Mining sensor data from complex systems

Vespier, U.

Author: Vespier, Ugo
Title: Mining sensor data from complex systems
Issue Date: 2015-12-15

Chapter 4

Identifying the Relevant Temporal Scales

4.1 Introduction

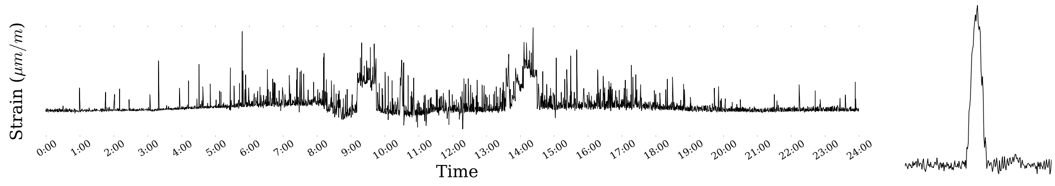
When monitoring complex physical systems over time, one often finds multiple phenomena in the data that work on different time scales. If one is interested in analyzing and modeling these individual phenomena, it is crucial to recognize these different scales and separate the data into its underlying components. In this chapter, we present a method for extracting the time scales of various phenomena present in large time series. The method combines concepts from the signal processing domain with feature selection and the Minimum Description Length principle [34] which we introduced in Chapter 3.

We introduced the need for analyzing time series data at multiple time scales in Chapter 1 and we discussed in Chapter 2 how this is nicely demonstrated by the InfraWatch project.

In this project, we employ a range of sensors to measure the dynamic response of the Hollandse Brug, a large Dutch highway bridge to varying traffic and weather conditions. When viewing this data (see Fig. 4.1a), one can easily distinguish various *transient events* in the signal that occur on different time scales. Most notable are the gradual change in strain over the course of the day (as a function of the outside temperature, which influences stiffness parameters of the concrete), a prolonged increase in strain caused by rush hour traffic congestion, and individual bumps in the signal due to cars and trucks traveling over the bridge. In order to understand the various changes in the sensor signal, one would benefit substantially from separating out the events at various scales. The main goal of the work described here is to do just that: we consider the temporal data as a series of superimposed effects at different time scales, establish at which scales events most often occur, and from this we extract the underlying signal components.

We approach the scale selection problem from a Minimum Description Length (MDL) perspective (see Section 3.4). The motivation for this is that we need a framework in which we can deal with a wide variety of representations for scale components. The MDL framework was shown to be sufficiently general to provide this flexibility by Hu et al. [42] for the problem of choosing the best model for a given signal. Our main assumption here is that separating the original signal into components at different time scales will simplify the shape of the individual components, making it easier to model them separately. Our results show that, indeed, these multiple models outperform (in terms of MDL score) a single model derived from the original signal. While introducing multiple models incurs the penalty of having to describe these multiple models, there are much fewer 'exceptions' to be described compared to the single model, yielding a lower overall description length. For instance, in the sensor data of Fig. 4.1a, cars are often passing in one direction while there is rush hour congestion in the opposite direction. Using multiple models, this is modeled accurately, while a single model will easily ignore these events.

As we discussed in detail in Section 3.3, the analysis of time scales in time series data is often approached from a *scale-space* perspective, which involves convolution of the original signal with Gaussian kernels of increasing size [103] to remove information at smaller scales. By subtracting carefully selected components of the scale-space, we can effectively cut up the scale space into k ranges. In other words, signal processing offers methods for producing a large



(a)

(b)

Figure 4.1: (a) One day of strain measurements from a large highway bridge in the Netherlands. The multiple external factors affecting the bridge are visible at different time scales. (b) A detail of plot (a) showing one of the peaks caused by passing vehicles.

collection of derived features, and the challenge we face in this chapter is how to select a subset of k features, such that the original signal is decomposed into a set of meaningful components at different scales.

Our approach applies the MDL philosophy to various aspects of modeling: choosing the appropriate scales at which to model the components, determining the optimal number of components (while avoiding overfitting on overly specific details of the data), and deciding which class of models to apply to each individual component. For this last decision, we propose two classes of models representing the components respectively on the basis of a discretization and a segmentation scheme. For this last scheme, we allow three levels of complexity to approximate the segments: piecewise constant approximations, piecewise linear approximations, as well as quadratic ones. These options result in different trade-offs between model cost and accuracy, depending on the type of signal we are dealing with.

A useful side product of our approach is that it identifies a concise representation of the original signal. This representation is useful in itself: queries run on the decomposed signal may be answered more quickly than when run on the original data. Furthermore, the parameters of the encoding may indicate useful properties of the data as well.

The rest of the chapter is organized as follows. Section 4.2 introduces the concept of scale-space decomposition. Section 4.3 shows how we encode the signal decompositions and use MDL to select the best subset of scales. Section 4.4 presents an empirical evaluation of our method on both real-world and artificial data. Section 4.5 links our method to related work. Finally, Section 4.6 states our main conclusions and ideas for future work.

4.2 Scale-Space Decomposition

In this section, we show how to manipulate the scale-space image to filter out the effects of transient events in a specific range of scales. This will lead to the definition of a signal decomposition scheme.

