A Fast Time Series Rule Finding Based on Motif Searching

Tingting Dou[1,a,*], Haizhou Du[1,b] , Yuchen Mao[2,c] and Shaohua Zhang [1,d]

[1]School of Computer Science and Technology, Shanghai University of Electric Power
[2] School of Energy and Mechanical Engineering, Shanghai University of Electric Power
a. dttdoutingting@163.com, b. duhaizhou@shiep.edu.cn, c. maoyuchen5@126.com,
d. lanfengjin@126.com
*corresponding author

Keywords: Thermal power plants, Time series rule, Motif, Energy conservation.

Abstract. With the rapid economic development, people's demand for control of pollution emissions more and more intense. Thermal power plants must find ways to keep units running economically and efficiently, meet the minimum energy efficiency and emission standards and meet the environmental requirements. So we propose the algorithm of fast time series rule finding based on motif searching in this paper. We can use it to find what the reason is to achieve the optimal conditions of thermal power plants. What's more, the optimal time for the power plant units can be longer, the cost of the plant will be lower, and the goal of energy saving and emission reduction can be achieved. It has a guiding significance on the thermal power plant energy conservation and cost increasing.

1. Background

In September 3, 2016, China's National People's Congress (NPC) approved China's accession to the Paris Agreement on climate change. The agreement states that the global response to keep the global average temperature 2℃higher than the pre-industrial levels and make efforts to keep the temperature within 1.5 ℃. So thermal power plants need to achieve energy saving.

However, in the actual operation of thermal power plants, because thermal power units can't run at full capacity in long term, the load of thermal power units change frequently, which results in a serious deviation from the designed load of the power plant. As the optimal conditions are difficult to quickly retrieve large amounts of data, and the staff is difficult to know what causes the optimal conditions.

Some works proposed to the discovery of time series rule algorithm [1–6]. However, there is too many meaningless time series motifs and rules, these algorithms do not meet the needs of thermal power plants. We would like to know what causes are to achieve the optimal conditions of thermal power plants, rather than the future trend of thermal power plant time series. In view of these problems, we propose FTSRFMS algorithm in this paper.

The remaining part of this paper is organized as follows: Section 2 contains a series of notations and definitions which is needed in our time series algorithm. In Section 3, we introduce our method.

The experiments on FTSRFMS and visualization will be discussed in Sections 4, respectively. Section 5 offers concluding remarks and directions for future work.

2. Notations

We will give some definitions which are used in this paper before describing our algorithm. It is necessary to define the existing problems and explain our solution. Now we start with definitions of the time series:

Definition1: A *Time Series* $T = (t_1, t_2, \cdots, t_n)$ is an ordered set of real-valued numbers, the length of T is n.

$t_1, t_2, \cdots, t_n$ are data points of T which have the same time interval. A time series is often very large, sometimes it may contain billions of data. A local subsection of time series is termed as a subsequence. We need to define a distance metric to measure the distance between two subsequences. We use the ubiquitous Euclidean distance measure.

Definition3: The distance between two subsequences $S_{i,k}$ and $S_{j,k}$ is the Euclidean distance between $S_{i,k}$ and $S_{j,k}$. Note that both subsequences must be in the same length. It is:

$$Dis(S_{i,k}, S_{j,k}) = \sqrt{\sum_{l=0}^{k-1} (s_{i,k} - s_{j,k})^2} \qquad (1)$$

Euclidean distance is not sufficient to support cluster clustering for time series. However, it is still a useful subroutine to speed up our broader approach. As described in [7–10], the Euclidean distance is a fast and powerful distance measurement [11].

