Regular Paper

Motif Density for Selecting Optimal Window Length in Motif Discovery

Makoto Imamura*, Mao Inoue*, Masahiro Terada*, and Daniel Nikovski**

** School of Information and Telecommunication Engineering, Tokai University, Japan
** Mitsubishi Electric Research Laboratories, USA
imamura@tsc.u-tokai.ac.jp

Abstract- Motif discovery is not only a fundamental method for finding repetitive subsequences in a longer time series, but is also used as a sub-routine in higher-level analytics including classification, clustering, visualization, and rule discovery. However, existing motif discovery algorithms depend critically on the knowledge of the correct subsequence length. Therefore, deciding an appropriate window length for subsequences is required before using those algorithms. In this work, we investigate how to decide an appropriate window length. We propose a novel index called a 'motif index' that counts the number of similar subsequence occurrences within the neighborhood in the space of subsequence, while avoiding trivial matches. We also propose a heuristic method to select an appropriate error distance for the neighborhood required as a parameter to define motif density. Furthermore, we show that motif density can decide an optimal window in the simulation data in which motifs are intentionally embedded.

Keywords: Time series data mining, Motif discovery, Window length selection, Motif density

1 INTRODUCTION

Time series motifs [1][2] are approximately repeating subsequences embedded in a time series. Motifs are one of the most important primitives in time-series data mining, and motif discovery has been used as a sub-routine in higher-level analytics, including classification, clustering, visualization and rule-discovery. Moreover, motif discovery has been applied to domains as diverse as factory operation [3], medicine [4], and seismology [5]. The notion of a motif is useful for a wide range of applications, because a repeated and frequently occurring pattern implies a latent system that occasionally produces a repeatable output. For example, this system may be an over-caffeinated heart, sporadically introducing a motif pattern containing an extra beat [6], or the system may be a factory worker, producing repetitive movement in a series of assembly operations [3].

Since the Matrix Profile [7], a fast and scalable algorithm for subsequence all-pairs-similarity-search in time series, has been introduced, it has helped to develop new innovative ideas for time-series data mining [8]. However, because a motif is defined as a pair of subsequences the distance between which is the smallest, it does not necessarily imply the frequent occurrence of a motif subsequence. That is, there are not necessarily many subsequences in the neighborhood of a motif. Furthermore, motif discovery algorithms expect that a subsequent length be chosen beforehand, which usually means in practice that users must try several possible lengths, and must confirm that the discovered motif indeed has frequent similar subsequences in a time series.

In this work, we propose a novel index called a 'motif density' that counts the number of similar subsequence occurrences within the neighborhood in the space of subsequences, ignoring trivial matches. We also propose a heuristic method to select an appropriate error distance for the neighborhood, where error is a parameter that decides the similarity level in motif density. Furthermore, we show that motif density can decide an optimal window in simulation data in which motifs have been embedded intentionally.

The rest of our paper is organized as follows. Section 2 describes the definition of a motif, and the criteria to determine the appropriateness of a subsequence as a motif. Section 3 defines motif density, based on a neighborhood of a subsequence in a set of subsequences without trivial matching subsequences. Section 4 proposes an algorithm to calculate motif density. Section 5 evaluates our proposed algorithm empirically. First, we show that motif density can decide an optimal window-length for finding motifs. Second, we evaluate a heuristic method to select an appropriate error distance for the neighborhood, required as a parameter to define motif density.

2 MOTIF CRITERIA

This section describes the commonly used definition of motif and summarizes the problem of deciding optimal window-length of a motif as motif criteria.

2.1 Our Approach

A motif is defined by using the nearest-neighbor distance in the space consisting of subsequences in a time series.

Definition: time series X

A *Time Series* $X=[x_1,\cdots,x_m]$ is a continuous sequence of real values. We denote the value of the i-th time point by $X[i] = x_i$.

Definition: subsequence $X[p:q]$

A *subsequence* $s = [x_p, x_{p+1},...,x_q] = X[p:q]$ is a list which consists of continuously occurring values of X, starting at position p and ending at position q.

The *length* w of a subsequence s is $w = q - p + 1$, and we denote it by *length*(s). We also denote a subsequence $X[p:q]$ by $X_w(p)$, which means a subsequence staring at p with length w.

Definition: support of a subsequence

The *support* of a subsequence S is a set of time points [p:q] = [p, p +1,..., q-1, q], and we denote it by *support* (s).

Definition: subsequence space $S_w(X)$

A subsequence space is the set of all the subsequences with length w in a time series X. We denote it by $S_w(X)$. A subsequence space is the w-dimensional Euclidean space. Therefore, for given subsequences s_i and s_j, the distance between s_i and s_j, which we denote by dist (s_i, s_j), can be defined similarly to that in a vector space. In this paper, we use L_1 distance defined below.

$$dist(X_w(p), X_w(q)) \equiv \sum_1^w |X(p+i-1) - X(q+i-1)|$$

Definition: disjoint subsequences

Let s_i and s_j be subsequences. When *support* (s_i) and *support* (s_j) are disjoint, that is, *support* (s_1) ∩ *support* (s_2) = ∅, we say that s_i and s_j are disjoint.

Definition: Motif subsequence (1-NN)

Let w be a window-length, and let X be a time series. A subsequence s with length w in X is said to be *motif*, if it satisfies the below condition.

There is a subsequence s′ with window-length w, such that

$$\text{dist}(s, s') = \min_{i,j} \{ \text{dist}(s_i, s_j) \mid s_i, s_j \in S_w(X) \text{ and } \text{support}(s_i) \cap \text{support}(s_j) = \emptyset \}$$

The above definition is based on the one-nearest-neighbor (1-NN) distance. We can extend this definition to k- nearest-neighbor distance by replacing the minimum with the k-th minimum in the above condition.

2.2 Challenges in Defining Motif Criteria

In this subsection, we investigate criteria to determine the appropriateness of a subsequence as a motif, which we call motif criteria. The intuitive meaning of a motif is a subsequence which has many similar subsequences in a time series, therefore we will try to define an index to measure the meaning of 'similar' and 'many' according to the intuition above. A similar sequence is measured by the distance between subsequences. For defining "many", we should count the number of similar subsequences to a motif. We call this number "occurring frequency". Challenges in defining occurring frequency are summarized in the following three points.

(1) Error dependency

When we say a sequence s_i is similar to a subsequence s, it means that $dist(s_i, s)$ is small. Therefore, the threshold of an error distance parameter ϵ is required for counting similar subsequences. A naïve definition of occurring frequency of s is $|\{s_i \mid s_i \in S_w(X) \text{ and } dist(s_i, s) \leq \epsilon\}|$, where |A| means the number of elements of a set A. This definition of occurrence frequency requires a window-length w and an error ϵ as parameters. That is, how to decide an appropriate pair of a window-length w and an error value ϵ is the first challenge.