Along the scale dimension of the scale-space image, short-time transient events in the signal will be smoothed away sooner than longer ones. In other words, we can associate with each event a maximum scale σ_{cut} such that, for $\sigma > \sigma_{cut}$, the transient event is no longer present in $\Phi_{\mathbf{x}}(\sigma_{cut})$. This fact leads to the following two observations:

- Given a signal scale-space image $\Phi_{\mathbf{x}}$, the signal $\Phi_{\mathbf{x}}(\sigma)$ is only affected by the transient events at scales greater than σ. This is conceptually equivalent to a *low-pass filter* in signal processing.
- Given a signal scale-space image $\Phi_{\mathbf{x}}$ and two scales $\sigma_1 < \sigma_2$, the signal $\Phi_{\mathbf{x}}(\sigma_1) - \Phi_{\mathbf{x}}(\sigma_2)$ is mostly affected by those transient events present in the range of scales (σ_1, σ_2). This is similar to a *band-pass filter* in signal processing.

As an example, reconsider the signal $\mathbf{x}$ and its scale-space image $\Phi_{\mathbf{x}}$ of Figure 3.2. Figure 4.2 shows (from top to bottom):

- the signal $\Phi_{\mathbf{x}}(0) - \Phi_{\mathbf{x}}(2^4)$, which is the result of a high-pass filtering; this feature represents the short-term events (peaks),
- the signal $\Phi_{\mathbf{x}}(2^4) - \Phi_{\mathbf{x}}(2^{10})$, which is the result of a band-pass filtering; this feature represents the medium-term events (bumps),
- the signal $\Phi_{\mathbf{x}}(2^{10})$, which is the result of a low-pass filtering; this feature represents the long-term trend.

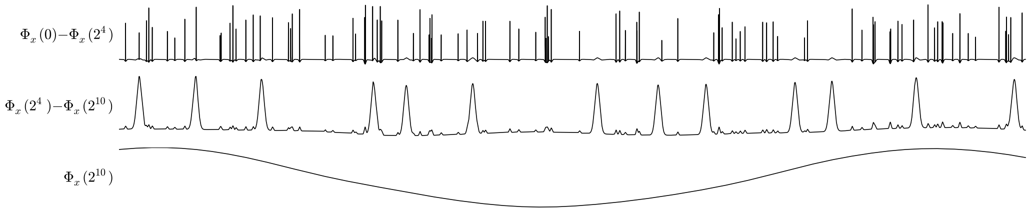


Figure 4.2: Examples of signal decomposition obtained from the scale-space image in Figure 3.2.

Generalizing the example in Figure 4.2, we can define a decomposition scheme of a signal $\mathbf{x}$ by considering adjacent ranges of scales of the signal scale-space image. We formalize this idea below.

Definition (Scale-Space Decomposition). Given a signal $\mathbf{x}$ and a set of $k-1$ scale parameters $C = \{\sigma_1, \ldots, \sigma_{k-1}\}$ (called the cut-points set) such that $\sigma_1 < \ldots < \sigma_{k-1}$, the **scale decomposition** of $\mathbf{x}$ is given by the set of component signals $D_{\mathbf{x}}(C) = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, defined as follows:

$$\mathbf{x}_i = \begin{cases} \Phi_{\mathbf{x}}(0) - \Phi_{\mathbf{x}}(\sigma_1) & \text{if } i = 1 \\ \Phi_{\mathbf{x}}(\sigma_{i-1}) - \Phi_{\mathbf{x}}(\sigma_i) & \text{if } 1 < i < k \\ \Phi_{\mathbf{x}}(\sigma_{k-1}) & \text{if } i = k \end{cases}$$

Note that for k components we require $k-1$ cut-points. This decomposition has several elegant properties:

- $\mathbf{x}_k$ can be seen as the baseline of the signal, as obtained by a low-pass filter;
- $\mathbf{x}_i$ for $1 \leq i < k$ are signals as obtained by a band-pass filter, and can be used to identify transient events;
- $\sum_{i=1}^{k} \mathbf{x}_i = \mathbf{x}$, i.e., the original signal can be recovered from the decomposition.

4.3 MDL Scale Decomposition Selection

Given an input signal $\mathbf{x}$, the main computational challenge we face is twofold:

- find a good subset of cut-points C such that the resulting k components of the decomposition $D_{\mathbf{x}}(C)$ optimally capture the effect of transient events at different scales,
- select a representation for each component, according to its inherent complexity.

As stated before, the rationale behind the scale decomposition is that it is easier to model the effect of a single class of transient events at a given scale than to model the superimposition of many, interacting transient events at multiple scales. We thus need to trade off the added complexity of having to represent multiple components for the complexity of the representations themselves.

We approach this model selection problem by using the Minimum Description Length (MDL) principle introduced in Section 3.4. The possible candidate models depend on the scale decomposition $D_{\mathbf{x}}(C)$ considered[1] and on the representations used for its individual components. An ideal set of representations would adapt to the specific features of every single component, resulting in a concise summarization of the decomposition and, thus, of the signal. In order to apply the MDL principle, we need to define a model $M_{D_{\mathbf{x}}(C)}$ for a given scale decomposition $D_{\mathbf{x}}(C)$ and, consequently, how to compute both $L(M_{D_{\mathbf{x}}(C)})$ and $L(\mathbf{x} \mid M_{D_{\mathbf{x}}(C)})$. The latter term is the length in bits of the information lost by the model, i.e., the residual signal $\mathbf{x} - M_{D_{\mathbf{x}}(C)}$.