In this work, we use a version of the minimum description length (MDL) as a time series which was applied in some previous studies, such as [1, 12–15]. The MDL principle is described as follows:

Definition7: *Description Length DL*: a description of the time series T length DL is the total number of bits, it represents:

$$DL(T) = n \times H(T) \qquad (2)$$

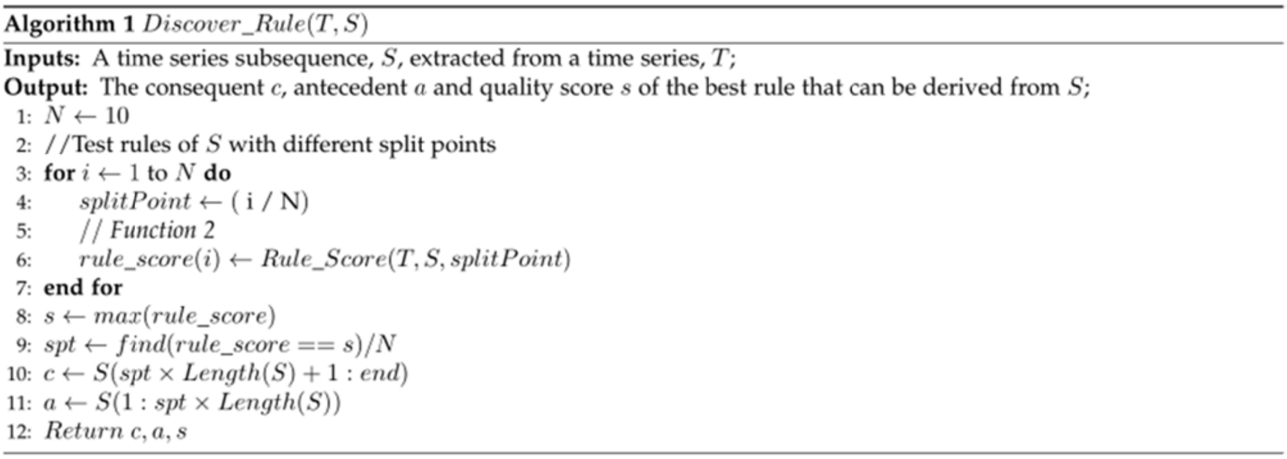
where n is the length of T. Note that a hypothesis H is a time series used to encode more time series of the same length [12].

3. Our Proposed Approach

The main idea of the algorithm of FTSRFMS is to match all the time series motifs from a single time series T, which is similar to the time subsequence S. We will create the rules from these motifs, calculate the bit save scores of these rules, and get the best time rule. Finally, we will find the corresponding antecedents based on these rules.

In lines 1 to 7 iterate over all possible split points of the time subsequence S, and calculate the quality score, and the quality score is described in the *Algorithm 2*. In line 8 calculates the highest quality score s. In lines 9 to 11 find the split point spt, which is corresponded to the highest quality score. The split point is used to split consequent C and antecedent a. Eventually, the Function returns C, a, and S.

In line 1 of *Algorithm 2* finds a set of subsequence in the time series T that are similar to the time subsequence S, which is described in *Algorithm 3* in detail. In line 2 uses the Euclidean distance to obtain a threshold of the distance, and then get the highest quality score of S described in detail in *Algorithm 4*. In line 3 the maximum number of bits, which is stored in discovered time rule, is calculated and eventually the value is returned as the quality score of the rule. It is described in *Algorithm 5*.

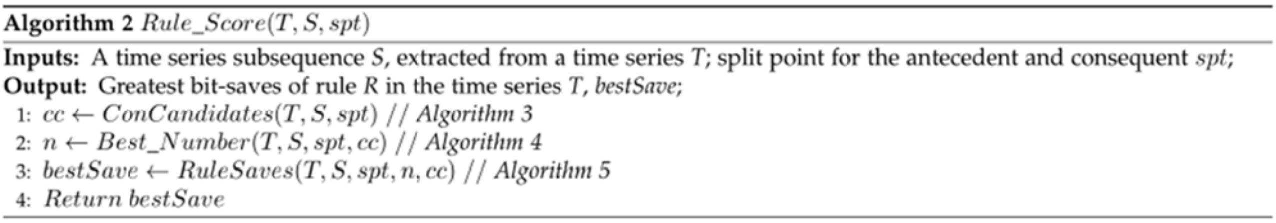


Algorithm 1 *Discover_Rule*(T, S)

Inputs: A time series subsequence, S, extracted from a time series, T;

