An Efficient Method for Estimating Time Series Motif Length Using Sequitur Algorithm

Nguyen Ngoc Phien[1,2(✉)]

[1] Center for Applied Information Technology,
Ton Duc Thang University, Ho Chi Minh City, Vietnam
nguyenngocphien@tdtu.edu.vn
[2] Faculty of Information Technology, Ton Duc Thang University,
Ho Chi Minh City, Vietnam

Abstract. Motifs in time series are approximately repeated subsequence found within a long time series data. There are some popular and effective algorithms for finding motif in time series. However, these algorithms still have one major weakness: users of these algorithms are required to select an appropriate value of the motif length which is unknown in advance. In this paper, we propose a novel method to estimate the length of 1-motif in a time series. This method is based on GrammarViz, a variable-length motif detection approach which has Sequitur at its core. Sequitur is known as a grammar compression algorithm that is able to have enough identification not just common subsequences but also identify the hierarchical structure in data. As GrammarViz, our method is also based on the Sequitur algorithm, but for another purpose: a preprocessing step for finding motif in time series. The experimental results prove that our method can help to estimate very fast the length of 1-motif for some TSMD algorithms, such as Random Projection.

Keywords: Motif detection · Sequitur algorithm · Grammar inference
Motif length · Time series

1 Introduction

Time series (TS) data plays an important role in many fields of life, such as securities, health, communications, financial data, astronomical data, weather, and environmental pollution levels. People have paid great attention to the problem of discovering motif subsequences. Motifs are usually seen but with unidentical arrangements of a longer TS. Motif which is extracted from a TS represented as one of the most remarkable patterns [5, 15].

Since the first work on time series motif discovery (TSMD) was given by Lin et al., 2002 [10], many researchers have proposed expressive algorithms for finding motifs. Some typical algorithms for TSMD can be listed as follow. Chiu et al. in 2003 [4] proposed Random Projection algorithm which utilizes locality-preserving-hashing in TSMD. Tanaka et al. in 2005 [16] proposed EMD algorithm which applies MDL principle. Gruber et al. in 2006 [6] proposed an algorithm by dynamic RBF network. Mueen et al. in 2009 [13] proposed MK algorithm, is an exact algorithm for TSMD.

L. Meng and Y. Zhang (Eds.): MLICOM 2018, LNICST 251, pp. 531–538, 2018.
https://doi.org/10.1007/978-3-030-00557-3_52

Castro and Azevedo in 2010 proposed a mutiresolution algorithm for TMMD [2]. However, many of them are limited: the requirement the length acknowledgement of the motifs in advance. However, there is an unavailability of those.

Recently in 2012, Li et al., proposed an approximate variable length TSMD algorithm, called GrammarViz [9], which is based on Sequitur algorithm [14]. Sequitur is known as a grammar compression algorithm that is able to have enough identify not just common subsequence but also identify the hierarchical structure in data. GrammarViz does not require users to provide the value of motif length. However, since Sequitur is an online and greedy algorithm, discovering motifs by GrammarViz are not optimal and most of them are relatively short. Therefore, GrammarViz still needs further enhancements to improve its performance.

In this paper, we propose a novel method to estimate the length of 1-motif in a TS. Similar to GrammarViz, our method is also based on the Sequitur algorithm, but for another purpose: a preprocessing step for TSMD. Our method is effective and efficient. The experimental results on six datasets show that our proposed method can help to estimate very fast the length of 1-motif for some motif detection algorithms, such as Random Projection or MK algorithm.

The rest of the paper is organized as follow. Section 2 presents some background and related works. Section 3 presents our proposed method for estimating the length of 1-motif in a time series. Section 4 describes experimental results. Section 5 gives some conclusions.

2 Background and Related Works

2.1 SAX and ESAX

TS are very high dimensionality and large noises. So we have to reduce time executing and space. The popular approach is to use Symbolic Aggregate Approximation (SAX) [11]. For a TS, it is divided into equal-sized segments by the PAA representation [7]. Next, for each value $\bar{C}_i$ in PAA, SAX maps it to a symbol based on a set of "breakpoints" which are the list of number under *N(0,1)* Gaussian curve.

Lkhagva et al. in 2006 [12] proposed another discretization method, called ESAX (Extended SAX), which is an extended variant of SAX. Since the SAX method is based on the PAA to reduce the number of dimensionality, resulting in the loss of some important data, and in financial-economic data, these losses are sometimes important data. So the ESAX representation adds two special points to complete the SAX, the smallest and largest points of each segment. Thus, each segment represented by a triple <mean, min, max> which is mapped to three symbols rather than one symbol as in SAX. Experimental results revealed that ESAX can work with higher accuracy than SAX.