As the MDL framework is only applicable to discrete data, we assume that the input signal $\mathbf{x}$ and the results of all the subsequent operations are discretized as discussed in Section 3.4.1. In the next sections, we introduce the proposed representation schemes for the components and define the bit complexity of the residual and the model selection procedure.

[1] Including the decomposition formed by zero cut-points ($C = \emptyset$), i.e., the signal itself.

4.3.1 Component Representation Schemes

Within our general framework, many different approaches could be used for representing the components of a decomposition. In the next paragraphs we introduce two such methods.

Discretization-based representation

In some components of our data transient events always occur with similar amplitudes, mixed with long stretches of baseline values (see Figure 4.2). Hence, a desirable encoding could be one that captures this repetitiveness in the data by giving short codes to long stretches of the baseline and the commonly occurring amplitudes. Unfortunately, our original discretization is too fine-grained to capture regular occurrences of similar amplitudes. As a first representation, we hence propose to also consider more coarse-grained discretizations of the original range of values. We do this by discretizing each value v in a component to a value $\lfloor Q(v)/2^i \rfloor$, where several values for i are considered for each component, typically $i \in \{2, 4, 6\}$. By doing so, similar values will be grouped together in the same bin. The resulting sequence of integers is compacted further by performing run-length encoding, resulting in a string of (v, l) pairs, where l represents the number of times value v is repeated consecutively. This string is finally encoded using a Shannon-Fano or Huffman code (see Section 4.3.2).

As a simplified illustration of how the MDL principle helps here to identify components, consider data generated by the expression $(67)^n(01)^n$ ($4n$ integers from the range $\{0, \ldots, 2^3 - 1\}$), where we assume n and the range are fixed. In this data, each symbol occurs with the same frequency; we can encode the time series hence with $-\log_2(1/4) \cdot 4 \cdot n = 8n$ bits for the data, plus $8 \log n$ bits for the dictionary of frequencies. Consider now the decomposition of the signal into two time series, $6^{2n}0^{2n}$ and $(01)^{2n}$. The first component, of which the run-length encoding is $(6, 2n)(0, 2n)$, can be encoded using only 2 bits for the time series (as there is only one possible run-length value, we use 0 bits to encode the run-lengths), $8 \log n$ bits for the dictionary of amplitudes, and $3 \log n$ bits to identify the length of the one run-length ($\log n$ bit for identifying the number of run-lengths, in this case one, $\log n$ to identify the one run-length present, and $\log n$ to identify its frequency, from which the encoding with 0 bits follows). The second component can be encoded using $4n$ bits for the time series, as well as $8 \log n$ bits for the dictionary. Assuming we also use 1 bit per component to identify the type of encoding used, this gives us an encoding in $4 + 19 \log n + 4n$ bits. Comparing this to $8n + 8 \log n$ bits, for $n \geq 11$ we will hence correctly identify the two components in this simplified data.

Segmentation-based representation

The main assumption on which we base this method is that a clear transient event can be accurately represented by a simple function, such as a polynomial of a bounded degree. Hence, if a signal contains a number of clear transient events, it should be possible to accurately represent this signal with a number of segments, each of which represented by a simple function.

Given a component $\mathbf{x}_i$ of length n, let

$$z(\mathbf{x}_i) = \{t_1, t_2, \ldots, t_m\}, \quad 1 < t_i \leq n$$

be a set of indexes of the segment boundaries.

Let $\texttt{fit}(\mathbf{x}_i[a : b], d_i)$ be the approximation of $\mathbf{x}_i[a : b]$ obtained by fitting a polynomial of degree d_i. Then, we represent each component $\mathbf{x}_i$ with the approximation $\hat{\mathbf{x}}_i$, such that:

$$\begin{aligned}
\hat{\mathbf{x}}_i[0 : z_1] &= \texttt{fit}(\mathbf{x}_i[0 : z_1], d_i) \\
\hat{\mathbf{x}}_i[z_i : z_{i+1}] &= \texttt{fit}(\mathbf{x}_i[z_i : z_{i+1}], d_i), \quad 1 \leq i < m \\
\hat{\mathbf{x}}_i[z_m : n] &= \texttt{fit}(\mathbf{x}_i[z_m : n], d_i)
\end{aligned}$$

Note that approximation $\hat{\mathbf{x}}_i$ is quantized again by reapplying the function Q to each of its values.

For a given k-components scale decomposition $D_{\mathbf{x}}(C)$ and a fixed polynomial degree for each of its components, we calculate the complexity in bits of the model $M_{D_{\mathbf{x}}(C)}$, based on this representation scheme, as follows. Each approximated component $\hat{\mathbf{x}}_i$ consists of $|z(\mathbf{x}_i)| + 1$ segments. For each segment, we need to represent its length and the $d_i + 1$ coefficients of the fitted polynomial. The length ls_i of the longest segment in $\hat{\mathbf{x}}_i$ is given by

$$ls_i = \max(z_1 \cup \{z_{i+1} - z_i \mid 0 < i \leq m\})$$

We therefore use $\log_2(ls_i)$ bits to represent the segment lengths, while for the coefficients of the polynomials we employ floating point numbers of fixed[2] bit complexity c. The MDL model cost is thus defined as:

$$L(M_{D_{\mathbf{x}}(C)}) = \sum_{i=1}^{k} \left(|z(\mathbf{x}_i)| + 1\right) \left(\lceil \log_2(ls_i) \rceil + c\,(d_i + 1)\right)$$