Output: The consequent c, antecedent a and quality score s of the best rule that can be derived from S;

```
1:  $N \leftarrow 10$
2:  //Test rules of $S$ with different split points
3:  for $i \leftarrow 1$ to $N$ do
4:      $splitPoint \leftarrow$ ( i / N)
5:      // Function 2
6:      $rule\_score(i) \leftarrow Rule\_Score(T, S, splitPoint)$
7:  end for
8:  $s \leftarrow max(rule\_score)$
9:  $spt \leftarrow find(rule\_score == s)/N$
10: $c \leftarrow S(spt \times Length(S) + 1 : end)$
11: $a \leftarrow S(1 : spt \times Length(S))$
12: $Return\ c, a, s$
```

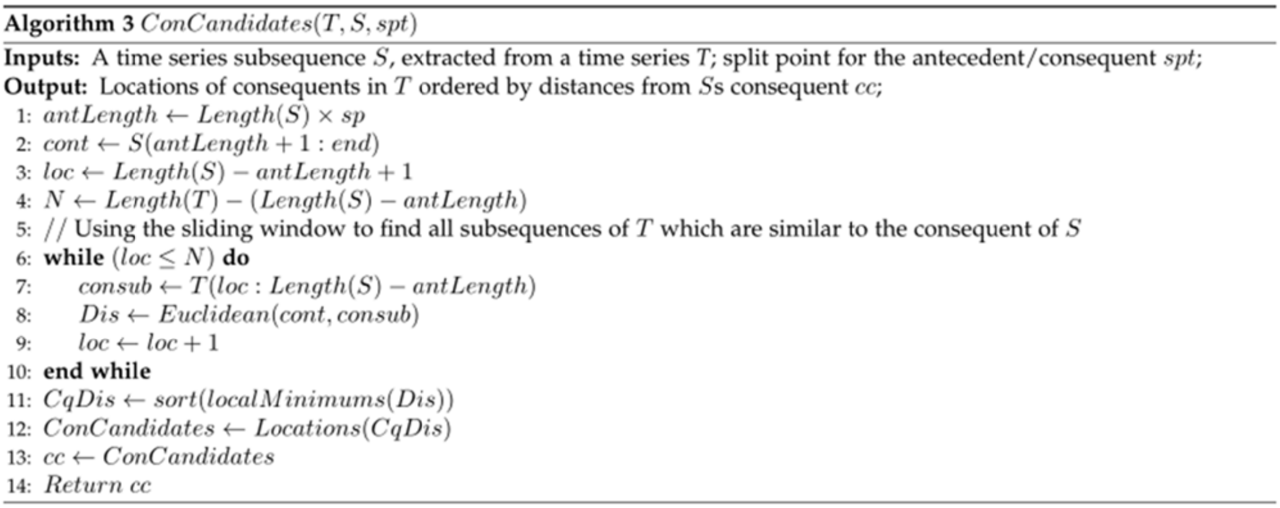


Algorithm 2 *Rule_Score*(T, S, spt)

Inputs: A time series subsequence S, extracted from a time series T; split point for the antecedent and consequent spt;

Output: Greatest bit-saves of rule R in the time series T, *bestSave*;

```
1: $cc \leftarrow ConCandidates(T, S, spt)$ // Algorithm 3
2: $n \leftarrow Best\_Number(T, S, spt, cc)$ // Algorithm 4
3: $bestSave \leftarrow RuleSaves(T, S, spt, n, cc)$ // Algorithm 5
4: $Return\ bestSave$
```



Algorithm 3 *ConCandidates*(T, S, spt)

Inputs: A time series subsequence S, extracted from a time series T; split point for the antecedent/consequent spt;

Output: Locations of consequents in T ordered by distances from Ss consequent cc;

```
1:  $antLength \leftarrow Length(S) \times sp$
2:  $cont \leftarrow S(antLength + 1 : end)$
3:  $loc \leftarrow Length(S) - antLength + 1$
4:  $N \leftarrow Length(T) - (Length(S) - antLength)$
5:  // Using the sliding window to find all subsequences of $T$ which are similar to the consequent of $S$
6:  while $(loc \leq N)$ do
7:      $consub \leftarrow T(loc : Length(S) - antLength)$
8:      $Dis \leftarrow Euclidean(cont, consub)$
9:      $loc \leftarrow loc + 1$
10: end while
11: $CqDis \leftarrow sort(localMinimums(Dis))$
12: $ConCandidates \leftarrow Locations(CqDis)$
13: $cc \leftarrow ConCandidates$
14: $Return\ cc$
```

