Mining Characteristic Multi-Scale Motifs in Sensor-Based Time Series

Ugo Vespier[1] Siegfried Nijssen[1,2] Arno Knobbe[1]

[1] LIACS, Universiteit Leiden, The Netherlands
[2] Department of Computer Science, KU Leuven, Belgium
{uvespier,snijssen,knobbe}@liacs.nl

ABSTRACT

More and more, physical systems are being fitted with various kinds of sensors in order to monitor their behavior, health or intensity of use. The large quantities of time series data collected from these complex systems often exhibit two important characteristics: the data is a combination of various superimposed effects operating at different time scales, and each effect shows a fair degree of repetition. Each of these effects can be described by a small collection of *motifs*: recurring temporal patterns in the data. We propose a method to discover characteristic and potentially overlapping motifs at multiple time scales, taking into account systemic deformations and temporal warping. Our method is based on a combination of scale-space theory and the Minimum Description Length principle. We show its effectiveness on two time series datasets from real world applications.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*

Keywords

Time Series; Motif Discovery; Multiple Scales; Minimum Description Length;

1. INTRODUCTION

This paper is concerned with the discovery of temporal patterns in large time series produced from physical sensors. In all but the most trivial applications, such sensor data will reflect the complexity of the physical system under investigation and will show a combination of multiple effects. The systems we aim to investigate here often have two important characteristics: a) multiple phenomena are at play in the sensor signal and typically occur at different time scales, b) each phenomenon will involve recurring events that will show up in the signal as repeating segments of data, often deformed and warped. In this paper, we propose a method that elegantly combines these two characteristics in order to discover recurring events at multiple time scales.

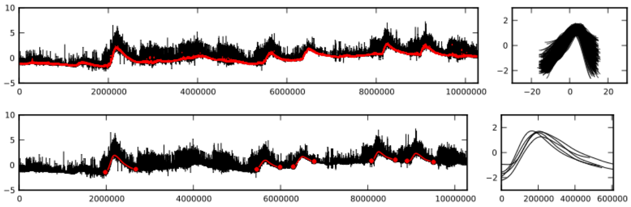

Figure 1: Recurring events in the 12 days highway bridge data. Left side: events occurrences in the series (red). Right side: events shapes after z-normalization.

As a motivating example, we consider InfraWatch [1, 4, 9], a Structural Health Monitoring (SHM) project involving huge quantities of sensor data collected at a major Dutch highway bridge. Such data fits our topic well, as it is subject to a number of effects that show both recurring events (traffic, daily temperature cycles) and largely varying time scales. Figure 1 shows 12 days of strain measurements collected at this bridge (some 10 million readings) and the recurring events present in it: 1) individual peaks due to passing vehicles lasting a few seconds (top) and 2) recurring patterns due to changes of the external temperature (bottom). Note that different effects appear in a mixed fashion, and events at different time scales overlap.

We assume that the recurring events will appear in a relatively small set of classes, which we will refer to as *motifs.* The (scale-aware) motif discovery method presented here will then determine what the relevant motifs are, and when the different instances of each motif occur. In the specification of motifs, we intend to allow for a certain degree of flexibility in terms of duration and magnitude of the event. Moreover, we deal with the possibility of superimposed events as instances of motifs in one scale could overlap with those in other scales and the recognition of similar instances will be disturbed if the multi-scale interference is not taken into account.

We propose an approach based on scale-space images [10] and the *minimum description length* (MDL) principle [3]. The reason for choosing an MDL-based approach is that it allows us to find sets of motifs that represent a good trade-off between representation power and model simplicity. This

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505620.

guarantees that the reported motifs are actual recurring phenomena, rather than accidental coincidences, and that the motifs found are not too similar to each other.

The main contribution of our work is an algorithm for effectively finding multi-scale motifs that score well with respect to the MDL principle. Our algorithm combines several key ideas to achieve this: a) it uses *scale-space images* to characterize the contribution of the motifs at different temporal scales, b) it uses the *zero-crossings of derivatives* of the time series at different scales to identify repeating linear segments in the time series, c) it uses a *a symbolic representation* in combination with suffix trees to identify promising motifs consisting of these linear segments, d) it uses a greedy algorithm to select characteristic motifs that score well with respect to an MDL score.