2.2 Sequitur Algorithm

Sequitur, the algorithm proposed by Nevill-Manning and Witten in 1997 [15], is known as a grammar compression algorithm that is able to have enough identify

common subsequence and the hierarchical structure in data. Sequitur has applied in several areas, for example, looking for repetitive DNA sequences [3]. Sequitur generates grammar rules from a string based on repeated substring. Each repeated substring was replaced by a grammatical rule that produces a shorter original string. The Sequitur algorithm reads the input string and restructures the grammar rules to maintain the two following properties:

(i) Digram uniqueness: There is no pair of adjacent symbols appearing more than once in grammar.
(ii) Rule utility: All grammar rules (except the start rule) are used more than once.

Sequitur is an online algorithm, effective in terms of memory space and execution time, requiring $O(m)$ complexity to compress a string of size m [9].

For example, string *S1* = "*abcabdabcabd*" can be generated from the grammar rules shown in the Table 1.

Table 1. An example of Sequitur algorithm

Grammar rule	The string
S1-> BB	abcabdabcabd
A-> ab	ab
B-> AcAd	abcabd

The main advantages of Sequitur are three-fold: (i) it automatically detect repeated patterns, for example *"abcabd"* in the above example, and hierarchical structure; (ii) the grammar rules found may be any lengths and (iii) it is appropriate for streaming data.

3 Our Proposed Method for Estimating the Length of 1-Motif

Our proposed method for estimating the length of 1-Motif employs the GrammarViz approach for discovering TS motifs of variable lengths. But in our method, we use ESAX rather than SAX for discretization. Our method consists of four phases:

Phase 1: [Discretization] In this step, the original TS is discretized into a symbolic string by using ESAX transformation.

Phase 2: [Applying Sequitur algorithm on ESAX strings] After having transformed the TS into ESAX strings, we apply Sequitur on the ESAX strings to obtain the grammar rules.

Phase 3: [Post-processing] Since the original TS has been discretized before running the algorithm, we have to map the frequent subsequence back to the original TS. The amount of generated rules may be large and similar to association rule mining [1], so we do some refinements on the grammar obtained:

1. Eliminate trivial matches for a motif. The trivial match of a subsequence M is any sequence that overlaps M.
2. Arrange the rules according to "interestingness" such as the frequency of occurrence, and the length of the motif.

Phase 4: [Estimating the length of 1-motif] Based on the table storing all the induced grammar rules and the plots of all motifs, we can identify the region with the highest density of motif instances. Since this region contains the most significant motif (1-motif) in the TS, we will use it to estimate the length of the 1-motif. This phase consists of 3 steps:

1. Based on the plots of all motifs (one motif for each grammar rule) we can identify the region with the highest density of motif instances. And from our inspection on this region, we can determine the length n of the instances of the 1-motif. This length can be converted to w, the corresponding number of ESAX symbols.
2. Looking up at the table which stores all the grammar rules, we can find all the rules (motifs) which have the length approximately equal to w. We can also know the frequency of each such rule (motif).
3. In Step 1 we've already known the length n of the 1-motif. We can use this value as the length of 1-motif for the Random Projection algorithm.

To visualize all the motifs found by Sequitur, in Phase 3, for each found grammar rules found we need to record the length of the rule and its start position.

The proposed algorithm can estimate the length of 1-motif with high efficiency. The algorithm requires $O(n)$ to convert the TS into ESAX string, and then requires $O(n)$ to perform Sequitur algorithm.

4 Experimental Results

We tested the Sequitur-based method for estimating the length of 1-motif and the TSMD algorithm, Random Projection, MK. Random Projection is chosen in this experiment due to its popularity. It is the most cited time series discovery method up to date and is the basis of many current approaches that handle this problem. We implemented the two algorithms with Microsoft Visual Studio C# 2017 and conducted the experiments on an Intel Core ™i5-525U, CPU@1.6Ghz, 8G RAM PC. We used six datasets from the UCR TS Data Mining Archive [8] for the experiments. The datasets are from different areas. Their names are: ECG, EEG, MEMORY, POWER, STOCK and ERP. We use SAX as discretization method in Random Projection and ESAX in our proposed method.

We set the alphabet size a for SAX and ESAX to 6 and the size of PAA-segment to 10 and the size of *EPAA-segment* to 10.

4.1 Accuracy

In this section, we deduce the feasible of using Sequitur to estimate the length of 1-motif in TS. To prove that Sequitur induction can use to identify variable-length motifs,

we show an example from the ECG dataset (3500 data points). A part of the grammar rules generated by Phase 2 in our proposed method is shown in the Table 2. Each found grammar rule represents a motif, column 1 and column 3 in Table 2 records the frequency and the length of each rule, respectively.

