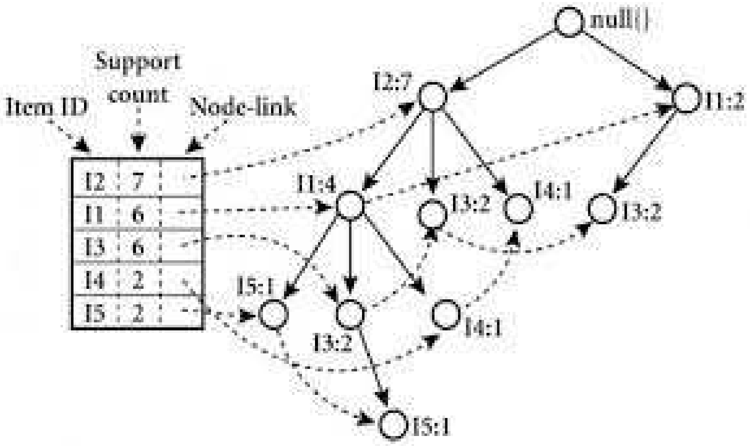

Figure 3.1 The suffix tree for the string *abcabbabb$*

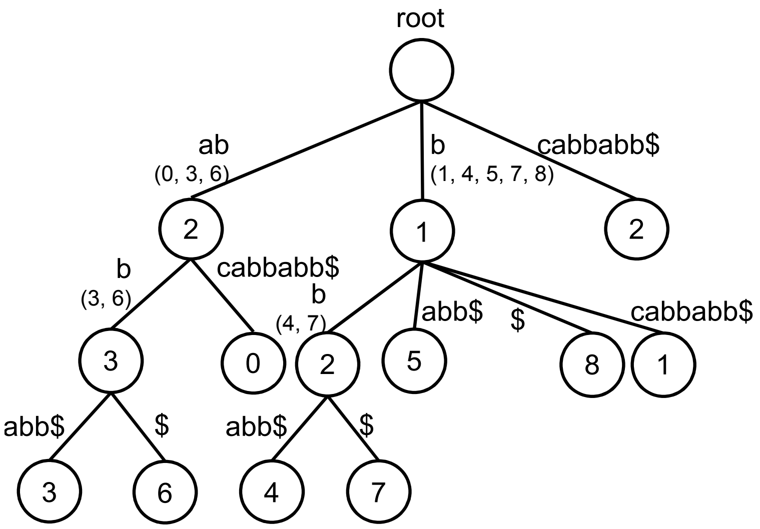

Figure 3.2 Suffix tree for string *abcabbabb$* after bottum-up traversal

CHAPTER IV

Performance

In this section, we compare the performance of our time-position join with the Phase 1 of Rasheed *et al.*'s approach. Our experiments are performed on an Intel(R) Core(TM) i7 2.93GHZ CPU computer with 4GB memory, running Windows 7 Ultimate version, and coded in JAVA. We generate synthetic data by different parameters.

4.1 Ch4 Ch4

The parameters are controlled during data generation including the number of patterns, the length of patterns and threshold. During our experiments, we will change one parameter at a time. For the synthetic data, the parameter *Ps* means the the number of the patterns and the parameter *Ls* means the length of the patterns. Our performance measure is the processing time. Table 4.1 summarizes the abbreviations for all electrode positions.

4.1.1 Apriori Algorithm

Starting from the root node, the subset function finds all the candidates contained in a transaction t as follows. If we are at a leaf, we find which of the itemsets in the leaf are contained in t and add references to them to the answer set. If we are at an interior node and we have reached it by hashing the item i, we hash on each item that comes after i in t and recursively apply this procedure to the node in the corresponding bucket. For the root node, we hash on every item in t.

To see why the subset function returns the desired set of references, consider what happens at the root node. For any itemset c contained in transaction t, the first item of c must be in t. At the root, by hashing on every item in t, we ensure that we only ignore itemsets that start with an item not in t. Similar arguments apply at lower depths. The only additional factor is that, since the items in any itemset are ordered, if we reach the current node by hashing the item i, we only need to consider the items in t that occur after i [2].

4.2 Performance

First, we set $Ps = 1000$, $Ls = 10000$ with threshold between 3 and 7. Figure 4.1 shows the comparison of the processing time with different threshold. From Figure 4.2, we show that as the threshold increases, the processing time of our method is faster than Rasheed *et al.*'s approach. The reason is that we can prune some patterns when the count does not satisfy the threshold in the modified adjacent matrix.