We evaluate our method on two sensor-based time series from real world applications. Results show that our approach can effectively discover a small set of characteristic motifs in data, often directly related to particular events in the corresponding application domain.

2. BACKGROUND AND PROBLEM SETTING

2.1 Notation and Preliminaries

We deal with finite sequences of numerical measurements (samples), collected by observing some property of a system with a sensor and represented as time series as defined below.

Definition 1. A **time series** of length n is an ordered sequence of values $\mathbf{x} = x[1], \ldots, x[n]$ of finite precision. A subsequence $\mathbf{x}[a:b]$ of $\mathbf{x}$ is defined as follows:

$$\mathbf{x}[a:b] = (\mathbf{x}[a], \mathbf{x}[a+1], \ldots, \mathbf{x}[b]),\ 1 \leq a < b \leq n$$

Moreover, and without loss of generality, we assume that the values are collected at a constant rate, no measurements are missing and that the data has been z-normalized.

As motivated in the introduction, our goal is to find characteristic motifs in the input time series at multiple temporal scales. There are two equivalent ways of looking at motifs. The first is that a motif is a structure that approximately repeats itself in a large number of places in the time series. The second is that a motif is a set of subsequences in the data, each pair of which is similar to each other [6]. We will refer to a structure that is approximately repeated in the data as a *motif*; subsequences of the data in which this motif occurs are referred to as *motif instances.*

An important feature of the motifs that we are looking for is that their instances can be warped or deformed to deal with potential slight variations in the duration and intensity of the events. This motivates our choice to represent motifs using linear segments as follows.

Definition 2. A **motif m** is a sequence of linear segments $[(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)]$, where a_i indicates the length of a segment (the duration) and b_i indicates the difference in value between the begin and end points of the segment.

In principle, higher order polynomials or other more complex functional representations may also be used to represent the segments, but we found that linear segments are simpler, have the advantage of avoiding overfitting, and are accurate enough in most cases.

We will be looking for instances of these motifs in the data.

Definition 3. Given a set of motifs M, let I be a function that maps a motif $\mathbf{m} \in M$ and a segment t of this motif to a set of subsequences of $\mathbf{x}$:

$$I(\mathbf{m}, t) = \{\mathbf{x}[a_{1t} : b_{1t}], ..., \mathbf{x}[a_{kt} : b_{kt}]\},$$

for some $a_{it}, b_{it} \in \{1, \ldots, n\}$ such that $a_{it} = b_{i(t-1)} + 1$ for $t > 1$ (i.e., the end of a segment determines the start of the next segment). Then $I(\mathbf{m}, t)$ determines the set of **instances** in $\mathbf{x}$ of segment t of motif $\mathbf{m}$.

Some choices for I are better than others; ideally instances closely resemble their associated motifs. The MDL score introduced in the next section will be used to evaluate the quality of a set of motifs M and of a function I.

Note that instances for the same motif and motif segment can have different lengths. This is necessary to deal with time warping.

From a high-level perspective, the problem that we are interested in is to identify a set of motifs that characterizes the data well. Taking into account the multi-scale nature of the data, it is desirable that instances of different motifs can overlap. In this way, one motif can reflect a regularity at a coarse scale, and another can reflect a regularity at a finer scale superimposed on top of the coarse structure. The next section defines more precisely how we evaluate a set of motifs and its instances to reflect these requirements.

2.2 Minimum Description Length

Our main idea is to approach the problem of selecting motifs as a *model selection* problem. This allows us to employ the Minimum Description Length [3] principle to rank motifs. MDL is an information-theoretic model selection framework that selects the best model according to its ability to *compress* the given data. In our setting, a model consists of a set of motifs M. Following the two-part MDL principle, the best set of motifs to describe the time series $\mathbf{x}$ is the one that minimizes the sum $L(M) + L(\mathbf{x} \mid M)$. $L(M)$ is the length, in bits, of the description of the motifs, corresponding to a model. $L(\mathbf{x} \mid M)$ is the length, in bits, of the description of the time series when encoded with the help of the motifs M, that is the residual information not represented by M.