Table 2. An example of grammar rules found.

Frequency	Rule	Length of motifs
0	R0 -> R1 R1 R2 R3 R4 R5 R6 R7 R6 R8 R9 R10 R11 R12 R13 R13 R14 R7 R14 R14 R14 R15 R16 R17 R17 R17 R18 R18 R19 R20 R21	1050
5	R1 -> e e	2
4	R2 -> R1 e	3
3	R3 -> R22 R22	4
5	R4 -> R23 R24	14
4	R5 -> R8 R8	32
2	R6 -> R5 R25	34
3	R7 -> R26 b R27 R28 b R29 R2 R30 R23 R4 R31	43
3	R8 -> R9 R9	16
6	R12 -> R21 R30 R1 R23 R3 R24	36
2	R13 -> R32 R32	24
7	R14 -> R32 R15	18
2	R16 -> R20 R12	41
3	R17 -> R33 R33	20
2	R18 -> R19 R16 R34 R35	210
2	R19 -> R36 R7 R35 R34	109
…	…	…

Figure 1 shows the plot of the ECG dataset. Figure 2 shows the plots of 37 motifs which correspond to 37 grammar rules found by Phase 2 in our proposed method on the ECG dataset. From the plots in Fig. 2, we can see that the region with the highest density of motif instances is around the starting part of the figure, and the span of this region is about 100-time points. Based on this observation, we determine the length of 1-motif as 100. Since the size of PAA-segment is 10, we can estimate the length w of 1-motif in ESAX symbols as about 10.

After determining the motif length w, we can execute Random Projection or MK to find the 1-motif in the ECG dataset with this parameter value. Random Projection discovered the 1-motif with 9 instances as illustrated in Fig. 3. By inspection, we can see that the shape of the motifs found in the starting part of Fig. 2 in the span of 100 time points is exactly the same as the shape of the 1-motif found by Random Projection shown in Fig. 3.

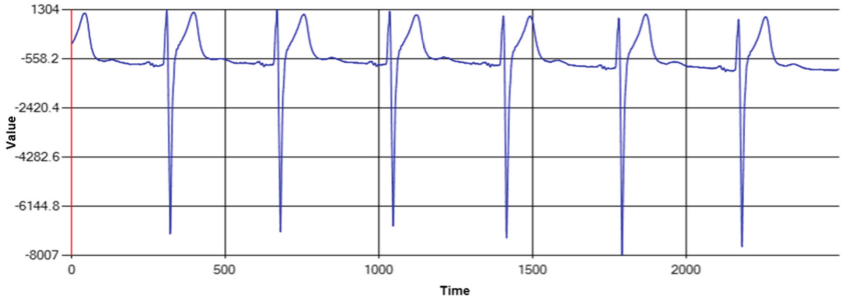

Fig. 1. The plot of the ECG time series

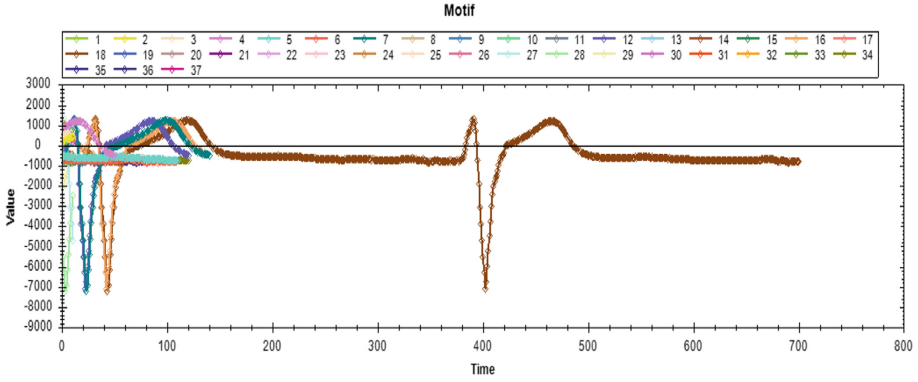

Fig. 2. All 37 motifs of different lengths found in ECG 3500 after Phase 2 in our method.

4.2 Efficiency

We measure the runtime of our proposed method for estimating the length of 1-motif and the runtime of Random Projection for discovering TS 1-motif. According to the Table 3 the runtime of our proposed method as a preprocessing step requires just a small percentage (about 4.6%) of the runtime for the Random Projection to find 1-motif in TS.