(2) Window-length dependency

If an error value is equal in subsequences with different lengths, the longer subsequence seems to be more appropriate than the shorter one as a motif. How to normalize by a window-length is the second challenge.

(3) Trivial match

Subsequences close to a subsequence s in a time series are similar to s, if the time series is continuous and varies slowly. We call this property "trivial match". A trivial match is described formally by the property that dist(X[p': $p'+w-1$], X[p: $p+w-1$]) is small, if $|p-p'| \ll w$. When we count similar subsequences, we must remove trivially matching sequences. The third challenge is how to count similar subsequences, while avoiding trivially matching subsequences.

3 MOTIF DENSITY

3.1 Our Approach

This subsection describes our approach to solving each of the problems described in the precious section.

(1) Error parameter dependency

We shall define the neighborhood of a subsequence in $S_w(X)$ for a given time series X, a window-length w and a threshold on the distance ϵ.

(2) Window-length dependency

We shall define '*motif density*' which expresses occurring frequencies normalized by window-lengths for comparing the appropriateness among motifs with different window-lengths.

(3) Trivial match

When we define the neighborhood of a subsequence s in a subsequence space $S_w(X)$, we remove trivially matching subsequences of s by using the concept of disjoint subsequences defined previously. That is, we shall define a special topology for a subsequence space generated by a time series.

3.2 Neighborhood of a Subsequence

We will define the neighborhood of a subsequence in a time series to avoid a trivial match problem.

Definition: *Disjoint neighborhood of a subsequence*

Let X, w , ϵ and s are a time series, a window-length, a positive real number, and a subsequence with length w respectively. A subset of $S_w(X)$, $D_{w,\epsilon}(s)$, is called a disjoint neighborhood of a subsequence, if it satisfies the following conditions.

(i) For every $s_i \in D_{w,\epsilon}(s)$, $dist(s_i, s) \leq \epsilon$

(ii)For every $s_i, s_j \in D_{w,\epsilon}(s)$, support $(s_i) \cap$ support $(s_j) = \emptyset$.

We select a maximal one for constructing the occurring frequency of a subsequence.

Definition: Maximal neighborhood of a subsequence

Let $\mathcal{D}_{w,\epsilon}(s)$ denote a set of all of the disjoint neighborhoods of a subsequence s. A disjoint neighborhood of a subsequence s is said to be a maximal neighborhood $B_{w,\epsilon}(s)$, if it has the largest number of elements in $\mathcal{D}_{w,\epsilon}(s)$. $B_{w,\epsilon}(s)$ can be defined formally by the following formula.

$$B_{w,\epsilon}(s) = \underset{D_{w,\epsilon}(s)\in\mathcal{D}_{w,\epsilon}(s)}{\operatorname{argmax}} \left|D_{w,\epsilon}(s)\right|,$$ where $\left|D_{w,\epsilon}(s)\right|$ means the number of elements of $D_{w,\epsilon}(s)$.

We shall define the occurring frequency of a subsequence s by the number of element of $B_{w,\epsilon}(s)$. The following theorem gives us how to construct a $B_{w,\epsilon}(s)$ for given w, ϵ, and s.

Theorem: Construction of a maximal neighborhood.

Let X, w, ϵ and s be a time series, a window-length, a positive real number, and a subsequence with length w, respectively. $B_{w,\epsilon}(s)$, which is constructed by the below procedure, is a maximal neighborhood of s.

(step1) Select the disjoint subsequences whose distances from s are smaller than ϵ from s towards right (later time) to the end of a time series in order. We call the set of those subsequences a right disjoint set.

(step2) Select disjoint subsequences whose distances from s are smaller than ϵ from s towards left (earlier time) to the beginning of a time series in order. We call the set of those a left disjoint set.

(step 3) Let $B_{w,\epsilon}(s)$ be the union of the right and left disjoint sets.

Proof:

Let $B'_{w,\epsilon}(s)$ be one of the maximal neighborhoods of s. It is enough to prove $\left|B_{w,\epsilon}(s)\right| = \left|B'_{w,\epsilon}(s)\right|$, where $\left|B_{w,\epsilon}(s)\right|$ means the number of the elements of $B'_{w,\epsilon}(s)$.

We show only the case from s toward right to the end, because the case towards left is similar.

Let the elements of $B_{w,\epsilon}(s)$ be sorted by time ordering, we obtain

$B_{w,\epsilon}(s) = \{\ldots, s = X_w(p), X_w(p_1), X_w(p_2), \ldots, X_w(p_n)\}$
where $p < p_1 < p_2 < \cdots < p_n$.

Similarly, we obtain

$B'_{w,\epsilon}(s) = \{\ldots, s = X_w(p), X_w(p_1'), X_w(p_2'), \ldots, X_w(p_n')\}$
where $p < p_1' < p_2' < \cdots < p_n'$.

By the above construction of $B_{w,\epsilon}(s)$, p_1 is the smallest, so $p_1 \le p_1'$. In the same way, we get $p_2 \le p_2'$, because "$X_w(p_2')$ is disjoint with $X_w(p_1)$" and "$X_w(p_2)$ is the leftmost disjoint subsequence with $X_w(p_1)$". By mathematical induction, we obtain $p_i \le p_i'$ for $1 \le i \le n$, where n is $\left|B'_{w,\epsilon}(s)\right|$. This shows that $\left|B'_{w,\epsilon}(s)\right| \le \left|B_{w,\epsilon}(s)\right|$.

If $\left|B_{w,\epsilon}(s)\right| < \left|B'_{w,\epsilon}(s)\right|$, it is contrary to the maximality of $\left|B'_{w,\epsilon}(s)\right|$. Therefore, $\left|B_{w,\epsilon}(s)\right| = \left|B'_{w,\epsilon}(s)\right|$, which is what we wanted to prove.

3.3 Occurring Frequency and Motif Density

First, we define the occurring frequency of a subsequence for each window-length.

Definition: Occurring frequency

Let w , ϵ and s are a window-length, a positive real number, and a subsequence with length w, respectively.

The occurring frequency of a subsequence s is the number of the elements of a maximal neighborhood of a subsequence $B_{w,\epsilon}(s)$, that is, $\left|B_{w,\epsilon}(s)\right|$.

Next, we define motif density to normalize the difference among window-length.

Definition: Motif density

Let w , ϵ and s are a window-length, a positive real number, and a subsequence with window-length w, respectively.

The motif density of a subsequence s is $w \times \left|B_{w,\epsilon}(s)\right|$.

We regard a subsequence that has the highest motif density as the best motif among all the subsequence with various window-lengths. We show a procedure to select the best motif.

1. Give a list of window-lengths $W = [w_1, \ldots, w_i, \ldots, w_n]$.
2. Select the subsequence s_i which has the largest occurring frequency for each window-length w_i in W. We call the subsequence s_i the optimal motif for a window-length w_i.
3. Select the motif that has the highest motif density among the optimal motifs $[s_1, \ldots, s_i, \ldots, s_n]$ for the window-lengths W. We call this motif *the best motif* among optimal motifs for window-lengths W. We also call the window-length of the best motif *the best motif length*.