In order to apply the MDL principle in practice, we need to define an encoding scheme for a given set of motifs M and, consequently, how to compute both $L(M)$ and $L(\mathbf{x} \mid M)$. However, we need first to clarify how we discretize the time series as the MDL principle is only applicable to discrete data. Both aspects are addressed in the following sections.

2.2.1 Time Series Values Discretization

We assume that the values $\mathbf{x}[i]$ of the input time series $\mathbf{x}$ have been quantized to a finite number of symbols by employing the function defined below:

$$Q(\mathbf{x}[i]) = \left\lfloor \frac{\mathbf{x}[i] - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}\, l \right\rfloor - \frac{l}{2}$$

where l, assumed to be even, is the number of bins to use in the discretization while $\min(\mathbf{x})$ and $\max(\mathbf{x})$ are respectively the minimum and maximum value in $\mathbf{x}$. Throughout the rest of the paper, we assume $l = 256$ as it makes the effect of quantization rather modest in most real-world time series.

2.2.2 Encoding of the Model

We will first discuss the encoding of the model, i.e. a set of motifs M. Each motif essentially consists of a sequence of linear segments, each described by two integers. The length of a segment cannot be longer than the total length of the time series; hence, we use $\log_2 n$ bits to encode it. The difference in value between the begin and end point is limited by the quantization used; in our setting 8 bits are sufficient. Finally, with $\log_2 n$ bits we can encode the number of segments in a motif. Summing up we have

$$L(M) = \sum_{i=1}^{m} (\log_2 n + k_i(\log_2 n + 8)),$$

where m is the number of motifs and k_i is the number of segments in motif i. We assume that these motifs are ordered in the encoding. We use this order to distinguish the scales at which the motifs are present.

2.2.3 Encoding the Data

We will now describe how we compute $L(\mathbf{x} \mid M)$, that is the description length of the time series when encoded with the help of a set of motifs M. In the definition of the code we will also use the instances I associated to each motif in M. Our assumption is that a good selection of motifs M and associated instances I will help to encode the data more concisely.

We will first define the entropy of a time series as it is a key concept we will need in the following paragraphs.

Definition 4. The **entropy** of a time series $\mathbf{x}$, discretized according to a set of values D, is defined as:

$$H(\mathbf{x}) = -\sum_{v \in D} P(\mathbf{x}[i] = v) \log_2 P(\mathbf{x}[i] = v),$$

where $P \log_2 P = 0$ in the case of $P = 0$ and $P(\mathbf{x}[i] = v)$ indicates the fraction of points in the time series which has value v.

Given the definition of entropy and assuming we have not identified any motifs, we can define the **description length** (in bits) of a time series $\mathbf{x}$ of length n as $L(\mathbf{x}) = nH(\mathbf{x})$.

Our main idea is now that a good choice of motifs M and associated instances $I(\mathbf{m}, t)$ should lead to a code for the time series with a description length shorter than $L(\mathbf{x})$. To this aim, we introduce a code for $\mathbf{x}$ given a choice of M and $I(\mathbf{m}, t)$. It will be the task of the search algorithm to determine the best configuration.

Concretely, for each chosen motif $\mathbf{m}$ and corresponding motif instances $I(\mathbf{m}, t)$, we first encode the time stamps and the (vertical) values at which the instances of the first segment of $\mathbf{m}$ start. For one motif with ℓ instances, this requires $\log_2 n + \ell(\log_2 n + 8)$ bits, where $\log_2 n$ bit are needed to encode the number of instances, and $\log_2 n$ and 8 are the bits needed to code the starting time stamp and vertical value, respectively.

There are instances for each segment in a motif. While encoding these, we need to allow for a certain amount of time warping, and hence the segment in each instance may deviate both in length and in amplitude from the segment in the motif. Both vertical and horizontal differences from the segment in the motif can be represented by sequences of integers: the deviations of segment lengths can be represented in one sequence $[a_{ijk} \mid 1 \leq i \leq m, 1 \leq j \leq \ell_i, 1 \leq k \leq k_i]$, where m is the number of motifs, ℓ_i the number of instances of motif i and k_i the number of segments in the motif; similarly, the differences in value can be listed. We encode these sequences in our code. In order to favor only small numbers of values, we compute the description length of these sequences employing an entropy-based encoding as in Definition 5.

