Mining sensor data from complex systems

Vespier, U.

Chapter 3

Preliminaries and Background

As discussed in the previous chapters, the primary focus of this thesis is the unsupervised analysis of sensor data collected from complex multi-scale systems. In this chapter, we review the main background concepts and tools that lay the foundations and prepare the discussion for the material in the remaining chapters.

We will start by introducing the concept of *convolution* and *signal filtering* [85]. As we deal with noisy measurements from real-world sensor networks, we will show how convolution and filtering can be used to reduce the effect of noise and support several other time series manipulation tasks. Building on the concept of convolution, we will then present a fundamental tool employed in the rest of the thesis that supports the analysis of a time series at multiple temporal scales: the *scale-space image* [103]. Finally, as our main focus is on the unsupervised modeling of phenomena in time series data, we will introduce the *Minimum Description Length principle* as our model selection framework of choice [34], motivate its adoption and discuss a simple application of it to noise removal.

3.1 Preliminaries

We start by giving some fundamental definitions used in this chapter and throughout the thesis. We deal with finite sequences of numerical measurements, collected by observing some property of a system with a sensor, and represented in the form of time series as defined below.

Definition (Time Series). A **time series** of length n is a finite sequence of values $\mathbf{x} = x[1], \ldots, x[n]$ of finite precision[1].

Throughout the thesis, we assume that the measurements are collected at uniformly spaced time points and according to a fixed sampling rate.

In many contexts, it is of particular interest to refer only to certain contiguous portions of a time series or subsequences, as defined below:

Definition (Subsequence). A subsequence $\mathbf{x}[a : b]$ of a time series $\mathbf{x}$ is defined as follows:

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

3.2 Convolution and Filtering

Convolution is arguably one of the most important techniques in signal processing, for both the one-dimensional (time series) and two-dimensional case (images), and the fundamental operation in linear filtering. From a mathematical standpoint, convolution combines two functions to produce a third one, as defined below.

Definition (Convolution). Given two functions f and g, their **convolution $\mathbf{f} * \mathbf{g}$** is defined as the integral of their product after one of the functions is reversed and shifted:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t - \tau)\, d\tau$$

[1] We assume 32-bit floating point values throughout the rest of this dissertation.

When referring to the process of filtering, the function f is said to be the signal to be filtered while the function g is called *convolution filter kernel*. Kernel functions directly define the properties of the filter. Specific kernel functions can be defined for amplification and attenuation, shifting or echoing of a signal. Other classes of kernel functions, as we will see shortly, can be used for low-pass and high-pass frequency filtering.

3.2.1 Convolution and LTI Systems

Although, in the context of this thesis, convolution is mainly used as a filtering operation, it is worth mentioning its deep connection with the theory of *linear time-invariant* (LTI) systems [85]. A system, defined by an input signal $x(t)$ and output signal $y(t)$, is said to be linear time-invariant if it satisfies the linearity and time-invariance properties. Linearity refers to the fact that a linear mapping exists between the inputs and the output of the system. More formally, given two inputs $x_1(t)$ and $x_2(t)$, respectively producing outputs $y_1(t)$ and $y_2(t)$, the scaled and summed input $a_1x_1(t) + a_2x_2(t)$ will produce $a_1y_1(t) + a_2y_2(t)$, where a_1 and a_2 are scalars. The property holds for any arbitrary number of terms.

On the other hand, time-invariance means that the output of the system does not depend on the particular time a given input is applied. In detail, given an input $x(t)$ and an output $y(t)$, the delayed input $x(t-\delta)$ will produce the delayed output $y(t-\delta)$.

Without diving into the specifics of the theory, we note that the operation of convolution fully describes the output of any arbitrary LTI system with a known impulse response. In fact, given an input signal $x(t)$ and an impulse response $h(t)$, the output of the associated LTI system is given by the convolution $x(t) * h(t)$. LTI systems and convolution play an important role in several technical fields such as signal processing, electronics, seismology [2], spectroscopy and control theory.