In the above procedure, the best motif length depends on an error parameter ϵ that determine the similarity level in counting occurring frequency. We call ϵ an error parameter hereafter. The error parameter in motif density is essential like a parameter k is essential in k-means clustering algorithm. We propose a method to help finding the appropriate error parameter ϵ like the Elbow method [11] for finding the appropriate number k of clusters in clustering. When we plot motif density against error parameter values, we get the graph of a monotonically increasing function. We can select an appropriate error parameter value where the rate of increase suddenly drops in the graph. This method based on the institution that a good motif has a clear boundary that divides similar subsequences from dissimilar ones after trivially matching sequences are removed.

An optimal motif for a smaller window-length than the best motif length has relatively high motif density value, because a part of a motif is also a motif. Furthermore, a motif with a smaller length might have a quickly rising motif density at very small error values.

We summaries the above considerations as three hypotheses.

Hypothesis 1: Motif density can decide the best window-length for motif discovery.

Hypothesis 2: An optimal motif for a smaller window-length than the best motif length has relatively high motif density values.

Hypothesis 3: We can select an appropriate error parameter by means of an Elbow method for the graph of a motif density functions against error parameter values.

We shall evaluate the above hypotheses in Section 5.

4 ALGORITHM

We can obtain algorithms for calculating occurring frequency and motif density by operationally interpreting the definitions and the theorem in the previous section.

Table 1 shows an algorithm that counts the occurring frequency of a given subsequence. The inputs are a time series X, a window-length w of the given subsequence s, a starting time t of s, and an error per window-length ϵ. The outputs are the occurring frequency and the motif density of the given subsequence s.

Line 01 calculates the distance between the given subsequence s and each subsequences in $S_w(X)$. Line 02 counts the number of elements that are in the right-hand side of s in the maximal neighborhood of s. Line 03 counts the number of those in the left-hand side of s. Line 04 counts the total occurrence frequency of s by adding the occurrence frequency in the right side obtained by line 02 to that in the left side obtained by line 03. Line 05 calculates the motif density of s by multiplying the window-length w and the occurring frequency obtained by line 04.

Table 2 shows an algorithm that counts the number of elements of a maximal neighborhood subsequence set whose elements are to the right of the given subsequence s. The inputs are the distance list DL obtained by line 01 in Table 1 the window-length w of a given subsequence s, a starting time t of s, and the error per window-length ϵ. The output is the number of maximal neighborhood subsequences in the right side of s.

Line 01 initializes a time cursor 'Cur' and a normalized error 'Thr'. Line 02-13 is a while-loop that chooses maximal subsequences that are in the right-hand side of the given subsequence s toward the end of the time series X. Line 03-05 is a while-loop that searches the next disjoint subsequence whose distance from s is smaller than 'Thr'. Line 06-09 increments 'Right' when the line 03-05 found a new disjoint subsequence. Line 10-12 exits while-loop 02-13 after checking all the subsequences in the right-hand side of s.

Table 3 shows an algorithm that counts the number of maximal neighborhood subsequences which are in the left-hand side of the given subsequence s. The left-hand case is reduced to the right-hand case by reversing the time series values from right to left.

Line 01 reverses the distance list 'DL' from right to left. Line 02 reverses the starting time t of s from right to left. Line 03 gets the value of the left-hand case by calling the algorithm 'CountRightOccurence' with reversed arguments.

Table 1. CountOccurringFrequency Algorithm.

Algorithm: CountOccurringFrequency *(X, w, t, ϵ)*

[Input] *X*: Given time series
w: Length of a given subsequence *s*
t: Stating time of a given subsequence *s*
ϵ: Error per window-length

[Output] OF: Occurring frequency of s for *w and ϵ*
MD: Motif density of s

```
01  DL = distanceListFromS(X, t, w);
02  OFR = countRightOccurence (DL, t, w, ϵ)
03  OFL = countLeftOccurence (DL, t, w, ϵ)
04  OF = OFR + OFL;
05  MD = OF * w;
06  return (OF, MD);
```

Table 2. CountRightOccurence.

Algorithm: *countRightOccurence* (DL, *t, w, ϵ*)

[Input] DL: Distance list
w: Window-length of a given subsequence *s*
t: Stating time of *s*
ϵ: Error per window-length

[Output] Right: the number of maximal neighborhood subsequences to the right of *s*.

```
01  Cur = t+1;  Thr = ϵ * W;
02  while Cur <= length(DL)
03    while DL(Cur) >  Thr or Cur <= length(DL)
04      Cur = Cur + 1;
05    end
06    if DL(Cur)  <=  Thr
07      Right := Right + 1;
08      Cur := Cur + w – 1;
09    end
10    if Cur > length(X)
11      break;
12    end
13  end
14  return Right;
```

Table 3. CountLeftOccurence Algorithm.

Algorithm: *countLeftOccurence (*DL, *w, t, ϵ*)

[Input] DL: Distance list
w: Window-length of a given subsequence *s*
t: Stating time of a given subsequence *s*
ϵ: Error per window-length

[Output] Left: the number of maximal neighborhood subsequences in the left of *s*.

```
01  DL_rev = fliplr(DL);
02   t_rev  = length(X) – t + 1;
03   Left = countRightOccurence (DL_rev, t_rev, w, ϵ)
```

5 EXPERIMENTAL EVALUATION

We evaluate the three hypotheses described in section 4.

5.1 Window Length Selection

This subsection evaluates the two hypotheses below in two simulated time series in which motif subsequences are intentionally embedded.
Hypothesis 1: The best window-length can be decided by selecting the one that has the highest motif density values.
Hypothesis 2: A maximal motif for a smaller window-length than the best motif length has relatively high motif density values.

(1) Experiment on data set 1

First, we will show that motif density can be used to decide the best motif length (15) by selecting the window-length that has the highest motif density among optimal motifs with window-lengths 5,9,15, and 31.

Figure 1 is a simulated time series that combines sine curves with length (period) 15 samples per one cycle, and random subsequences with various lengths. In Fig. 1, the horizontal axis means time points in the time series, and the vertical axis means the values of the time series. Sine curves with length 15 are intentionally embedded as motifs. We call this time series data set 1.

Data set 1 is obtained by alternatively arranging 'a noisy sine curve whose length of one cycle is 15' and 'a random subsequence that has a random length between 1 and 15' for twenty times. Each value in a random subsequence follows a random uniform distribution whose values are between -1 and 1. The noise included in a sine curve follows a random uniform distribution whose values are between -0.02 and 0.02.