So far we assumed to have a set of boundaries $z(\mathbf{x}_i)$, but we did not specify how to compute them. A desirable property for our segmentation would be that a segmentation at a coarser scale does not contain more segments than a segmentation at a finer scale. The scale space theory assures that there are fewer zero-crossing of the derivatives of a signal at coarser scales [103]. In our segmentation, we use the zero-crossings of the first and second derivatives. More formally, we define the segmentation boundaries of a component $\mathbf{x}_i$ to be

$$z(\mathbf{x}_i) = \left\{ t \in \mathbb{R} \;\middle|\; \frac{d\mathbf{x}_i}{dt}(t) = 0 \right\} \bigcup \left\{ t \in \mathbb{R} \;\middle|\; \frac{d^2\mathbf{x}_i}{dt}(t) = 0 \right\}.$$

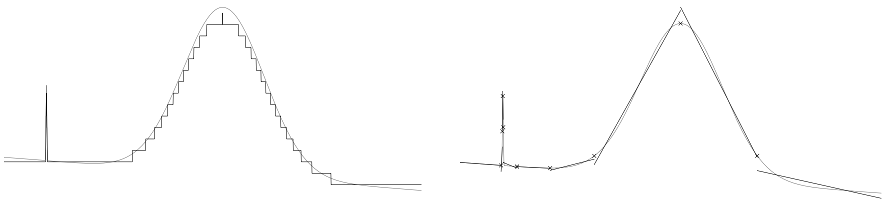
Figure 4.3b shows an example of segmentation obtained as above using fitted polynomials of degree 1.

However, many other segmentation algorithms are known in the literature [48, 53] and all of them can be interchangeably employed in this context.

4.3.2 Residual Encoding

Given a model $M_{D_{\mathbf{x}}(C)}$, its residual $\mathbf{r} = \mathbf{x} - \sum_{i=1}^{k} \hat{\mathbf{x}}_i$, computed over the components approximations, represents the information of $\mathbf{x}$ not captured

[2] In our experiments $c = 32$.



(a)

(b)

Figure 4.3: Example of discretization-based encoding (a) and segmentation-based encoding with first degree polynomial approximations (the markers show the zero-crossings) (b).

by the model. Having already defined the model cost for the two proposed encoding schemes, we only still need to define $L(\mathbf{x} \mid M_{D_{\mathbf{x}}(C)})$, i.e., a bit complexity $L(\mathbf{r})$ for the residual $\mathbf{r}$.

Here, we exploit the fact that we operate in a quantized space; we encode each bin in the quantized space with a code that uses approximately $-\log(P(x))$ bits, where $P(x)$ is the frequency of the xth bin in our data. The main justification for this encoding is that we expect that the errors are normally distributed around 0. Hence, the bins in the discretization that reflect a low error will have the highest frequency of occurrences; we will give these the shortest codes. In practice, such codes can be obtained by means of Shannon-Fano coding or Huffman coding; as Hu et al. [42] we use Huffman coding in our experiments.

4.3.3 Model Selection

We can now define the MDL score that we are optimizing as follows:

Definition (MDL Score). Given a model $M_{D_{\mathbf{x}}(C)}$, its **MDL score** is defined as:

$$L(M_{D_{\mathbf{x}}(C)}) + L(\mathbf{r})$$

In the case of discretization-based encoding, the MDL score is affected by the cardinality used to encode each component. In the case of segmentation-based encoding the MDL score depends on the boundaries of the segments and the degrees of the polynomials in the representation. In both cases, also the cut-points of the considered decomposition affect the final score.

The simplest way to find the model that minimizes this score is to enumerate, encode and compute the MDL score for every possible scale-space decomposition and all possible encoding parameters. As we shall now show, this brute-force approach is practically feasible.

The number of possible scale decompositions depends on the total number of cut-points sets we can build from the computed scale parameters in $\Phi_{\mathbf{x}}$. We fix the maximum number of cut-points in a candidate set to some value c_{max}. This also means that we limit our search to those scale decompositions having $c_{max} + 1$ components or less. Moreover, given our wish to consider only simple approximations of the signals, we can also assume a reasonably low limit d_{max} (in practice, $d_{max} = 2$) on the degree of the polynomials that approximate the segments of each given component.