In lines 1 and 2 we calculate the length of the antecedent of the time rule S, and get the consequent's sequence of the time subsequence S. In line 3 uses the sliding window to calculate the Euclidean distance of the same subsequence as an consequent length in the time series T. In line 11 the local minimum is found and sorted according to their distance. In line 12 we find the position of the local minimum distance after sorting in the time series T. In line 13 a set of consequent candidate sets are returned.

$Algorithm 4$ calculates the number of time rules in iteration. In lines 4 to 7 calculate the number of rule instances and the total bit save for each instance. If total bit save is not monotonically increasing, the iteration will be terminated. The time complexity of the function does not increase

```
Algorithm 4 $Best\_Number(T, S, spt, cc)$
Inputs: A time series subsequence $S$, extracted from a time series $T$; split point for the antecedent and consequent $spt$;
    Locations of antecedents in $T$ ordered by distances from $S$'s consequent $cc$;
Output: Best Number of instances of $S$ to pick in the time series $n$;
1: $talBS(1) \leftarrow 0$
2: $instans \leftarrow 1$
3: // If $talBS$ is not monotonically increasing, jump out of the loop
4: while ($talBS$ is monotonically increasing) do
5:     $insts \leftarrow insts + 1$
6:     $talBS(insts) \leftarrow Rule\_Bit\_Saves(T, S, spt, insts, cc)$
7: end while
8: $bestBits \leftarrow max(talBS)$
9: $n \leftarrow find(talBS == bestBits)$
10: Return $n$
```

greatly as the number of candidate time series rules increases. In line 8 we will choose the maximum total bit save. In line 9 we can find the number of instances.

So far, our algorithm has been introduced, and in the next section we will show section of our experiment.

4. Experiment

We have designed all our experiments to ensure that they are very easy to reproduce. We introduce our experimental data and analyse the results of our experiment.

We run FTSRFMS algorithm on a month of the data of Bns and find the rule shown in Figure 6. The split point of these time series rules is 0.1, total bit save is 506, there are 10 such rules in time series T in Figure 6. In Section 3 we have already mentioned that the higher the total bit score [1], indicating that the rules are credible. Below we will explain the results of Figure 6 in detail.

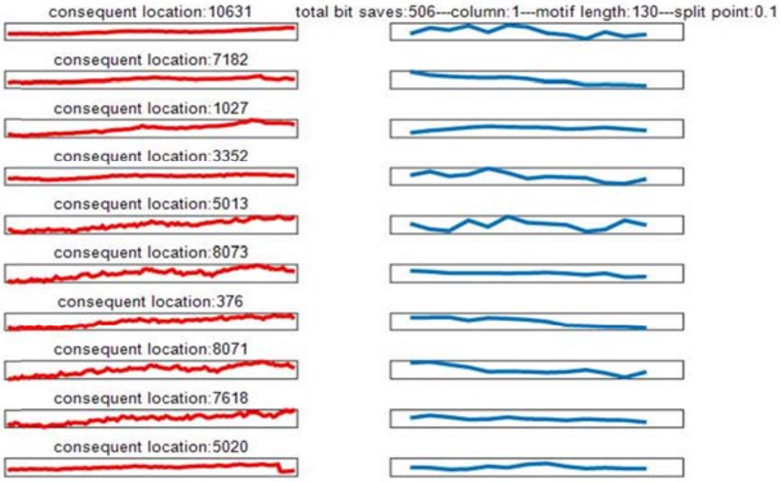


Figure 2. Time Series Rule of Bns, the split point of these time series rules is 0.1, total bit save is 506, there are 10 such rules in time series. The left is consequents of rules, and the right is antecedents of rules.