Figure 2 shows the motif density trend graph for each window-length in the case that an error per window-length parameter (we call it as EPA hereafter) is 0.01. In each graph of Fig. 2, the horizontal axis means time points in the time series, and the vertical axis means the motif density of each subsequence starting at each time point. A procedure how to decide an EPL will be described in the next subsection. The top graph is a motif density trend for window-length 5. The second, third, and fourth trend graphs from the top to the bottom are those for window-lengths 5, 9, 15, and 31 respectively. The third trend graph for window-length 15 has highest motif density values at the times when motif patterns start. The trend graphs for lengths 5 and 9 have times at which sub-patterns of the optimal motifs with length 15 have relatively high motif density values and longer peak durations than those of length 15. The reason for this observation is in the fact that the best motif pattern includes motifs with smaller window-lengths. They also support hypothesis 2.

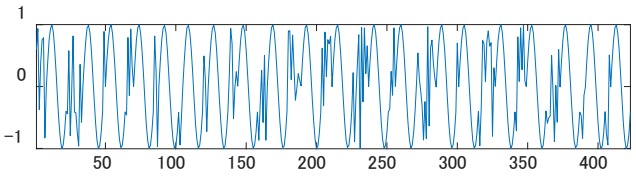


Figure 1: A time series with a motif of length 15 samples.

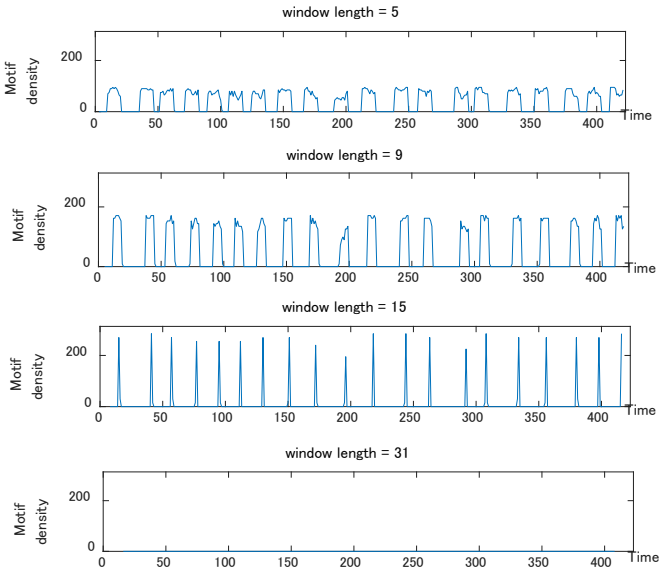


Figure 2: Motif density trend for each window-length (in case of EPL 0.01).

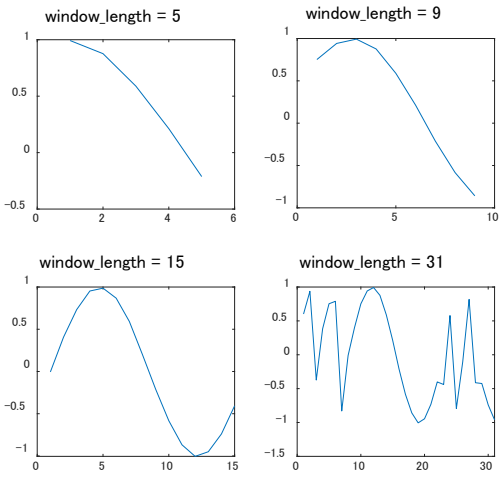


Figure 3: Optimal motif for each window-length.

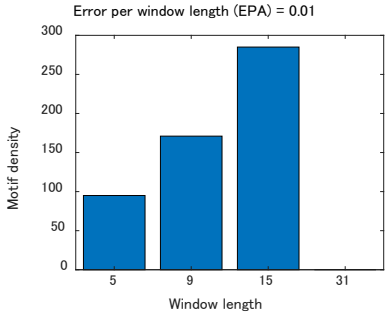


Figure 4: Highest Motif density of the optimal motif for each window-lengths (in case of EPL 0.01).

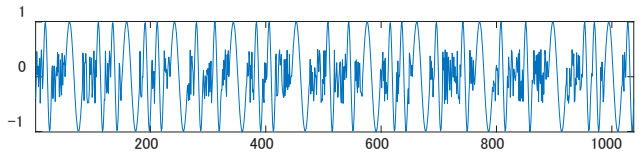


Figure 5: A time series with length 15 and 31 motifs.

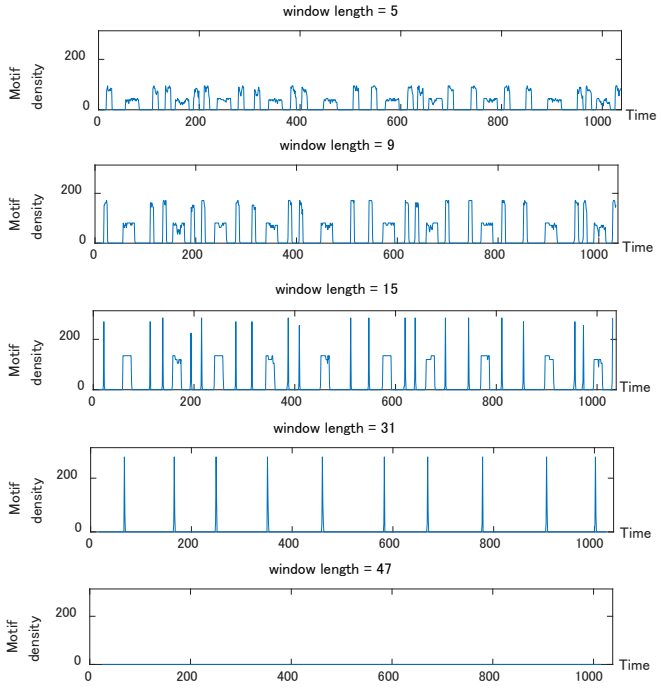


Figure 6: Motif density of each times for each window-lengths (in case of EPL 0.01).

Figure 3 shows each optimal motif in each window-length. The optimal motif in length 15 is also the best motif in the sense that it has the highest motif density as will be shown in Fig. 4. In each graph of Fig. 3, the horizontal axis means time points of each optimal motif subsequence in time series. The vertical axis means the values of each optimal motif. The best motifs for window-lengths 5 and 9 are the sub-patterns of the best motif with length 15. The optimal motif with length 31 is a subsequence including the optimal motif with length 15.

Figure 4 shows the motif density value for each optimal motif with each window-length in case of EPL 0.01. In Fig. 4, the horizontal axis means the length of each optimal motif, and the vertical axis means the motif density of each optimal motif. The window-length that has the highest motif density is 15. It supports hypothesis 1 that "motif density can be used to decide the best window-length". It also supports hypothesis 2, "a maximal motif for a smaller window-length 5, 9 than the best motif length 15 has relatively high motif density value".