This code for motifs and instances leads to an approximation of the data, as follows. For each position in the time series, we determine the last motif in the ordered set of motifs which has an instance at this position. Whether a position is covered by an instance is determined by taking into account the starting positions of the first segment of the motif, the lengths of the other segments in the motifs, and the deviations from these lengths as encoded in the code of deviations. The reason for using the order of motifs is that we explicitly allow motifs to overlap. This allows us to deal with the multi-scale aspects of the data.

The approximated value of a position in the time series covered by a motif is determined by linear interpolation between the two end points of the motif segment in which the position is included. These end points are determined similarly from the encodings of locations, motifs and deviations.

Our remaining code for the data now consists of two parts. First, for each position in the data covered by a motif, the error is encoded with respect to the approximation. An entropy encoding is used for these errors. Second, for the remaining time stamps, which are not covered by a motif, an entropy encoding is used as well to code the original value for that position.

Note that in this code we have a constant number of dictionaries (for duration, difference in value, errors, and remaining original time points). Hence, we do not need to calculate the size of these dictionaries explicitly.

The final description length $L(\mathbf{x} \mid M)$ is given by the sum of the lengths (in bits) of the code components described above.

2.3 Problem Statement

We have now introduced the necessary definitions and background material to state our problem.

> Given a time series $\mathbf{x}$, we want to find a set of motifs M and associated instances, such that the sum $L(M) + L(\mathbf{x} \mid M)$ is minimized.

Clearly, this problem is hard to solve exactly. Hence, in the next section we define a step-wise heuristic algorithm that works well in practice.

3. MOTIF SELECTION ALGORITHM

The proposed heuristic motif discovery algorithm consists of several steps, which will be shown to perform well in the next section. The first steps will identify a set of promising candidate motifs; from these candidates, the last steps select a characteristic subset of the motifs based on the MDL scoring function discussed earlier.

3.1 Finding Candidates Motifs

In this section we describe our candidate motif generation procedure. Several key ideas underly this procedure: a) it uses the *scale-space image* to characterize the contribution of the motifs at different temporal scales, b) it effectively identifies promising segments at multiple scales by discretizing the time series using *the derivatives of the signal in scale-space* in combination with *k-means clustering*, c) in the discretized representation, it merges recurring sequences of adjacent segments by employing a *suffix tree* based approach. The subsequent sections discuss this in more detail.

3.1.1 Scale-Space Image

Scale-space images [10] are a widely used scale parameterization technique for one-dimensional signals[1]. We use them to characterize the contribution of the motifs at increasingly higher temporal scales while, at the same time, removing (smoothing out) the effect of the motifs at finer scales.

Given a signal $\mathbf{x}$, its **scale-space image** is the family of σ-smoothed signals $\Phi_{\mathbf{x}}$ over the scale parameter σ defined as follows:

$$\Phi_{\mathbf{x}}(\sigma) = \mathbf{x} * \mathbf{g}_\sigma \, , \; \sigma > 0$$

where $*$ is the operation of convolution, $\mathbf{g}_\sigma$ is a Gaussian kernel having standard deviation σ, and $\Phi_{\mathbf{x}}(0) = \mathbf{x}$. In [9], we give a more extensive review of scale space images.

We quantize the scale-space image across the scale dimension by considering a fixed set of scale parameters S and computing $\Phi_{\mathbf{x}}(\sigma)$ only for $\sigma \in S$. The number of scale parameters considered, and thus the resolution of the quantization, depends on the final application and on the distribution of the motifs across the scale dimension. In this paper, we define two sets of scale parameters $S_{coarse} = \{2^i \mid 0 \le i \le \sigma_{max} \ \wedge i \in \mathbb{N}\}$ and $S_{fine} = \{\sqrt{2^i} \mid 0 \le i \le 2\sigma_{max} \ \wedge i \in \mathbb{N}\}$ which we found to adapt well to the practical cases we consider.

We deal with the multi-scale aspect of the data by identifying motifs in each of the scales in the scale-space image.