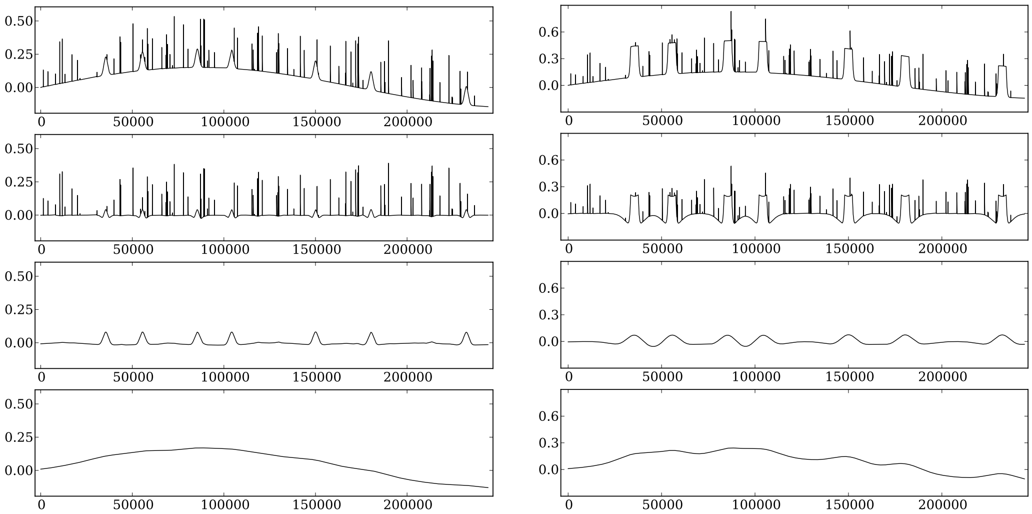
Computing the MDL score for each encoded scale decomposition, obtained by ranging over all the possible configurations of cut-points $C_1, \ldots, C_{k-1}$, and all the possible configurations of polynomial degrees $d_1, \ldots, d_k$, hence requires calculating MDL scores for

$$\sum_{k=2}^{c_{max}+1} \binom{|\mathbf{S}|}{k-1} d_{max}^{k}$$

scale decompositions. This turns out to be a reasonable number in most practical cases we consider, and hence we use an exhaustive approach in our experiments.

4.4 Experiments

In this section, we experimentally evaluate our method, both on artificial data and on actual sensor data from the highway bridge mentioned in the introduction. To evaluate the strengths and weaknesses of our method, we have



(a) (b)

Figure 4.4: Signals (top) and top-ranked decompositions for the two artificial datasets.

tested it on a range of artificial datasets[3] that mimic some of the multi-scale phenomena present in the bridge data. Our constructed data deliberately varies from easy, with clearly separated scales, to challenging with a variety of event shapes and sizes. All artificial datasets represent sensor data measured at 1 Hz for a duration of three days (totaling 259,200 data points). The data was produced by combining three components at three distinct scales, resembling 1) individual events from vehicles, 2) traffic jams that last several tens of minutes, and 3) gradual change of the baseline, due to temperature changes of the bridge over the course of several days.

Artificial data

We start by considering one particular dataset in detail (see Figure 4.4a). This dataset was constructed by using Gaussian shapes for both the small and medium-scale events, and a sine wave of period 2.25 days at the largest

[3] The artificial datasets and the source code can be obtained by contacting the first author.

scale. Medium events have a constant height, whereas small-scale events have a random height. We limited the search space to decompositions having a maximum of 4 components (3 cut-points). As can be seen in Figure 4.4a, our method was able to identify the fact that this data contains three important scales. Furthermore, the method correctly identified the two necessary cut-points, such that the three original components were reconstructed. The selected cut-points[4] appear at scales $2^9 = 512$ and $2^{12} = 4096$. When considering the separated components in detail, some influence across the scale-boundaries is visible, for example where small effects of the 'traffic jams' appear among the small-scale events. These effects seem unavoidable, with the inherent limitations of the scale-space-based band-pass filtering and the discrete collection of scales we consider (powers of 2).

This optimal result has an MDL-score of 509,000 bits, being the sum of the model cost ($L(M) = 75,072$) and the error length ($L(D \mid M) = 433,928$). The second-ranked result on this data, with cut-points $C = \{2^{11}, 2^{13}\}$, shows a similar result, however with slightly more pronounced cross-boundary artifacts in the smallest scale, as is expected with a doubling of the lower cut-point. The MDL-score of this result is $64,896 + 450,487 = 515,383$. The $k = 1$ case, which corresponds to compression of the original signal without any decomposition, appears at rank three, with an MDL-score of $44,640 + 471,271 = 515,911$. This model obviously has a much lower model cost, due to having to represent only a single component, but this is compensated by the substantially higher error length, putting it below the scale-separated results. Ranks four and five represent two $k = 2$ results, where the former groups the small and medium scales together, and the latter the medium and large. All results in the top 10 relate to models that use polynomial representations ($d \leq 2$).

Not all artificial datasets considered produced perfect results. In Figure 4.4b, we show an example of a dataset that includes 'traffic jams' that resemble more closely some of the phenomena in the actual sensor data. In many cases, traffic jams appear fairly rapidly, and then show an increased load on

[4] Note that our method returns the boundaries between scales, rather than the actual scales of the original components.

the bridge over a prolonged period. This is modeled in the data by medium-scale events that start and stop fairly rapidly, and remain constant in the meantime. The best result found, with cut-points $C = \{2^{12}, 2^{13}\}$, is shown in Figure 4.4b. This demonstrates that the proposed method is not able to properly separate the medium and low-scale events. In fact, even though the medium component does identify the location of the 'traffic jams', most of the rectangular nature is accounted for by the small scale. To some extent, this is understandable, as the start and end of the event could be considered high-frequency events with rapid changes in value. Therefore, parts of these events appear at a small scale, and the algorithm is mirroring this effect. In any case, the algorithm *is* able to identify the correct number of components, and is able to produce indications as to the location of the traffic jams. The top four results all show similar mixtures of scales, whereas the rank-five result groups the lowest two scales together. The $k = 1$ result appears at rank 14.