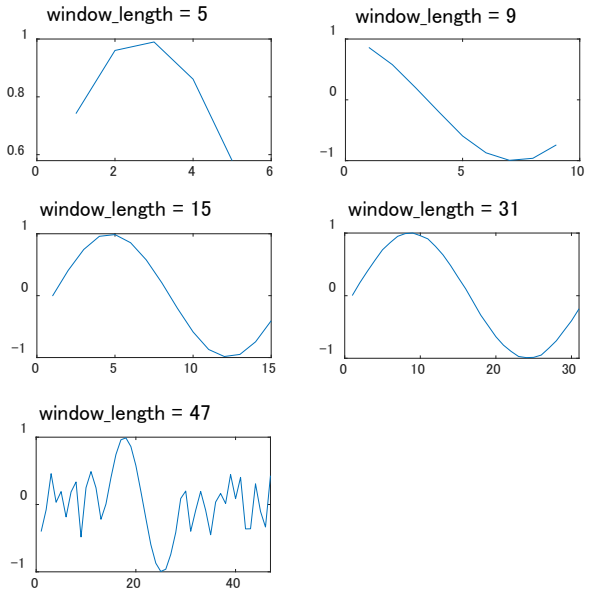


Figure 7: Best motifs for each window-lengths.

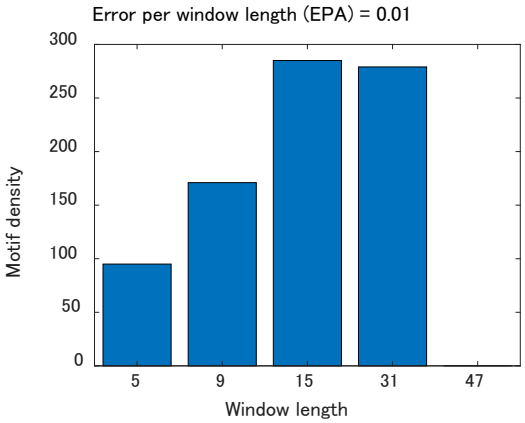


Figure 8: Highest Motif density of the optimal motif with each window-length (in case of EPL 0.01).

(2) Experiment on data set 2

Next, we show that motif density can be used to decide the best motif length (15 and 31) in time series in which two motifs with length 15 and 31 are intentionally embedded.

Figure 5 is a simulated time series that combines sine curves with length 15 and 31 with random subsequences of various lengths. The horizontal axis and the vertical axis in Fig. 5 have the same meanings as those in Fig. 1. Sine curves with length 15 are intentionally embedded motifs. We call this time series data set 2.

Data set 2 is obtained by alternatively arranging ‘a random subsequence which has random length between 1 and 31’, ‘a noisy sine curve whose length of one cycle is 15’ and ‘a noisy sine curve whose length of one cycle is 31’. In data set 2, a 5-subsequence pattern in which a random subsequence, a sine curve with length 15, a random one, a sine curve with length 31 and a random one are arranged in this order repeat for 10 times. Those random subsequences and the noise of sine curves follow the same random uniform distributions in data set 1.

Figure 6 shows the motif density trend for each window-length for the case of a EPL of 0.01. The horizontal axis and

the vertical axis in Fig. 6 have the same meanings as those in Fig. 2. The trend graphs are for window-lengths 5, 9, 15, 31, and 47, from the top to the bottom, respectively. As in the first experiment, the trend graphs for lengths 15 and 31 have the high peaks of motif density at the times when motif patters occur. The trend graphs for 5, 9, and 15 have relatively high motif density values at times when sub-patterns of the motif patterns with length 15 or 31 occur.

Figure 7 shows each optimal motif in each window-length. The optimal motifs in window-lengths 15 and 31 are the best motifs in the sense that they have larger motif densities as will be shown in Fig. 8. (and have been intentionally embedded). The horizontal axis and the vertical axis in Fig. 7 have the same meanings as those in Fig. 3. The optimal motifs of window-lengths 5 and 9 are sub-patterns of the best motifs with window-length 15 or 31. The optimal motif with window-length 47 is a subsequence including the best motif with window-length 31.

Figure 8 shows each motif density trend for each window-lengths. The horizontal axis and the vertical axis in Fig. 8 have the same meanings as those in Fig. 4. It shows that 15 and 31 are the top 2 window-lengths. This supports hypothesis 1. It also shows that window-lengths smaller than 15 have relatively high motif density values. This supports hypothesis 2.

5.2 Error Parameter Selection

This subsection evaluates the hypothesis 3 below.

Hypothesis 3: we can select an appropriate error parameter by means of an Elbow method for the graph of a motif density functions for window-lengths

(1) Experiment on data set 1

Figure 9 shows each error dependency graph of the motif density for each optimal motif with window-length 5, 9, 15, and 31. In Fig. 9, the horizontal axis means EPL values, and the vertical axis means the motif density of at each EPL value.

In data set 1, the window-length of intentionally embedded motifs is 15. The range of EPL for the top graph (a) is from 0 and 2, and that for the bottom one (b) is from 0 to 0.015. The graph (a) shows that when EPL is over 0.8, there are no differences among motif densities for all the window-lengths even though the best motif length is 15. The graph (b) shows that motif densities for window-lengths 5,9 and15 rise quickly at EPL of 0.005, and increase while EPL is from 0.005 to 0.01 and then become constant from EPL values of 0.01. Therefore, 0.01 is an elbow point for window-lengths 5, 9 and 15. On the other hand, the motif density for window-length 31 has constant value 0 for EPL ranging from 0 to 0.015. This observation shows that an appropriate EPL is 0.01 for finding the best motif length shown in the previous subsection. That is, this observation supports hypothesis 3 in case of data set 1.

We compare motif density trend with different EPLs in order to understand the intuitive meaning of EPL. Figure 10 shows the motif density trend for optimal motifs with window-length 5, 9, 15, and 31. The horizontal axis and the vertical axis in Fig. 10 have the same meaning as those in Fig. 2.

The EPL of the top graph (a), that of the middle one (b) and that of the bottom one (c) are 0.01, 0.1 and 1, respectively. In the case of EPL equal to 0.01, the graph for the best motif length (15) has sharp peaks when similar subsequences occur. On the other hand, in the case of EPL=0.1, the graph for it has only blunt peaks. Furthermore, in the case of EPL=1, there seems to be no peaks. The graphs for smaller window-lengths (5, 9) than the best motif length (15) have similar trends to that for 15. Motif densities for 5 and 9 have relatively high values, because subsequences of a motif are motifs. That is, if X(i:i+14) is a motif , X(i:i+4), X(i+1:i+5), …, and X(i+10:i+14) are also motifs. This is why motif density for window-length 5 and 9 have less sharp peaks than those for window-length 15. This observation also supports hypothesis 2.