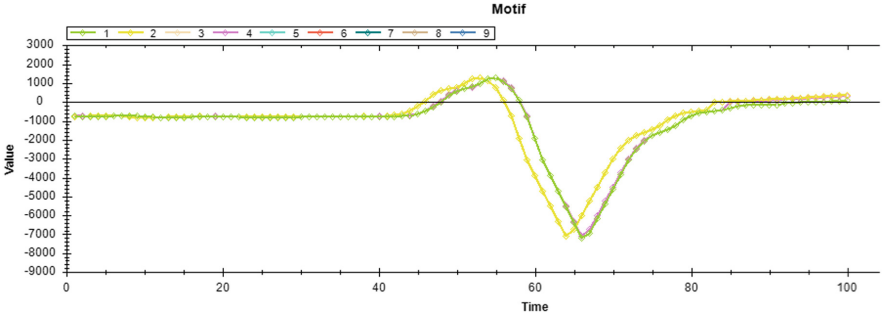

Fig. 3. Nine instances of the 1-motif found in ECG 3500 by Random Projection.

Table 3. Runtimes of our proposed method and Random Projection on 15 experiments.

No	Dataset	Length	Our method	Random projection
			Runtime (msecs)	
1	koski_ecg -3500	3500	13	240
2	koski_ecg -7000	7000	22	893
3	koski_ecg -10000	10000	43	1777
4	koski_ecg -14000	14000	68	3369
5	koski_ecg -20000	20000	98	4533
6	Stock1	5056	45	506
7	Stock2	5056	35	558
8	Stock3	5508	52	579
9	Stock4	5508	55	599
10	ERP	6622	62	803
11	ERP2	10120	94	1846
12	EEG	12137	66	2616
13	Memory1	13636	91	3342
14	Memory2	12000	53	2766
15	Power	12000	51	2612

5 Conclusion and Future Works

We presented a method for estimating the length of 1-motif in a TS which is based on Sequitur algorithm. Our method can be used as a preprocessing step for any TSMD algorithms which requires the length of 1-motif as an input parameter. The results on six datasets prove that our method may use to estimate very fast the length of 1-motif for some motif discovery algorithms, such as Random Projection. As for future work, we plan to apply our method in some real world applications of TS motif discovery.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., pp. 207–216, 26–28 May 1993
2. Castro, N., Azevedo, P.: Multiresolution motif discovery in time series. In: Proceedings of SIAM International Conference on Data Mining, 29 April–1 May, Columbus, Ohio, USA, pp. 665–676 (2010)
3. Cherniavsky, N., Ladner, R.: Grammar-based Compression of DNA Sequences. UW CSE Technical Report 2007-05-02 (2007)
4. Chiu, B., Keogh, E. Lonardi, S.: Probabilistic discovery of time series motifs. In: Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining (KDD 2003), pp. 493–498 (2003)
5. Fu, T.C.: A review on time series data mining. Eng. Appl. Artif. Intell. **24**(1), 164–181 (2011)

6. Gruber, C., Coduro, M., Sick, B.: Signature verification with dynamic RBF network and time series motifs. In: Proceedings of 10th International Workshop on Frontiers in Hand Writing Recognition (2006)
7. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. J. Knowl. Inf. Syst. **3**(3), 263–286 (2000)
8. Keogh, E.: The UCR Time Series Data Mining Archive (2018)
9. Li, Y., Lin, J., Oates, T.: Visualizing variable-length time series motifs. In: Proceedings of SDM (2012)
10. Lin, J., Keogh, E., Patel, P., Lonardi, S.: Finding motifs in time series. In: Proceedings of the 2nd Workshop on Temporal Data Mining, at the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2002)
11. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, CA, 13 June 2003
12. Lkhagva, B., Suzuki, K., Kawagoe, K.: Extended SAX: extension of symbolic aggregate approximation for financial time series data representation. In: Proceedings of DEWS Data Engineering Workshop, DNEWS 2006, 4A-i8 (2006)
13. Mueen, A., Keogh, E., Zhu, Q., Cash, S., Westover, B.: Exact discovery of time series motif. In: Proceedings of 2009 SIAM International Conference on Data Mining. SIAM (2009)
14. Nevill-Manning, C.G., Witten, I.H.: Identifying hierarchical structure in sequences: a linear-time algorithm. J. Artif. Intell. Res. **7**, 67–82 (1997)
15. Ratanamahatana, C.A., Lin, J., Gunopulos, D., Keogh, E., Vlachos, M., Das, G.: Mining time series data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, 2nd edn, pp. 1049–1077. Springer, Heidelberg (2010)
16. Tanaka, Y., Iwamoto, K., Uehara, K.: Discovery of time series motif from multi-dimensional data based on MDL principle. Mach. Learn. **58**(2–3), 269–300 (2005)