Structural health monitoring meets data mining

Miao, S.

Chapter 6

Predefined Pattern Detection

6.1 Introduction

This chapter focuses on the problem of detecting instances of predefined patterns in large time series data [28, 68]. While most pattern detection algorithms in time series deal with discovering previously unknown, frequently recurring regularities in the streaming data, here we assume that one or more example sequences (the *templates*) are provided by a domain expert, and instances of these need to be identified in the actual data. During this detection, one needs to allow for a certain degree of difference between the template and the instances, for example because the instance is somewhat longer or shorter in duration, the magnitude of the signal is different, or parts of the signal are either stretched or compressed in time (so-called warps).

Li Wei et al. [68] mention a number of use-cases that motivate the predefined pattern detection problem. For example, in ECG monitoring, a cardiologist may observe some interesting pattern that he or she wants to annotate, and flag any future occurrences, to be investigated by the cardiologist or fellow experts. Alternatively, in insect pest control, one would like to observe specific cases of harmful insects, as identified by specific patterns of audio signal (wing beats). In our application to infrastructure monitoring, the predefined pattern detection problem is relevant for specifying and detecting known disturbances in the data, that can

then be removed from the signal, or accounted for in subsequent modelling steps. For example, when monitoring the structural health of a bridge, the measured signal is dominated by recurring and understandable peaks due to vehicles crossing the bridge and traffic jams. One can imagine an expert providing a template for each of theses phenomena, after which all instances should be identified, regardless of the speed and weight of the vehicles (influencing the width and height of the hump in the signal), or the duration of the traffic jam.

When matching a predefined phenomenon (a template [31, 33, 69]) with the time series under investigation, it is not always required to involve every individual measurement in the selected interval and in the template. In fact, when a certain level of fuzzy matching is required, it makes sense to somehow simplify the signal, or extract some key features that are characteristic for the sequence in question. This condensed representation can then be used to compare the time series with the template, both effectively (the matching is only based on the characteristic aspects) and efficiently (no computation is wasted on insignificant details). Specifically when large time series with high sampling rates are concerned, and the matching is nontrivial due to warps, efficient representation methods can be helpful. A considerable number of such methods have been proposed in the past, including Symbolic Aggregate approXimation (SAX) [70], bit-level approximation [71], and Piecewise Aggregate Approximation (PAA) [72][1]. In this chapter specifically, we focus on the representation of time series by means of *landmarks* [73] (also referred to as key-points [74], break-points [75] and change-points [76]), which can be thought of as those points in the time series that are obviously remarkable (peaks, valleys, inflection points, ...). Rather than matching every detail of the data and the template, only the landmarks will be matched, and subsequent landmarks will be checked for their relationship to one another.

We match the given template to the actual data in three steps. The first step involves transforming the time series into a landmark sequence, which preserves all the prominent features. The second step is landmark subsequence selection, which is based on the constraints over the landmarks occurring in the template. The third step is landmark model construction, which introduces *trust feature*

[1] A comprehensive list of representation methods for time series is given in Section 6.7.

and *trust region* to model the time series segments corresponding to the selected landmark subsequence. Unlike most of the representation and similarity methods, which are designed mainly for full sequence matching [28], our proposed approach is capable of processing both full sequence and subsequence matching of various length, while being less sensitive to noise, and being able to handle deformations in both magnitude and temporal dimensions.

One of the challenges when extracting landmarks from actual data is the noise and high-frequency vibrations that are included. An obvious step to get rid of such distractions and to produce a set of meaningful landmarks is to convolve the signal with a smoothing kernel. The question now becomes what level of smoothing is appropriate for the template in question. Too much smoothing may cause one to miss characteristic landmarks in the data, and too little smoothing will cause an abundance of landmarks at every little disturbance in the data. We propose an MDL-based solution to this challenge, that picks the correct smoothing level. Minimum Description Length (MDL) [77, 78, 79] is an information-theoretic model selection framework that selects the best model according to its ability to *compress* the given data.