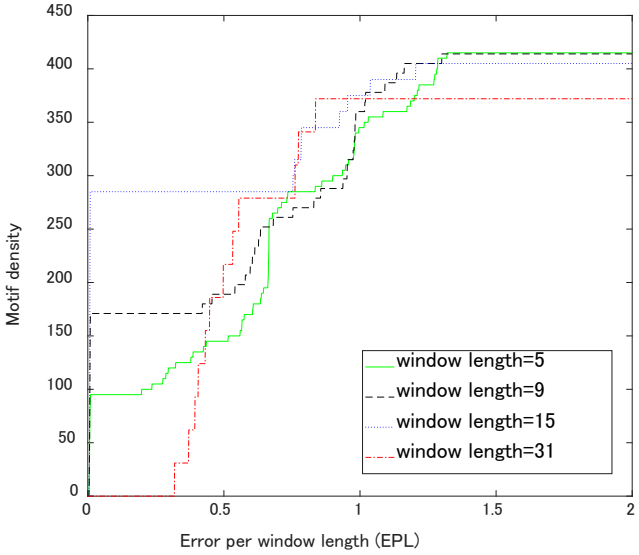
(2) Experiment on data set 2

Figure 11 shows each error dependency graph of the motif density for each optimal motif with window-length 5, 9, 15, 31, and 47. The horizontal axis and the vertical axis in Fig. 11 have the same meaning as those in Fig. 9. In data set 2, the window-lengths of intentionally embedded motifs are 15 and 31. The range of EPL for the top graph (a) is from 0 and 2, and that for the bottom one (b) is from 0 to 0.015. Graph (a) shows that when EPL is over 1, there are no differences among motif densities for all the window-lengths, even though the best motif lengths are 15 and 31. Graph (b) shows that motif densities for window-lengths 5,9, and15 rise quickly at EPL=0.005 and increase while EPL ranges from 0.005 to 0.01, and then become constant from about EPL=0.01. Therefore, 0.01 is an elbow point for window-lengths 5, 9, 15, and 31. On the other hand, the motif density for window-length 47 has a constant value 0 for EPL ranging from 0 to 0.015. This observation shows that an appropriate EPL is 0.01 for finding the best motif length shown in the previous subsection. That is, this observation supports hypothesis 3 in the case of data set 2.

As with experiment 1, we investigate density trend graphs with different EPLs. Figure 12 shows the motif density trend for optimal motifs with window-length 5, 9, 15, 31, and 47. The horizontal axis and the vertical axis in Fig. 12 have the same meaning as those in Fig. 2.

The EPL of the top graph (a), that of the middle one (b), and that of the bottom one are 0.01, 0.1 and 1, respectively. In the case of EPL=0.01, the graph for the best motif lengths 15 and 31 have sharp peaks when similar subsequences occur. The blunt peaks in the graph for window-length 15 correspond to the occurrences of the subsequences of the best motifs with window-length 31. On the other hand, in the case of EPL=0.1, the graphs for window-length 15 and 31 have only blunt peaks. Furthermore, in the case of EPL=1, they have no peaks. As with the experiment on data 1, this observation supports hypothesis 2.

(a) The range of error per length (EPL) is from 0 to 2



(b) The range of EPL is from 0 to 0.015

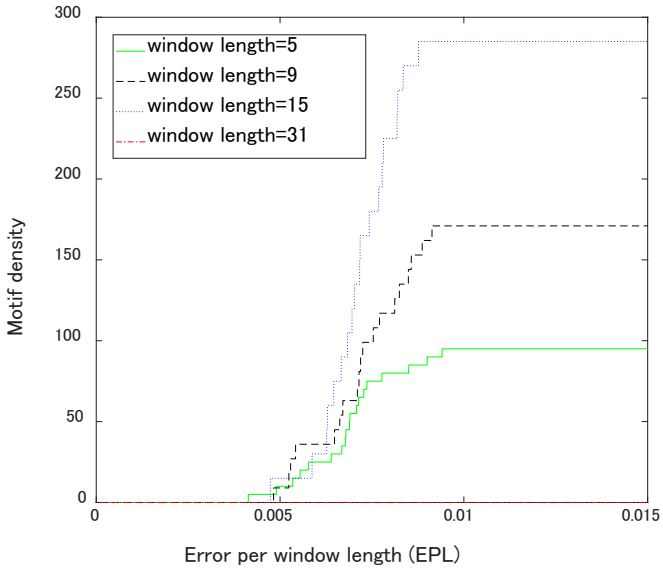
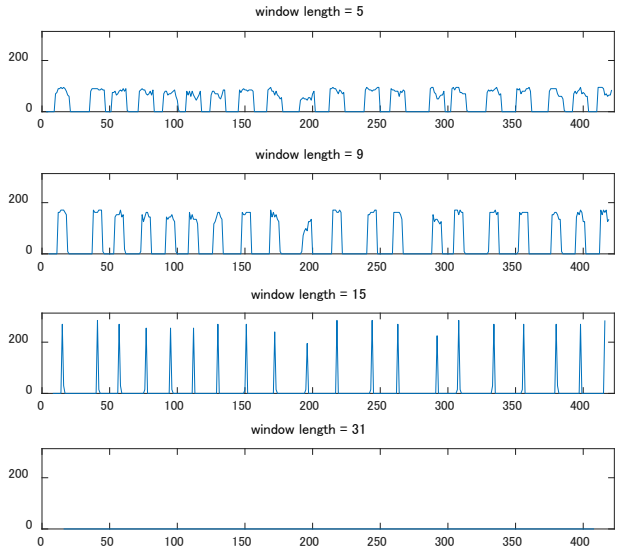
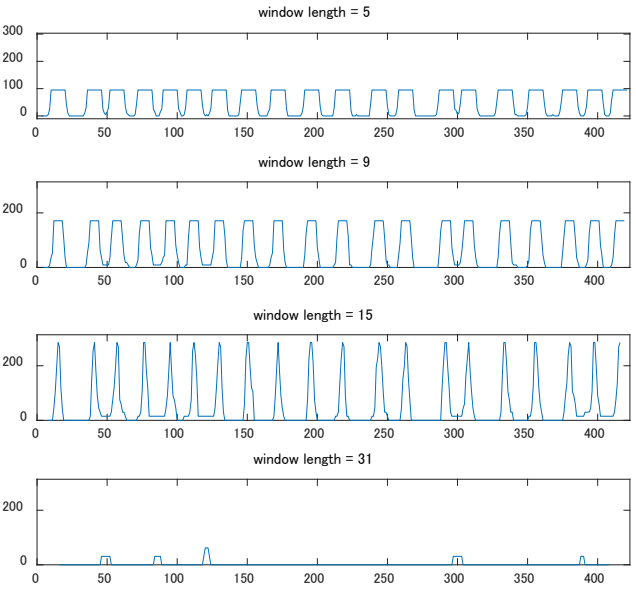


Figure 9: Error dependency of motif density (data 1).