3.1.2 Finding Candidate Segments

Before identifying candidate motifs, we first identify candidate linear segments. A useful tool to quickly identify promising boundaries for linear segments in the time series are the *zero-crossings of derivatives.*

Given a time series $\mathbf{x}$ and one of the components of its scale-space image $\Phi_{\mathbf{x}}(\sigma)$, let $Z = z(1) \cup \cdots \cup z(d_{max})$ with

$$z(j) = \{t_1, \ldots, t_m\} \text{ and } \frac{d^j \Phi_{\mathbf{x}}(\sigma)}{dt^j}(t_i) = 0,$$

be the sorted locations in $\Phi_{\mathbf{x}}(\sigma)$ of the zero-crossing of its derivatives until order d_{max}. Note that d_{max} will typically be low, e.g. just 1 or 2.

These zero-crossings are informative as they indicate points in the time series at which the direction of the signal changes; these positions are good candidates for a change of the linear coefficients as well. Thus, each segment bounded by two consecutive zero-crossings could be an instance of a segment in a motif. We use $k-$means clustering to identify a small set of prototype segments, as follows. Each segment between zero-crossings can be thought of as a data point in a feature space, where the features are the duration and difference in value between the zero-crossings of the derivatives. More precisely, we consider the data points $F_{\Phi_{\mathbf{x}}(\sigma)} = \{\mathbf{f}_i = (h_i, v_i)\}$ where

$$h_i = t_{i+1} - t_i \, , \, 1 \le i < n$$

is the time between each pair of consecutive zero-crossings and

$$v_i = \Phi_{\mathbf{x}}(\sigma)[t_{i+1}] - \Phi_{\mathbf{x}}(\sigma)[t_i] \, , \, 1 \le i < n$$

is their vertical distance.

These data points are clustered using the $k-$means clustering algorithm, where k is a parameter that determines the number of candidate segments. Preliminary experiments show that setting the parameter k in practice is not a critical problem.

The centers of the identified clusters are the candidate reference segments, which will be combined into motifs in the next step. Note that the clustering algorithm ensures that candidate segments will be not too dissimilar from each other. This procedure is repeated for each scale in the scale-space independently.

3.1.3 Finding Candidate Motifs

The key idea in identifying motifs is to represent time series symbolically. Each symbol in this representation corresponds to the candidate segment identified by the $k-$means algorithm for that segment.

After transforming each scale-space image component into the symbolic representation defined above, we identify motifs by looking for repeating subsequences in the obtained string as similarly done by previous approaches [2, 7], although using different representations such as SAX [5].

Algorithm 1 Find candidate motifs

```
Input: a time series $\mathbf{x}$, a set of scales parameters $S = \{\sigma_1, ..., \sigma_k\}$, the maximum order for the derivatives roots $d_{max}$, the cardinality $A$ of the symbolic representation, the number of motifs considered per scale $r$
Output: a set of candidate motifs $\mathcal{M} = \{M_{s,r}\}$ indexed by scale parameter $s$ and rank $r$.
  $\mathcal{M} = \{\}$
  $\Phi_{\mathbf{x}}(\sigma_1), ..., \Phi_{\mathbf{x}}(\sigma_k) = \texttt{ScaleSpaceImage}(\mathbf{x}, S)$
  for $i = 1 \ldots k$ do
    $Z_i = \texttt{ComputeZeroCrossings}(\Phi_{\mathbf{x}}(\sigma_i), d_{max})$
    $S_i = \texttt{SymbolicQuantization}(\Phi_{\mathbf{x}}(\sigma_i), Z_i, A)$
    $\Sigma_i = \texttt{FindRecurringSubstrings}(S_i)$
    $M_{\sigma_i,r_1}, ..., M_{\sigma_i,r_m} = \texttt{RankMotifsByCoverage}(\Sigma_i, r)$
    $\mathcal{M} = \mathcal{M} \cup \{M_{\sigma_i,r_1}, ..., M_{\sigma_i,r_m}\}$
  end for
```