The contributions of this chapter are summarised as follows:

- It provides a general definition of a template for time series, which can be represented by a landmark vector.
- It proposes the use of landmarks: a triple involving temporal, magnitude and type information.
- It takes the relationship between landmarks within a landmark sequence as constraints for landmark subset selection.
- It introduces the concept of a trust region from the image processing domain [80] to time series to build a reliable template model, which could help to detect the precise location of landmarks.
- It employs MDL [77, 78, 79] for selection of the right smoothing level for landmark extraction.

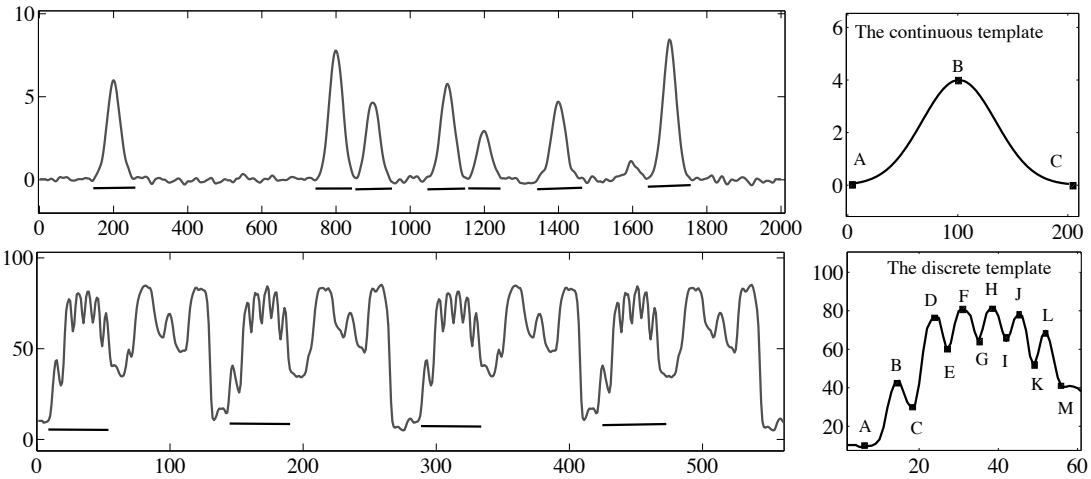

Figure 6.1: The continuous template and the discrete template - The signal in the top left picture is the time series. The curve in the top right picture is a continuous template (more specifically a Gaussian), which is marked with landmarks A, B, and C. The bottom left picture represent bird songs. The curve in the bottom right picture is a discrete template, corresponding to one of the selected subsequences, marked with landmarks A, B, . . . , M.

The rest of this chapter is organised as follows. Section 6.2 gives the definitions of template and landmark, and specifies the task of predefined pattern detection. Section 6.3 introduces the concept of landmark constraints. Section 6.4 introduces landmark model construction based on continuous and discrete templates. Section 6.5 uses MDL to select the optimal smoothing scale. Section 6.6 evaluates the proposed method by applying it to artificial and real datasets. Section 6.7 gives a literature review of related work, followed by a conclusion in Section 6.8.

6.2 Preliminaries

In order to specify the exact predefined temporal pattern one hopes to find in the time series, we define a *template* in one of two ways. In the first, *continuous*, way, we assume that a temporal pattern is defined by a function that specifies the shape of the pattern with infinite precision. In the second way, the *discrete*

one, a temporal pattern is defined by a sequence of values, for example obtained by averaging a number of selected subsequences of interest.

Definition 5 *A* continuous template $\boldsymbol{H}_c$ *is a function that can serve as a model for subsequences of a time series*

$$\boldsymbol{H}_c(x) = f_A(x)$$

where x is an integer, and f is a given function with coefficients A (for example $\{\mu, \sigma\}$ in the case of a Gaussian curve).