3.2.2 Discrete Convolution

Although we defined the convolution operation in the continuous domain, in practical cases, however, we deal with finite signals in the discrete domain (time series). The definition of convolution in the discrete case is presented below.

Definition (Discrete Convolution). Given a time series $\mathbf{x}$ of length n and a convolution filter kernel $\mathbf{h}$ of length m, the result of the **discrete convolution** $\mathbf{x} * \mathbf{h}$ is the time series $\mathbf{y}$ of length n, defined as:

$$\mathbf{y}[t] = \sum_{j=-m/2+1}^{m/2} \mathbf{x}[t-j]\,\mathbf{h}[j]$$

Note that since $\mathbf{x}$ is finite, $\mathbf{x}[t-j]$ may be undefined. To account for these boundary effects, $\mathbf{x}$ is padded with $m/2$ zeros before and after its defined range.

If not specified otherwise, from now on we will only refer to the discrete case of convolution.

3.2.3 Noise Filtering via Gaussian Smoothing

A common use of the convolution operator is smoothing a signal to remove noise or finer details. Smoothing can be obtained by employing several types of kernels like mean, median and Gaussian. In [12], the authors propose a methodology to mine interesting correlations in multivariate time series data based on generating features obtained by convoluting the input data with different kernels in order to enhance or reduce certain characteristic.

Gaussian smoothing, however, is a common noise filtering technique, as it cuts high frequencies, leaving untouched the low ones. It also has other useful properties as we will see in the next section.

Gaussian smoothing is based on the Gaussian kernel [75], as defined be-

low:

$$\mathbf{G}_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

which, in the scope of this thesis, has a mean of 0, standard deviation σ and area under the curve equal to 1.

We can now define Gaussian smoothing as a particular convolution operation employing a Gaussian kernel.

Definition (Gaussian Smoothing). Given a time series $\mathbf{x}$ of length n and a Gaussian kernel $\mathbf{G}_\sigma$ discretized into m values, the result of the **Gaussian smoothing** is the time series $\mathbf{y}$ of length n, defined as:

$$\mathbf{x}[i] * \mathbf{G}_\sigma[i]$$

Note that to capture almost all non-zero values, we define $m = \lfloor 3\sigma \rfloor$.

The convolution acts as a *smoothing filter* which smooths each value $\mathbf{x}[t]$ based on its surrounding values. The amount of removed detail is directly proportional to the standard deviation σ (and thus m), from now on referred to as the *scale parameter*. In the limit, when $\sigma \to \infty$, the result of the Gaussian convolution converges to the mean of the signal $\mathbf{x}$ over the entire period involved.

To better picture the effect of Gaussian smoothing, consider the example in Figure 3.1. The top plot shows a signal collected from a strain sensor and the middle plot shows the same signal after being convoluted with a Gaussian kernel having $\sigma = 2$. The bottom plot highlights the part of the signal that has been removed by the filtering operation. Note how this residual signal does not just contain noise but it is still somewhat influenced by phenomena actually present in the signal, i.e. the peaks induced by passing vehicles. Designing the right noise removal filter ultimately boils down to choosing the right σ parameter for the given data.

In general, however, the choice of σ is strictly related to the task at hand and, consequently, to what is actually considered 'noise' in the given data. Consider, for example, a time series representing several months of strain measurements from a bridge deck at a high sampling rate (say 100 Hz or