Our candidate motif generation procedure is summarized in pseudo-code in Algorithm 1. $\texttt{ScaleSpaceImage}(\mathbf{x}, S)$ returns the scale-space image of $\mathbf{x}$ defined over the scale parameters S. $\texttt{ComputeZeroCrossings}(\Phi_{\mathbf{x}}(\sigma_i), d_{max})$ calculates the zero-crossings of the derivatives for each scale. $\texttt{SymbolicQuantization}(\Phi_{\mathbf{x}}(\sigma_i), Z, A)$ transforms each time series $\Phi_{\mathbf{x}}(\sigma_i)$ into a symbolic string given the zero-crossings Z and cardinality A. $\texttt{FindRecurringSubstrings}(S_i)$ returns the set of all maximal substrings of length at least 2 that appear at least twice in the data (maximal in the sense that no longer substring occurs twice). In general, we could parameterize this; however, in our experiments we found these parameters to work in all cases. Furthermore, an important advantage of this setup is that we can calculate this set of substrings in linear time by using suffix trees. $\texttt{RankMotifsByCoverage}(\Sigma_i, r)$ selects the best scoring r motifs from this set of substrings. The evaluation is as follows: the occurrences of each string in the time series are deter-

[1] From now on, we will use the term signal and time series interchangeably.

mined; these occurrences are mapped back to the original time series; the total length of the original time series covered by these occurrences is determined. The main motivation is that we can expect the best coding motifs to be those that cover large parts of the time series. The final selection from the resulting set of candidate motifs is done in the next step.

3.2 Selecting Characteristic motifs

The naive way to select the best set of motifs would be to enumerate all potential subsets and choose the one that minimizes the sum $L(M) + L(\mathbf{x} \mid M)$. However, the space of motif sets grows exponentially with the number of candidate motifs and this makes an exhaustive evaluation computationally infeasible for large time series. Because of this, we propose a heuristic selection strategy that overcomes these computational limitations. Our motif selection heuristic is shown in pseudo-code in Algorithm 2.

Algorithm 2 Select characteristic motifs

```
Input: a time series $\mathbf{x}$, a set of candidate motifs $\mathcal{M} = \{M_{s,r}\}$ indexed by scale parameter $s$ and rank $r$.
Output: a set of selected motifs $C \subseteq \mathcal{M}$.
  $C = \{\}$
  for $i = k \ldots 1$ do
    $j = \arg\min_{j \in \{1,\ldots,m\}} L(C \cup \{M_{\sigma_i,j}\}) + L(\mathbf{x} \mid C \cup \{M_{\sigma_i,j}\})$
    $C = C \cup \{M_{\sigma_i,j}\}$
  end for
```

Essentially this algorithm traverses the candidate motifs starting at the coarsest scale and, for each scale, it adds the motif that improves the MDL score the most.

3.3 Computational Complexity

The construction of the scale-space image requires to compute $|S|$ convolutions. This can be done efficiently using the Fast Fourier Transform in $O(|S|\, n \log_2 n)$ time. The computation of the zero-crossing of the derivatizes can be done with a linear scan and thus has $O(n)$ complexity. The complexity of the symbolic transformation, carried out by k-means in $O(Ik\,|Z|)$ time depends on the number of zero-crossings features to cluster which, given a property of the scale-space image [10], can only decrease as the scale is increased; here I is the number of iterations of the $k-$means algorithm. Preliminary experiments even show that the decrease in $|Z_i|$ is exponential. Locating recurring substrings in the symbolic representation can be done in linear time employing a suffix tree; the number of such strings ($|\mathcal{M}|$) is $O(n)$ in the worst case and much smaller in practice. We calculate the instances of the corresponding motifs in $O(n)$ time for each motif identified. Sorting the resulting motifs takes $O(|\mathcal{M}| \log |\mathcal{M}|)$ time. During the final traversal of this set, we need to calculate the MDL score for each intermediate model. This calculation takes $O(|C|n)$ time; note that the size of the dictionaries can be considered constant. Overall, this gives our method a complexity of $O(n \log_2 n + |\mathcal{M}|(\log |\mathcal{M}| + |C|n)$ time.

4. EXPERIMENTAL EVALUATION

In this section, we evaluate our method experimentally, on two real-world sensor datasets and compare our approach with an existing one.