We demonstrate this type of template using an artificial dataset, shown as the curve in the top left picture of Fig. 6.1. The shape of the recurring subsequence can be modeled faithfully with a Gaussian function, an instance of which is shown as the curve in the top right picture of Fig. 6.1. The matching subsequences are identified by the bars below the graph.

Definition 6 *A* discrete template $\boldsymbol{H}_d$ *is a time series that can serve as a model*

$$\boldsymbol{H}_d = (h_1, h_2, \ldots, h_k), \quad h_i \in \mathbb{R}$$

where k specifies the size of the subsequence. The recurring subsequences in the bottom left picture of Fig. 6.1 (depicting bird songs [81]) are more complicated than the patterns in the top left picture of Fig. 6.1. We could choose one subsequence from the smoothed time series as a template, such that the discrete template becomes the one shown on the bottom right.

6.2.1 Landmark Extraction

Although we expect the user to specify the predefined pattern in terms of a template (be it discrete or continuous), we will not be matching the template directly to the given time series. Rather, we intend to extract important *landmarks* [73] from both the template and the time series, and use these to match more efficiently and effectively. A landmark is defined as follows:

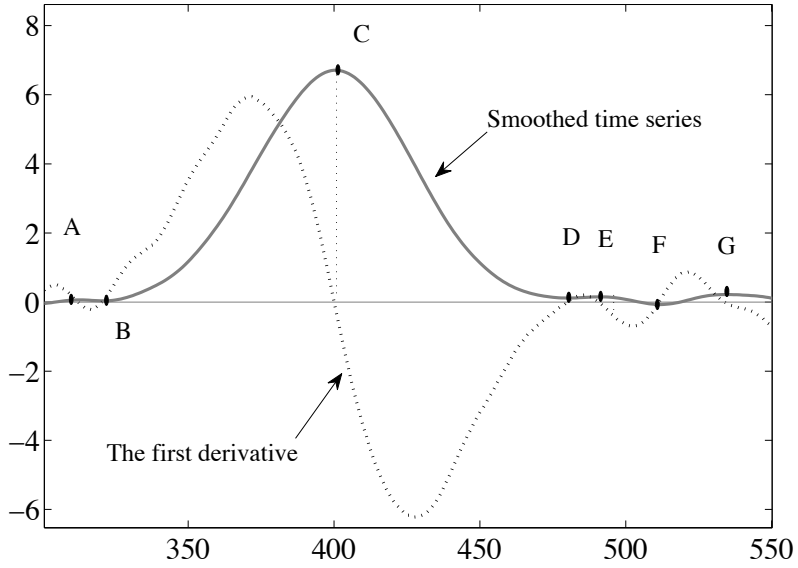

Figure 6.2: Landmark extraction - The dark curve is part of the smoothed time series. The dotted line is the scaled first derivative of the smoothed time series. The points marked with letters are landmarks.

Definition 7 *Given a time series* $\boldsymbol{T} = (t_1, t_2, \ldots, t_n)$*, a* landmark *is a remarkable point in* $\boldsymbol{T}$*, specified by a triple* l*:*

$$l = (id, m, type), \quad id \in \mathbb{N}, m \in \mathbb{R}$$

where id is the index of the landmark in the time series $\mathbf{T}$: the landmark is located at t_{id}. m is the magnitude of the landmark, *type* is the peak type indicator, which can be local extreme, inflection point or some other notable characteristic of the time series at this point.

In later sections, we will be introducing *landmark extraction* methods, which produce a sequence $\mathbf{L}$ of landmarks from a given time series. Such a method, generally identified as a function E, can be applied to obtain a sequence of remarkable points from a given time series, but equally, it can be used to produce such points from a (discrete) template, as that is essentially a time series also.