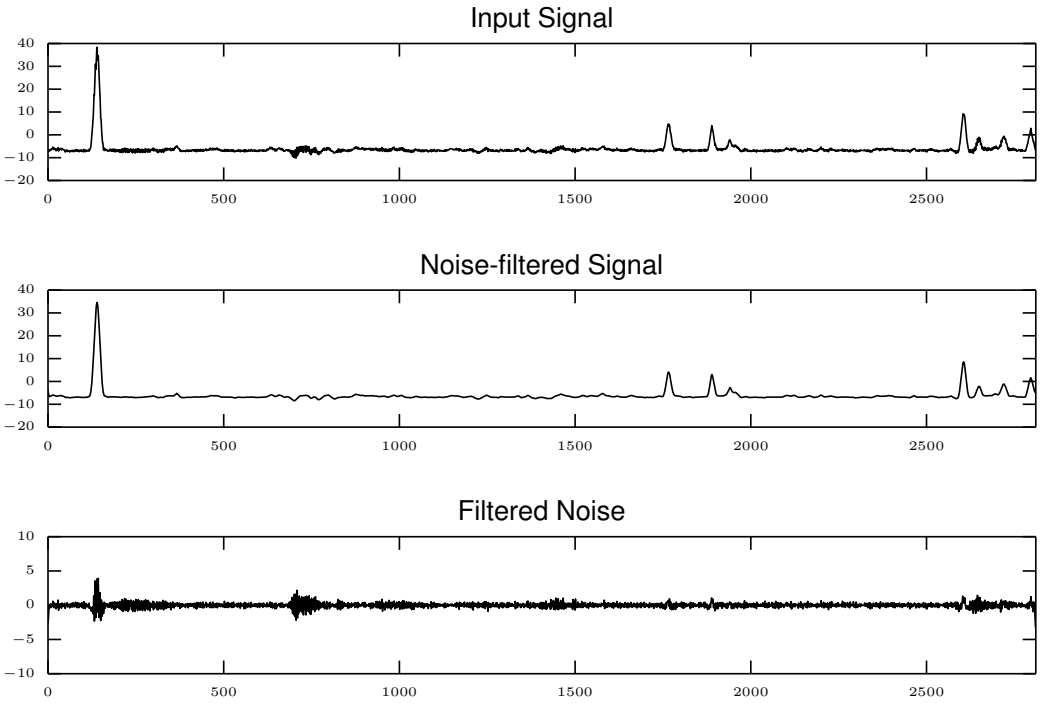

Figure 3.1: An example application of a Gaussian-based noise removal filter.

more). A typical bridge is affected by several phenomena at multiple temporal scales, ranging from events with a duration in the order of seconds such as passing cars and trucks, slightly longer ones such as congestion and weather conditions, to long-term ones like seasonal effects. The collected time series will represent all of these phenomena as a mixture. If we are interested in all the phenomena from the shortest to the longest, our concept of noise will coincide with anything lying below the temporal scale of the traffic events. On the other hand, if we are just interested in studying the effect of seasonal changes on the signal, we can ignore all the phenomena having shorter temporal scales and safely discard them as noise. This simple example illustrates how the concept of noise and the concept of scale of analysis are actually strictly interrelated and that a precise interpretation of noise can only be given by first looking at the task at hand.

In an unsupervised setting, it is not always clear what temporal scales we should look at and thus it is not always possible to determine in advance the

right value for σ. It follows from the definitions above that, by varying the parameter σ, Gaussian smoothing can be used to remove a fixed amount of detail from a signal. In other words, Gaussian smoothing can be interpreted as an operator that retains the information present above a certain temporal scale, where the scale is directly proportional to σ. This interpretation of Gaussian smoothing as scale parametrization is the core concept behind scale-space theory, a mathematical construction that we will use in the rest of the thesis and that we introduce in the next section.

3.3 Scale-Space Image

The *scale-space image* [103] is a scale parametrization technique for one-dimensional signals[2] based on convolution. Given a signal $\mathbf{x}$, the family of σ-smoothed signals $\Phi_{\mathbf{x}}$ over scale parameter σ is defined as follows:

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

where $\mathbf{g}_{\sigma}$ is a Gaussian kernel having standard deviation σ, and $\Phi_{\mathbf{x}}(0) = \mathbf{x}$.

The signals in $\Phi_{\mathbf{x}}$ define a surface in the time-scale plane (t, σ) known in the literature as the *scale-space image* [62, 103]. This visualization gives a complete description of the scale properties of a signal in terms of Gaussian smoothing. Moreover, it has other properties useful for segmentation, as we will see in Section 4.3.1.