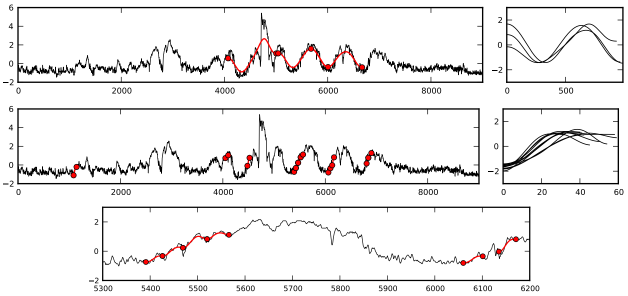

Figure 2: Selected motifs in the Snowboard data. Left side: motif occurrences in the series. Right side: motifs at the respective scale-space component after z-normalization.

4.1 Snowboard Data

The first experiment relates to physiologic data collected during a day of snowboarding in the Austrian Alps. The data considered here describes heart rate measurements taken during 2.5 hours of mixed activity, starting at 11:00 AM, with some 40 minutes actually spent on the slopes. We employed S_{fine} as scale parameters, set $d_{max} = 1$ and the cardinality of the symbolic representation to 10. Figure 2 shows two key selected motifs, which correspond to the phenomena described above. The top motif represents some 16 minutes, corresponding to recuperation (decreasing heart rate while on the lift), exercise and recuperation again. A full cycle of ascent and descent takes about 10 minutes, which corresponds with the manual annotations. This pattern occurs three times in this dataset, at the scale component $\Phi_x(\sqrt{2^{14}})$, as indicated by the red segments in the diagram. Note that two instances actually overlap, as the motif describes more than a single cycle. These two instances actually relate to two descents of a single slope. The second motif, at the scale component $\Phi_x(\sqrt{2^7})$, has 10 instances of increasing and then decreasing heart rate, presumably related to short exercise intervals of around 50 sec. A detail of this motif is shown in the bottom diagram, showing just 20 minutes at 12:25.

The overall number of scale components considered for this data is 22 for a total of 13 selected motifs. However, motifs selected at scales greater than 2^{16} did not show motifs relevant to this particular application domain.

4.2 Highway Bridge Data

We subsequently evaluate our approach on the time series data previously shown in Figure 1. The series, available for download at [1], consists of 12 days of strain measurements (for a total of $10,280,939$ data points) from one span of the monitored highway bridge. As the bridge is affected by several phenomena operating at multiple time scales, the strain measurements contain various classes of recurring motifs reflecting this fact and represents an ideal dataset to test our method. We employed S_{coarse} as scale parameters and set $d_{max} = 1$ and the cardinality of the symbolic representation to 10. Figure 1 shows two of the most interesting selected motifs, respectively at scale components $\Phi_x(2^3)$ and $\Phi_x(2^{15})$. The first motif identifies the most recurring events in the data, i.e. passing vehicles. In the graph, a red pixel is drawn for each instance, for a total of $58,646$ occurrences, which cover almost 22% of the data. On the right, we plot

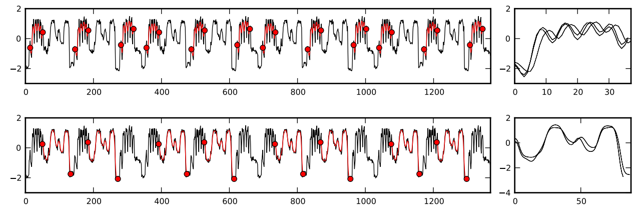

Figure 3: Selected motifs in the *bird calls* data from [8]. Left side: motif occurrences in the series. Right side: motifs at the respective scale-space component after z-normalization.

all the motif instances (after normalization) superimposed, as represented in the scale component $\Phi_x(2^3)$. The selected motif represents a high variability of instances, in both duration and amplitude, that can be directly related to the speed and weight of the vehicles. The second motif represents a much longer pattern occurring on a daily basis due to changes in temperature that, in turn, affect the response of the bridge to external forces. A total of 5 motif instances of this kind occur, covering around 24% of the data. Note how occurrences of the first motif are superimposed over the instances of this one.