(a) EPL is 0.01



(b) EPL is 0.1



(c) EPL is 1

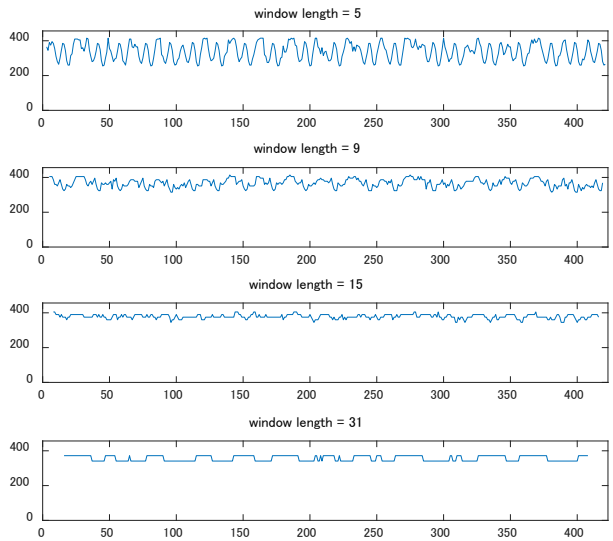
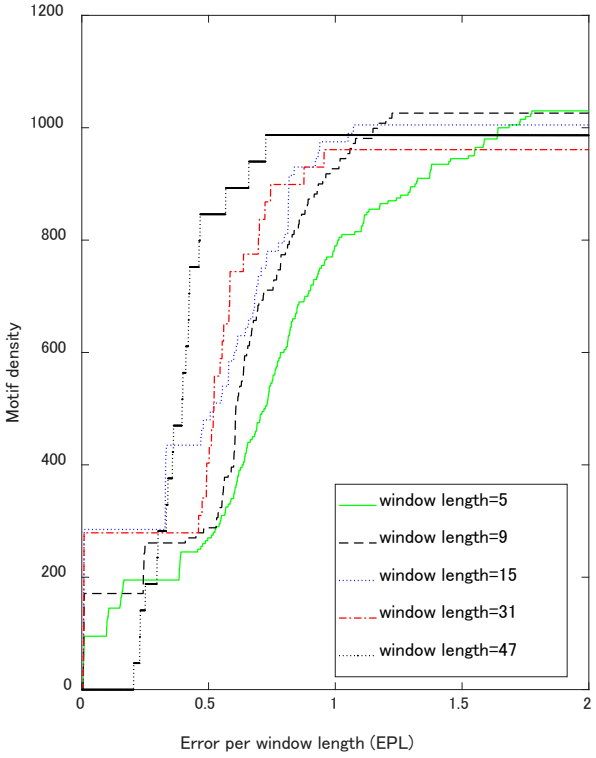


Figure 10: Motif density trends (data1).

(a) The range of EPL is from 0 to 1.5



(b) The range of EPL is from 0 to 0.15

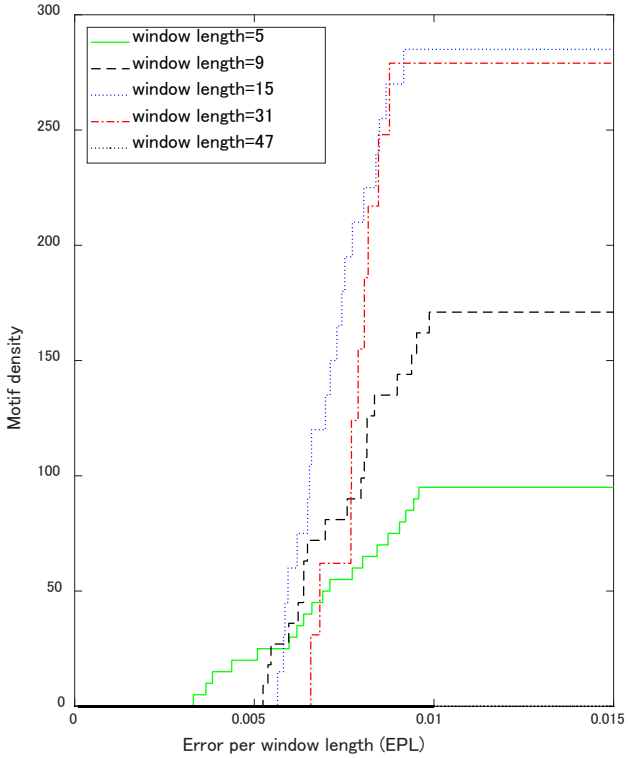
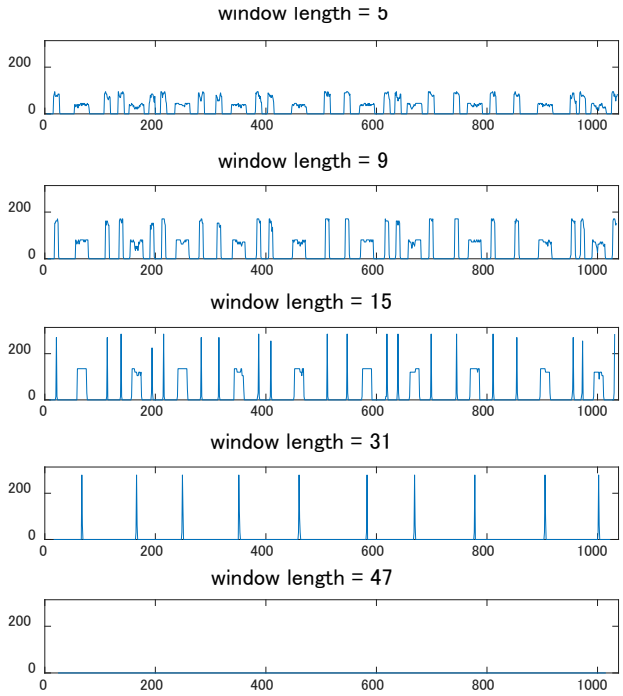
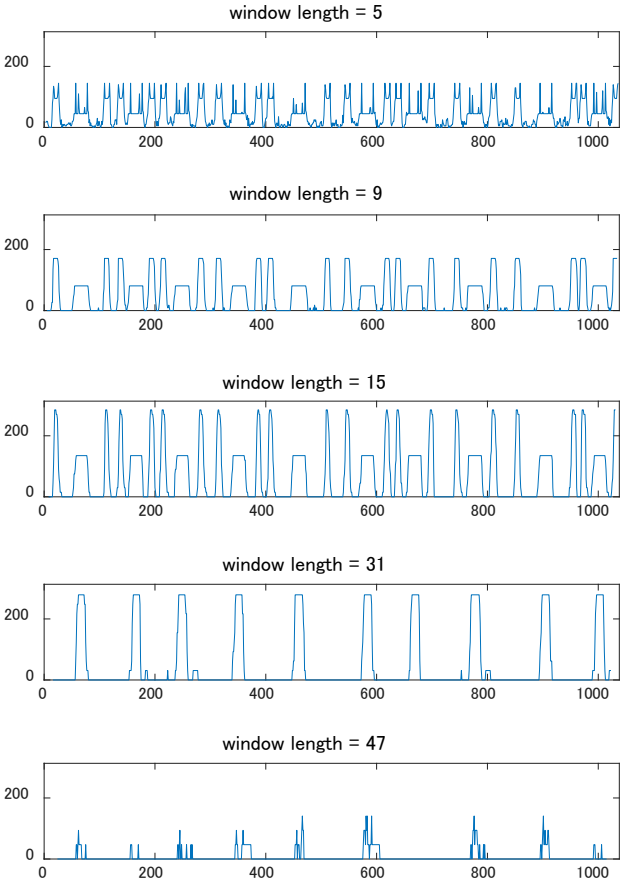


Figure 11: Error dependency of motif density (data 2).