Landmark extraction methods are typically application dependent. In general, local extrema of a smoothed time series are good landmark candidates. They

are found by considering the zero-crossings of the first derivative of the series. These zero-crossings (roots) correspond to the extrema in the time series, which we assume to be of interest. The *inflection points* derived from the extrema in the first derivative time series can also be considered landmark candidates. Such landmarks can be found by looking at the zero-crossings of the second derivative. A landmark sequence preserves the main features of the time series, but significantly reduces its representation size. As shown in Fig. 6.2, the time series segment with a length of 250 can be compressed to a landmark sequence of only 7 elements. Note the importance of convolution with a smoothing kernel (e.g. a Gaussian) in order to get rid of the noise, which would produce an over-abundance of landmarks.

For each sequence of landmarks, there is a time series *segment* corresponding to it, which is defined as:

Definition 8 *Given a landmark sequence* $\boldsymbol{L} = (l_1, l_2, \ldots, l_k)$ *of length* k*, a* landmark segment $\boldsymbol{S}$ *of* $\boldsymbol{L}$ *is defined as a subsequence of time series* $\boldsymbol{T}$*:*

$$\boldsymbol{S} = (s_1, s_2, \ldots, s_m) = (t_{start}, \ldots, t_{end}),$$

where t_{start} and t_{end} are the data points indicated by indexs (id) of l_1 and 1_k.

6.2.2 Predefined Pattern Detection

With the definitions of templates and landmarks now established, we can proceed by formally specifying the main task that we are concerned within this chapter, as follows:

Definition 9 *The task of* Predefined Pattern Detection *takes as input a time series* $\boldsymbol{T}$*, a (discrete or continuous) template* $\boldsymbol{H}$ *and a landmark extraction method* E_σ*, and produces a sequences of matches* $\boldsymbol{M} = (m_0, \ldots, m_k)$*, where each* m_i *is an index in* $\boldsymbol{T}$ *where a match is found between the template and the subsequence starting at* m_i*.*

Note the role of E_σ in this definition. As mentioned, we are not matching the template to the time series directly, but rather extracting landmarks from both

first, using E_σ. An important parameter in E is σ, which determines the level of smoothing applied to both the template and the time series. By smoothing, we prevent noise from playing a role in determining what constitutes a landmark. Of course, the level of noise (as opposed to the actual signal) depends on the application, so for the moment we assume this as simply a parameter of the task. In Section 6.5.1, we will describe how the MDL principle can be employed to decide on a proper choice of σ.

6.3 Landmark Constraints

In theory, for a given template landmark sequence of length n and a time series landmark sequence of length m, there are $m - n + 1$ candidate landmark subsequences. Compared with the subsequence candidates from the original time series, the number has already been reduced a lot. However, there are still many ways in which landmarks in the template can be matched with those in the time series. In this section, we introduce landmark constraints to break the landmark sequence into a number of meaningful landmark subsequences.

For a given template, the landmarks in its landmark sequence signify more than just several data points obtained with landmark extraction methods. There are two levels of constraint existing in the landmark sequence of the template:

1. The first level is local constraints. As defined in Section 6.2, each landmark has three properties: index, magnitude and type indicator. We can set constraints based on each landmark property.

2. The second level is global constraints. The number of landmarks within the template landmark sequence determines the length of interesting landmark subsequences: the relationship between properties of different landmarks could form an even richer constraint-set.

For example, based on the template landmark sequence $\mathbf{L}_H = \{\text{A, B, C}\}$ in the top right picture of Fig. 6.1, the constraint-set can be set as follows:

- The length of landmark subsequences should be 3.

- The first landmark A and the third landmark C should be valley points.
- The second landmark B should be a peak point.
- The magnitude of landmark B should be higher than the magnitude of A and C.
- The relative magnitude $B_m - A_m$ should be higher than 0.8.
- The peak duration $C_{id} - A_{id}$ should be longer than 50.

The thresholds of the above constraints should be general enough to include all the potentially interesting patterns, and at the same time, they should be strict enough to filter out false patterns.

6.4 Fitting Templates to the Data

In Section 6.2, a template is defined as either a continuous or discrete template. According to the type of the template, we build two types of landmark models: the continuous landmark model and the discrete landmark model.