In order to better understand to what extent the proposed method is able to separate components at different scales, we carried out a more controlled experiment. We generated 11 different datasets constructed from 3 components. We fixed the scales of the short-term and long-term components respectively around $\sigma = 2^3$ and $\sigma = 2^{15}$, while the scale of the medium-term component varies from dataset to dataset in the range $(2^4, \ldots, 2^{14})$. The table below shows the number of components (k) of the top-ranked decomposition for the 11 datasets according to the scale parameter σ of the medium-term component.

σ	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
k	1	2	2	2	3	3	3	3	1	1	1

As the table suggests, the proposed method fails to identify the right number of components when the scales are too close to each other. However, when the scales are separated sufficiently ($2^8 \leq \sigma \leq 2^{11}$), the right number of components is identified. Also in this case, all the top-ranked decompositions relate to models that use polynomial representations.

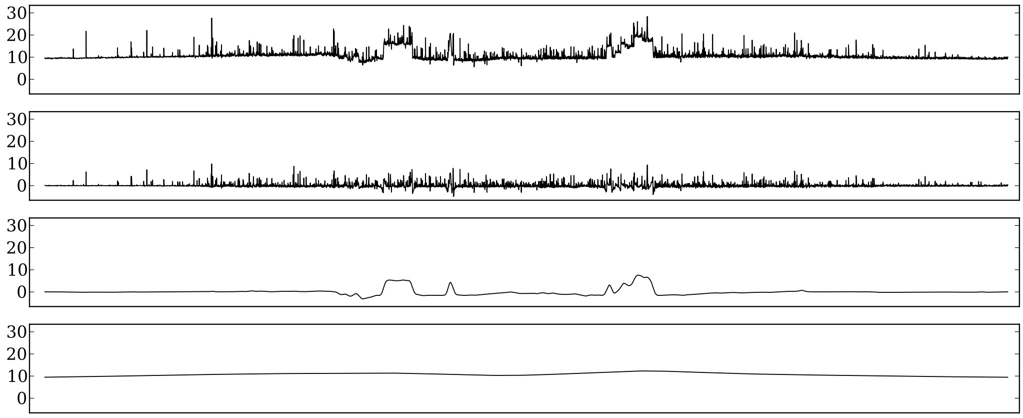


Figure 4.5: Signal (top) and top-ranked scale decomposition for the InfraWatch data.

InfraWatch data

As anticipated by the motivating example in the introduction, we consider the strain measurements produced by a sensors attached to a large highway bridge in the Netherlands. For this purpose, we consider a time series consisting of 24 hours of strain measurements sampled at 1 Hz (totaling $86,400$ data points). A plot of the data is shown in Figure 4.5 (topmost plot). We evaluated all the possible decompositions up to three components (two cut-points) allowing both the representation schemes we introduced. In the case of the discretization-based representations, we limit the possible cardinalities to 4, 16 and 64.

The top-ranked decomposition results in 3 components as shown in the last three plots in Figure 4.5. The selected cut-points appear at scales $2^6 = 64$ and $2^{11} = 2048$. All three components are represented with the discretization-based scheme, with a cardinality of respectively 4, 16, and 16 symbols. The decomposition has an MDL-score of $344,276$, where $L(M) = 19,457$ and $L(D \mid M) = 324,818$. The found components accurately correspond to physical events on the bridge. The first component, covering scales lower than 2^6, reflects the short-term influence caused by passing vehicles and represented as peaks in the signal. Note that the cardinality selected for this component is the lowest admissible in our setting (4). This is reasonable considering that

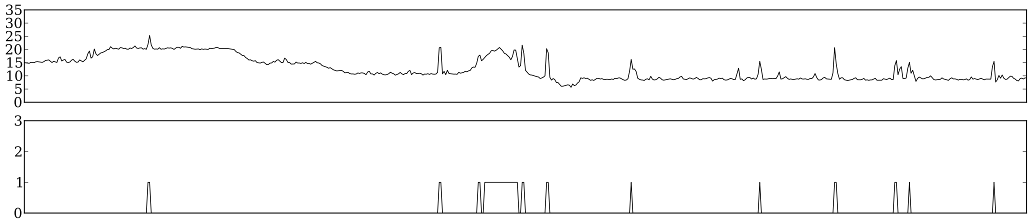


Figure 4.6: A detail of the original strain signal (one hour) and the selected first component as represented with 4 symbols.

the relatively simple dynamic behavior occurring at these scales, mostly the presence or not of a peak over a flat baseline, can be cheaply described with 4 or fewer states without incurring a too large error. The middle component, covering scales between 2^6 and 2^{11}, reflects the medium-term effects caused by traffic jams. As in the artificial data, the first component is slightly influenced by the second one, especially at the start and ending points of a traffic jam. Finally, the third component captures all the scales greater than 2^{11}, here representing the effect of temperature during a whole day. To sum up, the top-ranked decomposition successfully reflects the real physical phenomena affecting the data. The decompositions with rank 8 or less all present similar configurations of cut-points and cardinalities, resulting in comparable components where the conclusions above still hold. The first 2-component decomposition appears at rank 10 with the cut-point placed at scale 2^6, which separates the short-term peaks from all the rest of the signal (traffic jams and baseline mixed together). These facts make the result pretty stable as most of the good decompositions are ranked first.