(a) EPL is 0.01



(b) EPL is 0.1



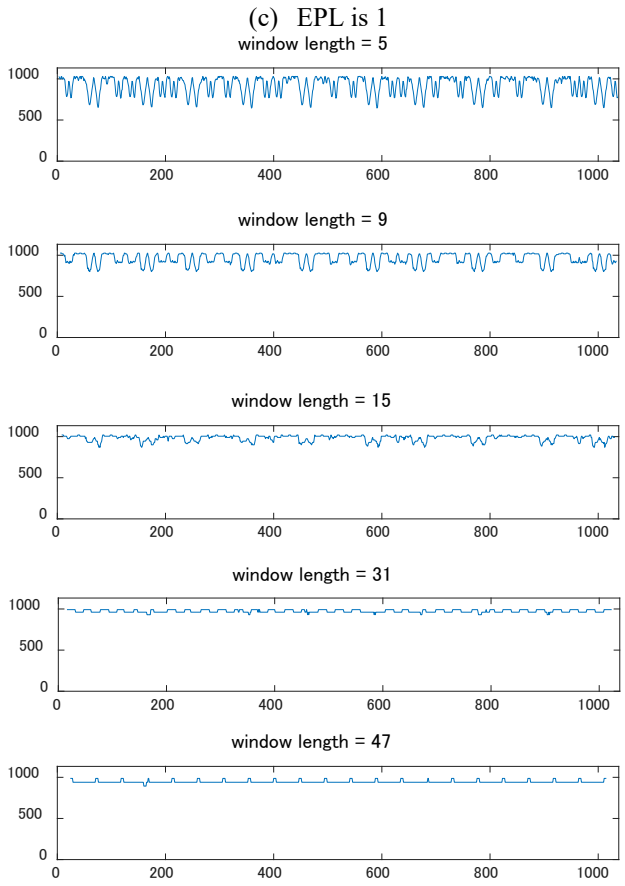


Figure 12: Motif density trends (data2).

6 CONCLUSIONS

We proposed a novel index called 'motif density' together with a selection method to find an appropriate EPL required for defining motif density. The core idea of motif density is in considering a special topology in a subsequence space generated by a time series for avoiding trivial matching and handling different window-lengths. Furthermore, we showed that motif density can decide an optimal window-length in simulated data.

In this paper, we treated the problem of finding one isolated motif in a time series. From a theoretical point of view, it remains as future work how to define and find a sequence of motifs. From an experimental point of view, we plan to apply our algorithms to more complex simulated data, as well as real data.

This work is supported by JSPS KAKENHI Grant Number 17K00161.

REFERENCES

[1] P. Patel, E. Keogh, J. Lin, and S. Lonardi: "Mining Motifs in Massive Time Series Databases", IEEE ICDM pp. 370-377 (2002).

[2] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and M. B. Westover: "Exact Discovery of Time Series Motifs", SDM pp. 473-484 (2009).

[3] T. Maekawa, D. Nakai, K. Ohara, and Y. Namioka: "Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory", UbiComp pp. 1088-1099 (2016).

[4] Z. Syed, C. M. Stultz, M. Kellis, P. Indyk, and J. V. Guttag: "Motif discovery in physiological datasets: A methodology for inferring predictive elements", TKDD Vol. 4, No.1: 2:1-2:23 (2010).

[5] Y. Zhu, Z. Zimmerman, N. S. Senobari, C. M. Yeh, G. Funning, A. Mueen, P. Brisk, and E. Keogh: "Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins", IEEE ICDM, pp. 739-748 (2016).

[6] W. R. Lovallo, M. F. Wilson, A. S. Vincent, B. H. Sung, B. S. McKey, and T. L. Whitsett: "Blood Pressure Response to Caffeine Shows Incomplete Tolerance After Short-Term Regular Consumption", Hypertension vol. 43, No. 4 p.760-765 (2004).

[7] C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh: "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets", IEEE ICDM, pp. 1317-1322 (2016).

[8] Y. Zhu, M. Imamura, D. Nikovski, and E. Keogh: "New Primitive for Time Series Data Mining", IEEE ICDM, pp. 695-704 (2017).

[9] E. Keogh, J. Lin, and W. Truppel: "Clustering of Time Series Subsequences is Meaningless: Implications for Previous and Future Research", IEEE ICDM, pp. 115-122 (2003).

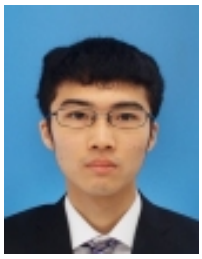
[10] T. Idé: "Why Does Subsequence Time-Series Clustering Produce Sine Waves?", PKDD, pp. 211-222 (2006).

[11] S. Raschka and V. Mirjalili: "Python Machine Learning Second Edition", Packt Publishing, p. 357-358 (2017).

(Received March 6, 2019)
(Revised August 24, 2019)

Makoto Imamura He received a M.E. degree from Kyoto University of Applied Mathematics and Physics in 1986 and a Ph.D. degree from Osaka University of the Information Science and Technology in 2008. From 1986 to 2016, he worked for Mitsubishi Electric Corp and he is presently a Professor at Tokai University. His research interests include machine learning, model-based design and PHM (Prognostics and Health Management). He is a member of IEEE.

Mao Inoue He received a B.E. degree from Tokai University, Japan in 2018. He is a master course student of the school of Information and Telecommunication Engineering at Tokai University. His research interests include IoT systems and data analytics.



Masahiro Terada He received a B.E. degree from Tokai University, Japan in 2019. He is a master course student of the school of Information and Telecommunication Engineering at Tokai University. His research interests include IoT systems and data analytics.

Daniel Nikovski He received a PhD in robotics from Carnegie Mellon University in 2002, and is presently the group manager of the Data Analytics group at Mitsubishi Electric Research Laboratories. He has worked on probabilistic methods for reasoning, learning, planning, and scheduling, and their applications to hard industrial problems. He is a member of IEEE.