6.4.1 Continuous Landmark Model

For a given continuous template $\mathbf{H}_c$, and a landmark segment $\mathbf{S}$, the continuous landmark model $\mathbf{LM}_c$ of the landmark segment $\mathbf{S}$ is an instance of the continuous template $\mathbf{H}_c$. The coefficient set A of the continuous template $\mathbf{H}_c$ can be obtained by extracting features from the landmark subsequence of $\mathbf{S}$.

For example, a Gaussian function is chosen as the continuous template $\mathbf{H}_c = ae^{-(x-\mu)^2/(2\sigma^2)}$ to model the bell-like landmark segments in the middle and right pictures of Fig. 6.3. An instance of the template is shown as the curve in the left picture of Fig. 6.3, which is marked with landmarks A, B and C. The template is characterized with three features: σ, the peak location μ and the magnitude a. The peak location is derived from landmark B. σ can be derived from the temporal difference between any pair of landmarks, so there are 3 combinations

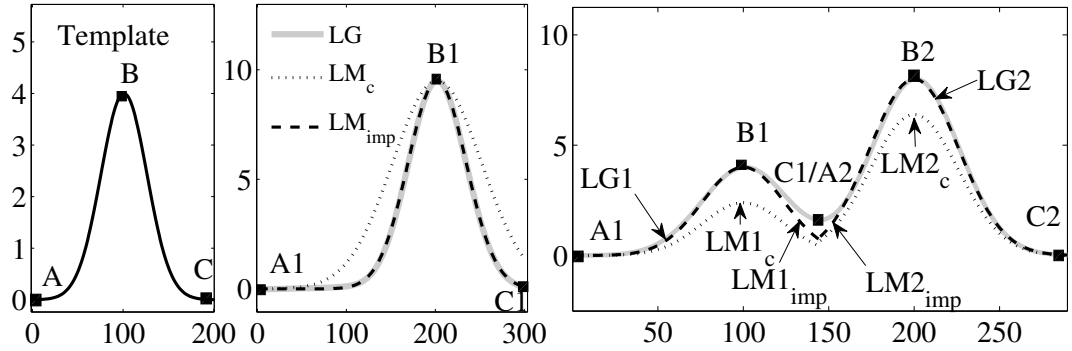

Figure 6.3: The continuous landmark model - The curve in the left picture an instance of the given continuous template, whose landmarks are A, B and C; the dotted curves in the middle and right pictures are false continuous landmark models; the dashed curves in the middle and right pictures are the improved continuous landmark models based on trust features.

for the template landmark sequence. The magnitude a can be obtained from the magnitude difference between the landmarks B and A, or that between B and C.

Incorrect feature choices may lead to false landmark models: the dotted curve LM_c in the middle picture of Fig. 6.3 is a false landmark model caused by incorrect choice of σ (the temporal difference between landmarks C1 and A1 divided by 2); the dotted curves in the right picture of Fig. 6.3 are false landmark models caused by incorrect choice of magnitudes (the magnitude difference between landmarks B1 and C1 for $\mathrm{LM1}_c$, and that between landmarks B2 and A2 for $\mathrm{LM2}_c$).

Trust feature To overcome the limitations existing in the continuous landmark models mentioned above, we introduce the notion of trust feature. A feature is considered to be a trust feature if it reflects the true characteristics of a given landmark segment. Shown as the dashed curve in the middle picture of Fig. 6.3, the improved landmark model LM_{imp} is obtained by computing σ from the temporal difference between landmarks C1 and B1, which can be further improved by selecting more reliable landmarks, such as inflection points. Similarly, we improve the landmark models in the right picture of Fig. 6.3 by employing the magnitude