The overall number of scale components considered is 19, although the motifs selected at scales greater than 2^{17} are not of any interest in relation to the application domain.

4.3 Comparison with related work

To the best of our knowledge, there are no published methods dealing with the discovery of characteristic sets of multi-scale and overlapping motifs in time series data. As we cannot compare our method with others in a multi-scale setting, we chose to also evaluate our algorithm on a time series presented in [8], in which no multi-scale events are present. A comparison on such data is of interest as our method should be able to identify the non-overlapping motifs present in this data as well.

The considered time series was produced by extracting the first MFCC coefficients from an audio file featuring two repeated kinds of bird calls, resulting in two motifs present in the data. The time series has a total of 1367 measurements. As the motifs in the data are rather similar in length, we do not need to consider the whole scale-space image. Instead, we set the scale parameters to $S = \{1, \sqrt{2}, 2, \sqrt{8}\}$. The result shown here was obtained by setting the cardinality of the symbolic representation to 6. However, in order to assess the sensitivity of the method in relation to the size of the alphabet, we tried cardinalities ranging from 5 to 15 obtaining qualitatively similar results. Figure 3 reports the motifs selected by our method. These motifs are similar to those obtained by the clustering method proposed in [8] for non-overlapping motifs. Although in this case we manually specified the scale parameters, we note that the algorithm in [8] also requires to provide an educated guess of parameters, i.e. of the approximate lengths of the motifs to look for.

5. RELATED WORK

Although several papers address the problem of discovering recurring patterns in time series (see [2] for a survey on motif discovery techniques), none of them, to the best of our knowledge, consider data where combinations of effects at multiple temporal scales affect the patterns or motifs. The most similar work to ours is [8], where the authors propose a method to mine a set of clusters of motifs from a given time series. The clusters are formed according to an agglomerative procedure that iteratively refines the discovered clusters until a MDL-based stopping criteria is met. The motif discovery algorithm introduced in [6] is employed to generate the candidate motifs. This method does not however consider superimposed motifs like those present in the multi-scale data we consider in this paper.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a method to discover recurring and potentially overlapping patterns (motifs) in time series sensor data at multiple temporal scales. To cope with the high degree of variability present in sensor data, we have adopted a definition of motif based on structural complexity other than on point-wise similarity (i.e. Euclidean distance) as in much previous work. In order to discover the most characteristic recurring motifs, we proposed an algorithm based on a combination of scale-space theory, string processing and the Minimum Description Length principle. We showed the effectiveness of our method on sensor data from two real world applications.

Future work includes evaluating our method on additional data exhibiting multi-scale behavior, as a few datasets of this kind are currently publicly available.

7. REFERENCES

[1] Infrawatch project website. `http://www.infrawatch.com`.

[2] P. Esling and C. Agon. Time-series data mining. *ACM Comput. Surv.*, 45(1):12:1–12:34, Dec. 2012.

[3] P. Grünwald. *The minimum description length principle*. The MIT Press, 2007.

[4] A. Knobbe et al. InfraWatch: Data management of large systems for monitoring infrastructural performance. *Advances in Intelligent Data Analysis IX*, pages 91–102, 2010.

[5] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Min. Knowl. Discov.*, 15(2):107–144, Oct. 2007.

[6] A. Mueen et al., E. Keogh, Q. Zhu, S. Cash, and B. Westover. Exact discovery of time series motifs. In *Proceedings of SDM '09*, pages 473–484, 2009.

[7] D. Patnaik et al. Sustainable operation and management of data center chillers using temporal data mining. In *Proceedings of KDD '09*, pages 1305–1314, 2009.

[8] T. Rakthanmanon, E. Keogh, S. Lonardi, and S. Evans. Time Series Epenthesis: Clustering Time Series Streams Requires Ignoring Some Data. In *Proceedings of ICDM '11*, 2011.

[9] U. Vespier, A. Knobbe, S. Nijssen, and J. Vanschoren. MDL-Based Analysis of Time Series at Multiple Time-Scales. In *Proceedings of ECML PKDD '12*, 2012.

[10] A. Witkin. Scale-space filtering. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 329–332, 1987.