For practical purposes, the scale-space image is quantized across the scale dimension by computing the convolutions only for a finite number of scale parameters. More formally, for a given signal $\mathbf{x}$, we fix a set of scale parameters

$$S = \{2^i \mid 0 \leq i \leq \sigma_{max}\ \wedge i \in \mathbb{N}\}$$

and we compute $\Phi_{\mathbf{x}}(\sigma)$ only for $\sigma \in S$ where σ_{max} is such that $\Phi_{\mathbf{x}}(\sigma)$ is approximately equal to the mean signal of $\mathbf{x}$.

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

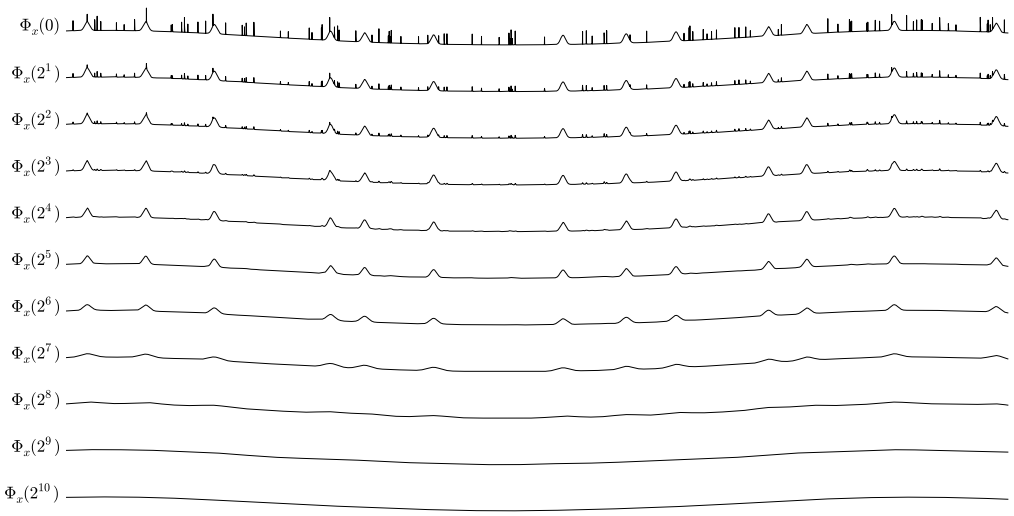

Figure 3.2: Scale-space image of an artificially generated signal totalling 259 200 points.

As an example, Figure 3.2 shows the scale-space image of an artificially generated signal. The top plot represents the original signal, constructed by three components at different temporal scales: a slowly changing and slightly curved baseline, medium-term events (bumps) and short-term events (peaks). It is easy to visually verify that, by increasing the scale parameter, a larger amount of detail is removed. In particular, the peaks are smoothed out at scales greater than $\sigma = 2^4$, and the bumps are smoothed out at scales greater than $\sigma = 2^8$, after which only the baseline remains.

3.3.1 Relation to the Zero-Crossings of Derivatives

The scale-space image has a number of interesting properties. An important one, that we will exploit throughout the thesis, is a property of Gaussian convolution that relates its zero-crossings, and those of its derivatives, to the scale parameter σ. In fact, as σ increases, the number of zero-crossings of the convoluted signal and of all its n-th derivatives can only remain constant or decrease [103]. Figure 3.3 demonstrates this concept in practice. The figure shows the relationship between the scale parameter σ and the number of zero-