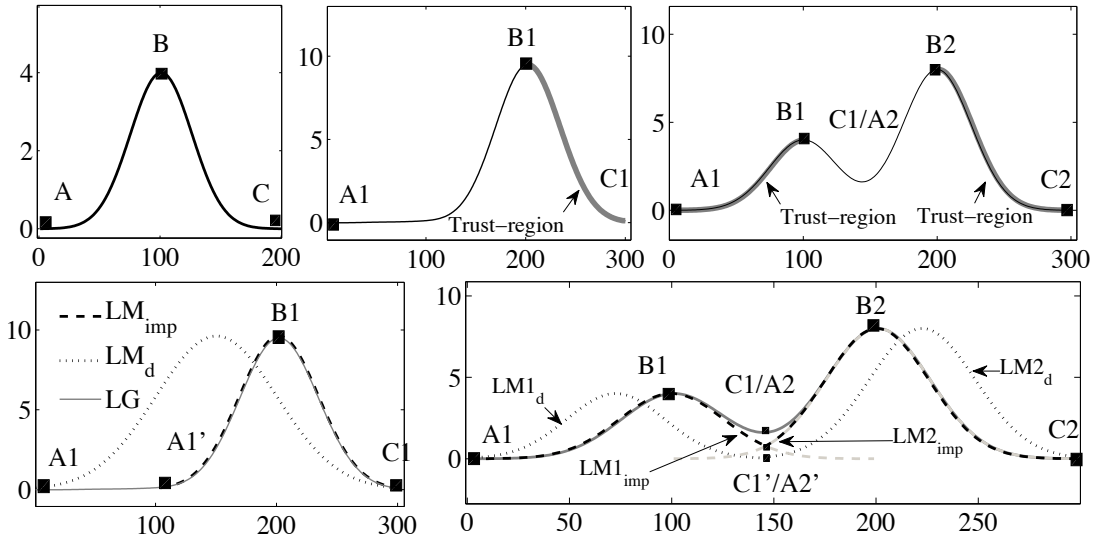

Figure 6.4: The discrete landmark model - The curve in the left picture is the given discrete template, which is marked with landmarks A, B and C; the segment between the landmark B1 and C1 in the top middle picture is chosen as the trust region; in the top right picture, the segment between the landmark A1 and B1 is the trust region of the first landmark segment, and that between the landmark B2 and C2 is the trust region of the second landmark segment; the dotted curves in the bottom left and right pictures are discrete landmark models simply obtained with transformations of the given template; the dashed curves in the bottom left and right pictures are improved landmark models based on the trust regions in the top middle and right pictures.

difference between landmarks B1 and A1 for LM1_{imp}, and that between landmarks B2 and C2 for LM2_{imp}, shown as the dashed curves.

6.4.2 Discrete Landmark Model

For a given discrete template $\mathbf{H}_d$ of length n and a landmark segment $\mathbf{S}$ of length m, we model $\mathbf{S}$ through scaling $\mathbf{H}_d$ in both temporal and magnitude dimensions. The temporal scale operation *X-scale* results in a new sequence $\mathbf{LM}_X$, whose length is the same as that of the landmark segment.

$$\mathbf{LM}_X = \textit{X-scale}(\mathbf{H}_d, m, n), \quad m, n \in \mathbb{N}$$

$$X\text{-}ratio = m/n$$

If *X-ratio* > 1, the *X-scale* is an up-sampling operation, and if *X-ratio* < 1, the *X-scale* is a down-sampling operation, otherwise, the $\mathbf{LM}_X$ is the same as the template $\mathbf{H}$.

We continue to process the obtained model $\mathbf{LM}_X$ with the magnitude scale operation *Y-scale*, and obtain a landmark model $\mathbf{LM}_Y$, which can be taken as a primary approximation of the landmark segment $\mathbf{S}$.

$$\mathbf{LM}_Y = Y\text{-}scale(\mathbf{H}_d, \mathbf{LM}_X)$$

$$Y\text{-}ratio = (\max(\mathbf{S}) - \min(\mathbf{S}))/(\max(\mathbf{H}_d) - \min(\mathbf{H}_d))$$

where $\mathbf{LM}_Y$ is obtained by first scaling $\mathbf{LM}_X$ with a ratio *Y-ratio*, resulting in a temporary model $\mathbf{LM}_{XY}$, then shifting along magnitude dimension by $\min(\mathbf{S}) - \min(\mathbf{LM}_{XY})$.