Dike monitoring data

We evaluated the method on an additional real-world scenario: the sensor-based monitoring of dikes [20]. We focus on the data produced by a network of pore sensors installed on a sea dike in Boston (UK). The considered data consists of one time series, representing one year of measurements sampled every 15 minutes (for a total of 27 610 values), ranging from October 2011 to October 2012. As many complex physical systems, also dikes are affected

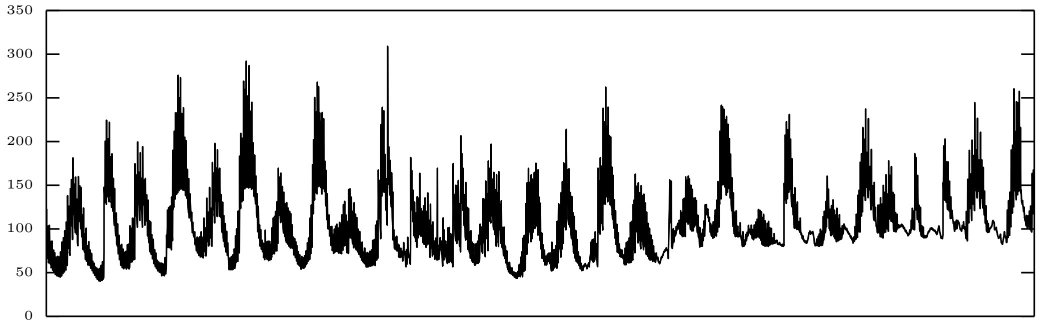


Figure 4.7: One year of pore pressure measurements from the one of the sensors installed on a sea dike in Boston (UK).

by multiple phenomena at different temporal scales and the effect is visible in the produced sensor data. Figure 4.7 shows one year of measurements from one of the pore sensors in the network. The sensor exhibits two clear periodic trends at different temporal scales. At the shortest temporal scale, the water level follows half-daily tides (see the detailed plots of Figure 4.7 and Figure 4.8). The water level, however, is also affected, on a two-weekly basis, by the lunar cycles that have a periodic effect on the overall amplitude of the signal.

For the considered time-series, we evaluated all the possible decompositions up to three components (two cut-points) employing the discretization-based scheme with possible cardinalities limited to 4, 8, and 16.

The top-ranked decomposition for the time series is shown in Figure 4.9. The selected cut-points appear at scales $2^5 = 32$ and $2^{13} = 8192$. All three components, represented with the discretization-based scheme, have a selected cardinality of 8 symbols. The decomposition has an MDL-score of $213,288$, where $L(M) = 19,075$ and $L(D \mid M) = 194,213$. Note that the first two components, from top to bottom, effectively separate the two periodic trends in the data, i.e. the half-daily tidal effects (a detail of which can be seen in Figure 4.10) and the two-weekly cycles due to lunar tides. The third long-term component reflects slow change happening in the dike due to the effect of humidity. However, the trend also includes a slight change in the sensor sensitivity itself, as its response is affected by the external temperature.

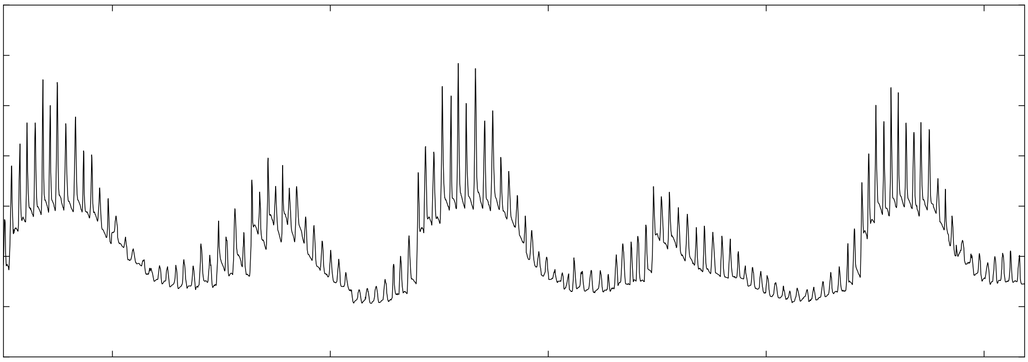


Figure 4.8: Zoomed detail of Figure 4.7. The figure clearly shows the presence of short term periodic trends, due to the half-daily effect of the tides.

An application: detecting passing vehicles

The component selection and representation generated by the MDL procedure may be useful in itself for tasks such as classification. For example, consider the short-term component of the previous example, Figure 4.5 (second plot). It represents the traffic activity over the bridge and has been represented with a discretization-based scheme using 4 symbols. Figure 4.6 shows a detail (1 hour) of the discretized component (bottom) and the relative original signal (top). The first 2 symbols (0 and 1) respectively classify the absence or presence of a passing vehicle, while the other two, considerably less frequent, are outliers in the data. The represented component, as selected by MDL, can thus be used to monitor traffic activity over the bridge, a task that is considerably more challenging using the original signal, due to the variations introduced by temperature fluctuations and traffic jams.