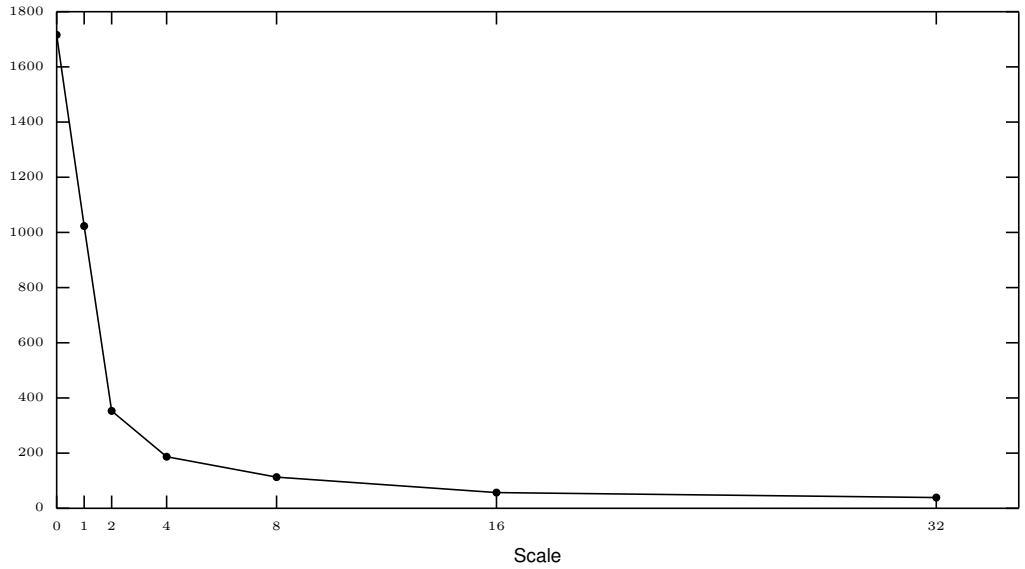

Figure 3.3: Relationship between the scale parameter and the number of zero-crossings of the first derivative of the signal shown in Figure 3.1.

crossing of the first derivative of the signal discussed in Figure 3.1.

3.4 Minimum Description Length

A recurring idea in this thesis is that learning and finding regularities in data can be seen as a form of *compression*. Compression is the act of representing some given data in the most compact way possible, such that its compressed representation has a lower number of bits than the original one [80]. The idea that the better we can compress a given data set, the more we can learn about it is a powerful one and it is formalized by the *Minimum Description Length principle*.

The Minimum Description Length [34] is an information-theoretic model selection framework that selects the best model according to its ability to *compress* the given data. In our context, the two-part MDL principle states that the best model M to describe the signal $\mathbf{x}$ is the one that minimizes the sum $L(M) + L(\mathbf{x} \mid M)$, where

- $L(M)$ is the length, in bits, of the description of the model,
- $L(\mathbf{x} \mid M)$ is the length, in bits, of the description of the signal when encoded with the help of the model M.

Given some data, MDL looks for a trade-off between the accuracy of a model and its complexity. Conceptually, MDL is a practical instantiation of the Occam's razor principle which states that, among several different hypotheses, the simplest is often also the best [91]. Moreover, MDL naturally protects against over-fitting as the principle takes into account the notion of model complexity and it discards models that are too complicated.

As we are dealing with unsupervised learning from time series data and we are interested in models that are as general as possible, the properties of the MDL framework makes it an ideal choice when designing model selection procedures.

In fact, prior work [42, 78, 88, 19, 10, 52] has already validated the effectiveness of the MDL approach when dealing with time series data and, throughout this thesis, we will further investigate its applicability to the analysis of sensor data.

3.4.1 Time Series Discretization

In order to use the MDL principle, we need to work with a quantized input signal and scale-space image. Because of this, we assume that the values v of the input signal $\mathbf{x}$ (and of the scale-space components $\Phi_{\mathbf{x}}(\sigma)$ for each considered σ) have been quantized to a finite number of symbols by employing the function defined below:

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

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 an maximum value in $\mathbf{x}$. Throughout the rest of the thesis, we assume $l = 256$.

One question that might arise is if such a quantization removes meaningful information from the time series. In [42], the authors show that the effect of quantization is rather modest on several time series from various domains.

3.4.2 MDL Noise Filtering

In 3.2.3, we have shown how Gaussian convolution can be used to remove high-frequency components from a given signal and, thus, serve as a noise removal filter. We stressed, however, that the choice of the parameter σ, is a critical one and strictly depends on the characteristics of the data at hand. In this section, we use the Minimum Description Length principle to select the optimal σ given a time series, where by *optimal* we mean the one that retains the most characteristic information in the data.

Assume we are given a time series $\mathbf{x}$ of length n and, as we are using MDL, its values have been discretized to a fixed cardinality (in this example, 256 possible values) using the quantization function Q introduced in above.