Based on the transformations mentioned above, we can define the discrete landmark model as:

Definition 10 *Given a landmark segment* $\boldsymbol{S} = (s_1, s_2, \ldots, s_k)$ *of length* k*, and a discrete template* $\boldsymbol{H}_d$*, the* discrete landmark model $\boldsymbol{LM}_d$ *of* $\boldsymbol{S}$ *is a sequence derived from transformations of the discrete template* $\boldsymbol{H}_d$*:*

$$\boldsymbol{LM}_d = Transf(\boldsymbol{S}, \boldsymbol{H}_d)$$

where *Transf* are the transformations defined above (*X-scale* and *Y-scale*).

The dotted curve $\mathbf{LM}_d$ in the bottom left picture of Fig. 6.4 is a discrete landmark model obtained by simply transforming the given template shown as the curve in the top left picture of Fig. 6.4. The obtained landmark model indicates that the left boundary of the landmark segment is incorrect, which is caused by the false landmark A1. The gray curves in the bottom right picture of Fig. 6.4 are two overlapping landmark segments, whose boundaries are correctly caught, but their landmark models, shown as the dotted curves $\mathrm{LM1}_d$ and $\mathrm{LM2}_d$, are still incorrect. This is caused by the complex structure of the time series, which has not been covered by the given template. To overcome all these limitations, we introduce the notion of trust region.

6.4.2.1 Trust Region

By assuming part of the landmarks within a landmark subsequence are reliable, we can define the corresponding segment between these landmarks as the trust region. Trust region is a concept borrowed from the field of image processing. We introduce the concept into time series (or to be more precise, into the discrete landmark model), and define it as:

Definition 11 *Given a landmark segment* $\boldsymbol{S} = (s_1, s_2, \ldots, s_n)$ *of length* n*, and its landmark sequence* $\boldsymbol{L} = (l_1, l_2, \ldots, l_k)$ *of length* k*, and assuming the segments between landmarks* l_i *and* l_j *(*$1 \leq i < j \leq k$*) are influenced less by noise, then the* trust region *of* $\boldsymbol{S}$ *is defined as:*

$$\boldsymbol{S}_{trust} = (s_a, \ldots, s_b), \quad 1 \leq a < b \leq n$$

where a is the index of landmark l_i in the landmark segment $\mathbf{S}$, and b corresponds to the index of landmark l_j. The dashed curves in the bottom left and right pictures of Fig. 6.4 are discrete landmark models obtained with the trust regions given in the top middle and right pictures, which indicate that the trust regions help to achieve improved discrete landmark models and updated landmark segments. The detailed procedure is illustrated in Algorithm 1.

The inputs of the algorithm are: a template $\mathbf{H} = (h1, \ldots, hn)$ of length n, a landmark segment $\mathbf{S} = (s1, \ldots, sm)$ of length m, and their landmark sequences ($\mathbf{L}_H = (lh_1, \ldots, lh_k)$ and $\mathbf{L}_S = (ls_1, \ldots, ls_k)$, respectively, both of length k). The outputs of the algorithm are an improved landmark model and an updated landmark segment.

A candidate trust region $\mathbf{H}_{cand}$ (identified by *index* and *len*) of the template and the landmark segment $\mathbf{S}_{cand}$ can be obtained with the $Region(\mathbf{S}, \mathbf{H}, index, len)$ operation, in which *index* is the index of the first landmark in $\mathbf{L}_H$ and $\mathbf{L}_S$, and *len* is the length of the subsequence. Based on these two parameters, we need to obtain the indexes of the first and last data points of the candidate trust regions in $\mathbf{S}$ and $\mathbf{H}$. Assuming the corresponding indexes of $\mathbf{H}$ are a and b, and that of $\mathbf{S}$ are c and d, the trust region of the template can be represented as

Algorithm 1 The landmark model.