It is easy to know from Figure 2, the length of these antecedents is 117, values of Bns is in in the range (0.2957 Kw/h, 0.2983 Kw/h). Back to our original datasets, we can find that Tpe is in the range (587MW, 614MW), and $BTpy$ is in the range (131.5 $°C$, 132.7 $°C$) in the period of these antecedents in Figure 2. That is to say that the operator can control the two parameters (Tpe and $BTpy$) as far as possible within the range, the value of Bns will be as low as possible, and the rate of cost coal will be reduced in a certain period of time. In other words, the operating hours of the power plant units, which achieve the best conditions, will be longer.

5. Conclusion and Future Works

We have introduced the problem of discovering rules in time series and formalized an algorithm to efficiently locate them. Our algorithm needn't to find motif in time series, so our algorithm is much more efficient than traditional time series rule algorithm. Generally speaking, our algorithm achieves the expected goal. Experimental results on data sets of the power plant demonstrate that the algorithm is very effective.

Although the present approach solves the problem of thermal power plants, the idea is also suitable for solving other problems.

Acknowledgments

We would like to thank all the donors of datasets. We would like to acknowledge the support for our research provided by Huaneng Taicang (2016/EQ/JG-29).

References

[1] Shokoohi-Yekta, M., Chen, Y., Campana, B., Hu, B., Zakaria, J., Keogh, E. Discovery of Meaningful Rules in Time Series. In proceedings of KDD 2015.

[2] Weiss, S., Indurkhya, N., and Apte, C., Predictive Rule Discovery from Electronic Health Records. ACM IHI, 2010.

[3] Mueen, A., Keogh, E., Zhu, Q., Cash, S. and Westover, B. Exact Discovery of Time Series Motif. SDM 2009.

[4] Abonyi, J., Feil, B., Nemeth, S., Arva, P. Modified GathCGeva clustering for fuzzing segmentation of multivariate time series. Fuzzy Sets and Systems, Data Mining Special Issue 149, 2005: 39 - 56.

[5] Tak-chung Fu. A review on time series data mining. In: Proceedings of Engineering Applications of Artificial Intelligence, 2011: 164 - 181.

[6] Hu, B., et al. Discovering the Intrinsic Cardinality and Dimensionality of Time Series Using MDL. ICDM 2011.

[7] G. E. A. P. A. Batista, X. Wang, E. J. Keogh, A Complexity-Invariant Distance Measure for Time Series, SDM, 2011: 699 - 710.

[8] E. J. Keogh, J. Lin, S. H. Lee, and H V. Herle. Finding the most unusual time series subsequence: algorithms and applications, Knowl. Inf. Syst., vol 11, no. 1, 2007: 1 - 27.

[9] K. Ueno, X. Xi, E. J. Keogh, D. J. Lee, Anytime Classification Using the Nearest Neighbor Algorithm with Applications to Stream Mining, ICDM, 2006: 623 - 632.

[10] D. Yankov, E. J. Keogh, U. Rebbapragada, Disk aware discord discovery: finding unusual time series in terabyte sized datasets, In proceedings of Knowl. Inf. Syst., vol 17, no. 2, 2008: 241 - 262.

[11] Rakthanmanon,T., Keogh, E., Lonardi,S,. MDL-Based Time Series Clustering. In proceedings of Knowledge and Information Systems. 2012, Volume 33, Issue 2: 371 - 399.

[12] Rakthanmanon, T., Keogh, E., Lonardi, S., Evans, S.: MDL-based time series clustering. Knowl. Inf. Syst. 33(2), 2012: 371- 399.

[13] Vinh, V.T., Anh, D.T.: Some novel improvements for MDL-based semi-supervised classification of time series. In: Proceedings of Computational Collective Intelligence. Technologies and Applications, Springer, Berlin. LNAI 8733, 2014: 483 - 493.

[14] Begum, N., Hu, B., Rakthanmanon, T., Keogh, E. A minimum description length technique for semi-supervised time series classification. In: Integration of Reusable Systems Advances in Intelligent Systems and Computing, 2014: 171 - 192.

[15] Vinh, V.T., Anh, D.T.: Constraint-based MDL principle for semi-supervised classification of time series. In: Proceedings of 7th International Conference on Knowledge and System Engineering, Ho Chi Minh City, 8 - 10 Oct 2015: 43 - 48.