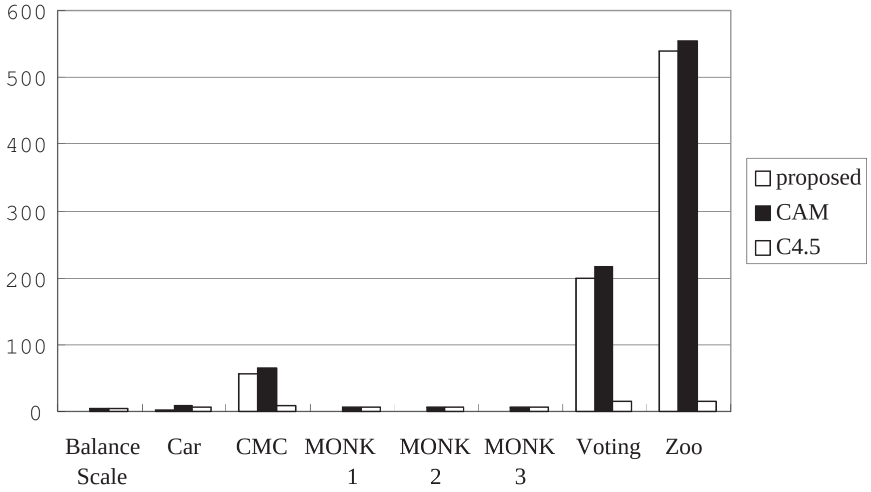

Figure 4.1 Scansone

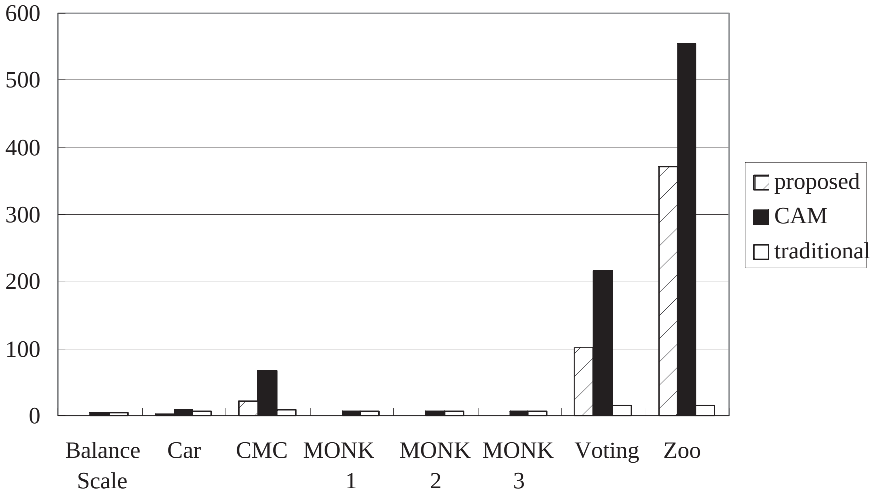

Figure 4.2 Scanstwo

Table 4.1 Apriori Sample

TID	**Items**
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

CHAPTER V

Conclusion

The time series data is a collection of data values generally at uniform interval of time to reflect certain behavior of an entity. Moreover, identifying periodic patterns can reveal important observations about the behavior and future trends of the case representation by the time series data. In this paper, we have proposed the time-position method to mine the periodicity patterns. We have avoided to traverse some paths repeatedly in traversing the graph step. From our performance study, we have shown that our method is more efficient than the Rasheed *et al*'s approach.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] A. A. Abdul-latif, I. Cosic, D. K. Kumar, B. Polus, and C. D. Costa, "Power Changes of EEG Signals Associated with Muscle Fatigue: The Root Mean Square Analysis of EEG Bands," *Proceedings of Intelligent Sensors, Sensor Networks and Information Processing Conference*, pp. 531–534, Dec. 2004.

[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," *Proceedings of the 20th international conference on Very Large Data Bases*, pp. 478–499, Sept. 1994.

[3] A. Anwar, "An Entropy-based Feature in Epileptic Seizure Prediction Algorithm," *IOSR-Journal of Computer Engineering*, Vol. 17, No. 6, pp. 47–54, Dec. 2015.

[4] J. A. Coan and J. J. Allen, "Frontal EEG Asymmetry and the Behavioral Activation and Inhibition Systems," *Society for Psychophysiological Research*, Vol. 40, No. 1, pp. 106–114, Feb. 2003.

[5] M. Mohammadi, F. Al-Azab, B. Raahemi, G. Richards, N. Jaworska, D. Smith, S. de la Salle, P. Blier, and V. Knott, "Data Mining EEG Signals in Depression for Their Diagnostic Value," *BMC Medical Informatics and Decision Making*, Vol. 15, No. 1, pp. 1–14, Dec. 2015.

[6] I. Mporas, V. Tsirka, E. I. Zacharaki, M. Koutroumanidis, M. Richardson, and V. Megalooikonomou, "Seizure Detection Using EEG and ECG Signals for Computer-based Monitoring, Analysis and Management of Epileptic Patients," *Expert Systems with Applications*, Vol. 42, No. 6, pp. 3227–3233, April 2015.

[7] M. Z. Parvez and M. Paul, "Epileptic Seizure Detection by Analyzing EEG Signals Using Different Transformation Techniques," *Neurocomputing*, Vol. 145, pp. 190–200, Dec. 2014.

[8] L. Patnaik and O. K. Manyam, "Epileptic EEG Detection Using Neural Networks and Post-classification," *Computer Methods and Programs in Biomedicine*, Vol. 91, No. 2, pp. 100–109, Aug. 2008.

[9] E. Pippa, E. I. Zacharaki, I. Mporas, V. Tsirka, M. P. Richardson , M. Koutroumanidis, and V. Megalooikonomou, "Improving Classfication of Epileptic and Non-epileptic EEG Events by Feature Selection," *Neurocomputing*, Vol. 171, pp. 576–585, Jan. 2016.

[10] A. Subasi and M. I. Gursoy, "EEG Signal Classification Using PCA, ICA, LDA and Support Vector Machines," *Expert Systems with Applications*, Vol. 37, No. 12, pp. 8659–8666, Dec. 2010.

[11] A. Tzallas, M. Tsipouras, and D. Fotiadis, "Epileptic Seizure Detection in EEGs Using Time-Frequency Analysis," *IEEE Trans. on Information Technology Biomed*, Vol. 13, No. 5, pp. 703–710, Sept. 2009.

[12] P. Valenti, E. Cazamajou, M. Scarpettini, A. Aizemberg, W. Silva, and S. Kochen, "Automatic Detection of Interictal Spikes Using Data Mining Models," *Journal of Neuroscience Methods*, Vol. 150, No. 1, pp. 105–110, Jan. 2006.

Appendixes