In order to frame the problem from an MDL perspective, we first have to define what the possible models for $\mathbf{x}$ are. We consider the components of the scale-space $\Phi_{\mathbf{x}}$, quantized to different cardinalities, as models. In other words, given a scale parameter σ and a cardinality c, we define a model for $\mathbf{x}$ as $M_{\sigma,c} = Q(\Phi_{\mathbf{x}}(\sigma), c)$, where Q is a quantization function.

The MDL principle states that the best model to compress $\mathbf{x}$ minimizes the sum $L(M) + L(\mathbf{x} \mid M)$. We define the description length of a model in terms of its entropy as $nH(\mathbf{x})$, where H is the entropy function. The description length of x when encoded with the help of a model refers to the complexity of the residual $nH(\mathbf{x}-M)$, that is the information discarded by the model.

Figure 3.4 shows the results of the approach that we just discussed in a practical scenario. The top plot depicts the input time series: a detail of a peak in a bridge's strain sensor signal caused by a passing vehicle. The plot in the middle shows the time series after being Gaussian-smoothed with the optimal σ, selected as discussed. Note how both the overall shape of the

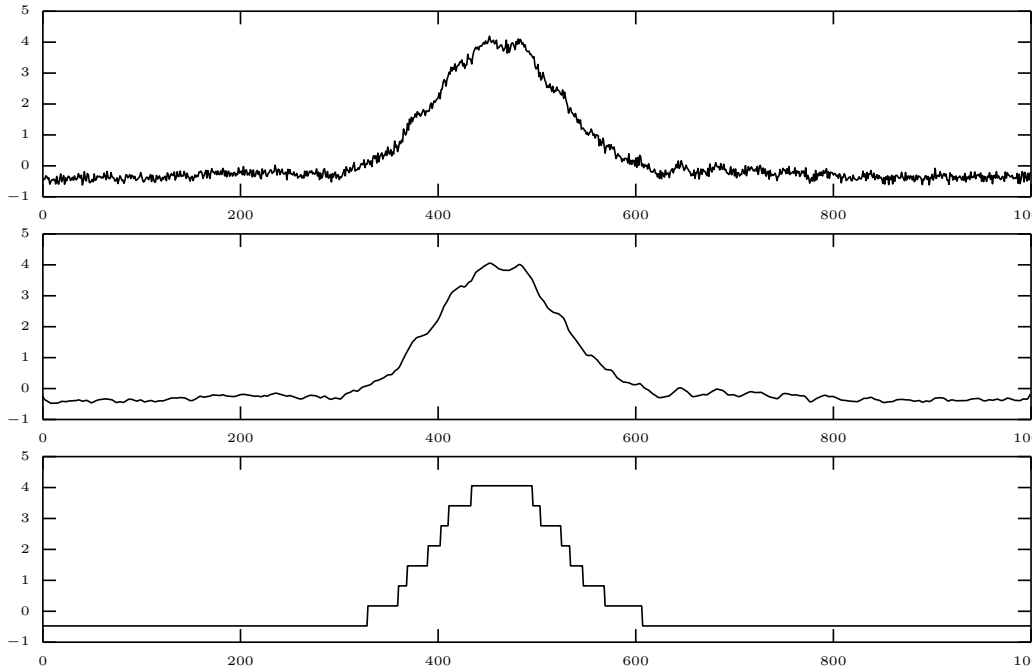

Figure 3.4: Example of MDL-based noise filtering.

signal and the subtle vibrations iduced by the passing truck are retained. The bottom plot, finally, shows the optimal model (quantized) according to MDL. For this particular example, we considered $\sigma \in \{2, 2^2, 2^3, 2^4, 2^5, 2^6\}$ and the cardinality $c \in \{4, 8, 16, 32, 64, 128, 256\}$. The optimal model is given by $\sigma = 2^3$ and $c = 8$.

A similar approach to noise removal has been taken by Miao et al. [71] as a preprocessing step in the context of pattern detection in time series data.