4.5 Related Work

Papadimitriou et al. [76] propose a method to discover the key trends in a time series at multiple time scales (window lengths) by defining an incremental version of Singular Value Decomposition. In signal processing, Indepen-

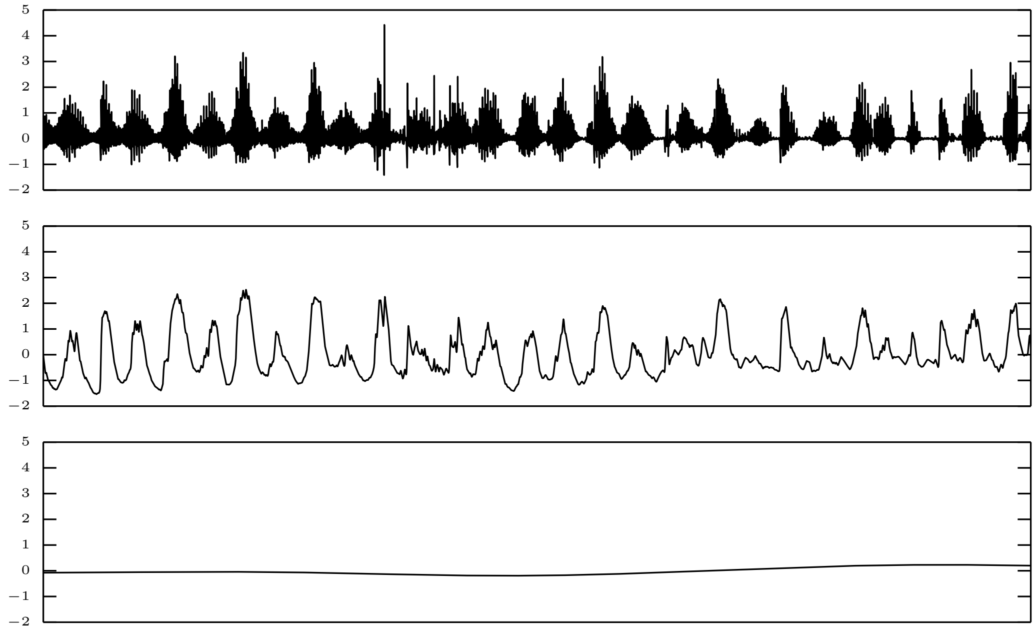


Figure 4.9: Top-ranked scale decomposition for the first sensor of the Dike data.

dent Component Analysis [18] aims at separating a set of signals from a set of mixed signals but, in its standard formulation, requires at least as many sensors as sources. Our method is able to operate on a single input sensor and a variable number of sources to be discovered. Megalooikonomou et al. [69] introduce a multi-scale vector quantized representation of time series which enables fast and robust retrieval. The considered scales are however predefined and our approach could be used as a preprocessing step to determine those to include in the dictionary. The Minimum Description Length principle has been applied to the problem of choosing the best representation for a given time series by Hu et al. [42]. The authors propose a method to choose the best representation (and its parameters) among APCA, PLA and DFT. While there are similarities with our method (we also use the MDL principle to select the best model parameters for a given component), the authors put the stress on discovering the intrinsic cardinality of the data, other than its constituent multi-scale components. MDL has also been adopted to detect changes in the distribution of a data stream by van Leeuwen et al. [94].

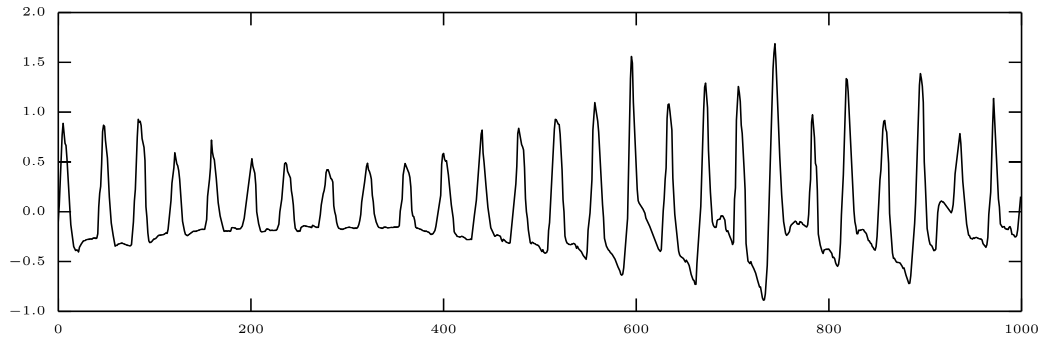


Figure 4.10: Detail of the first component.

In [44], the authors propose a method to select non-linear models from data, possibly generated by chaotic systems, having a good robustness to noise. Again, this work does not take into account the eventual multi-scale nature of the data.

4.6 Conclusions and Future Work

We introduced a novel methodology to discover the fundamental scale components in a time series in an unsupervised manner. The methodology is based on building candidate scale decompositions, defined over the scale-space image [103] of the original time series, with an MDL-based selection procedure aimed at choosing the optimal one.

A useful side product of the presented technique, due to the adoption of MDL, is that each discovered component is represented independently according to its inherent complexity and often results in a cheaper model (in terms of MDL score) in relation to the original raw time series. These cheaper per-component representations may better serve tasks like classification, regression or association analysis for time series produced by inherently multi-scale physical and artificial systems.

We have shown that our approach successfully identifies the relevant scale components in both artificial and real-world time series, giving meaningful insights about the data in the latter case. Future work will experiment with diverse representation schemes and hybrid approaches (such as using combinations of segmentation, discretization and Fourier-based encodings). Moreover, another interesting research question is how to substitute the presently employed exhaustive search of the optimal decomposition with a computationally cheaper heuristic approach, which is necessary in the case of large time series data.