Require: a template $\mathbf{H}$, and its landmark sequence $\mathbf{L}_H$ of length k, a landmark segment $\mathbf{S}$, and its landmark sequence $\mathbf{L}_S$ of length k.

Ensure: an improved landmark model $\mathbf{LM}_{imp}$ and an updated landmark segment $\mathbf{S}_{up}$

```
S_cand = S, H_cand = H
[LM_imp, S_up] = Model(S, S_cand, H, H_cand)
sim_max = Similarity(LM_imp, S_up)
for len = 2 to k do
  for index = 1 to k − len + 1 do
    [H_cand, S_cand] = Region(S, H, index, len)
    [LM_temp, S_temp] = Model(S, S_cand, H, H_cand)
    sim = Similarity(LM_temp, S_temp)
    if sim > sim_max then
      sim_max = sim
      LM_imp = LM_temp
      S_up = S_temp
    end if
  end for
end for
```

$\mathbf{H}_{cand} = (h_a, \ldots, h_b)$, and that of the landmark segment can be represented as $\mathbf{S}_{cand} = (s_c, \ldots, s_d)$, respectively.

Based on the candidate trust regions $\mathbf{H}_{cand}$ and $\mathbf{S}_{cand}$, the associated landmark model $\mathbf{LM}_{temp}$ and the updated landmark segment $\mathbf{S}_{temp}$ can be obtained with the $Model(\mathbf{S}, \mathbf{S}_{cand}, \mathbf{H}, \mathbf{H}_{cand})$ operation. The procedure of this operation is illustrated as follows:

- Step 1: based on the candidate trust regions $\mathbf{S}_{cand}$ and $\mathbf{H}_{cand}$, we can obtain a new sequence $\mathbf{LM}_X$ with the *X-scale* operation:

$$\mathbf{LM}_X = X\text{-}scale(\mathbf{H}, d - c + 1, b - a + 1)$$

- Step 2: following the same *Y-scale* operation procedure as mentioned above,

we obtain a landmark model $\mathbf{LM}_Y$:

$$\mathbf{LM}_Y = \textit{Y-scale}(\mathbf{H}, \mathbf{LM}_X)$$

- Step 3: pruning the obtained landmark model $\mathbf{LM}_Y$ or fixing the landmark segment $\mathbf{S}$, we finally achieve a landmark model $\mathbf{LM}_{temp}$ and landmark segment $\mathbf{S}_{temp}$.

  This step takes the segment, derived from the region $\mathbf{S}_{cand}$, in the temporary landmark model $\mathbf{LM}_Y$ as reference. If $\mathbf{LM}_Y$ is longer than the landmark segment $\mathbf{S}$, we need to prune the temporary model, otherwise, fix the landmark segment.

Finally, the fit of each candidate landmark model is evaluated with the *Similarity*() operation, which can be any of the choices as outlined in Chapter 2, most typically a similarity function based on the Euclidean Distance.

6.5 Determining the Smoothing Level

As noted, the Predefined Pattern Detection task requires the specification of a smoothing level, in order for the landmark detection to work effectively. When smoothing a time series for landmark extraction, there is clearly a trade-off at play. Smoothing too little will produce a time series that shows too many landmarks, and smoothing too much will remove too much of the interesting signal, such that important landmarks may be overlooked (see Fig. 6.5). In this section, we tackle the challenge of setting an appropriate value for the smoothing scale σ in E_σ.

Our solution to this challenge employs the Minimum Description Length principle [77]. The MDL principle states that, when choosing between several different candidate models of the data, the one that produces the cheapest encoding is the most desirable. In this context, the different candidate models are produced by the different choices of smoothing scale σ. In a nutshell, we consider a range of values for σ, applying landmark extraction E_σ to the smoothed time series. The

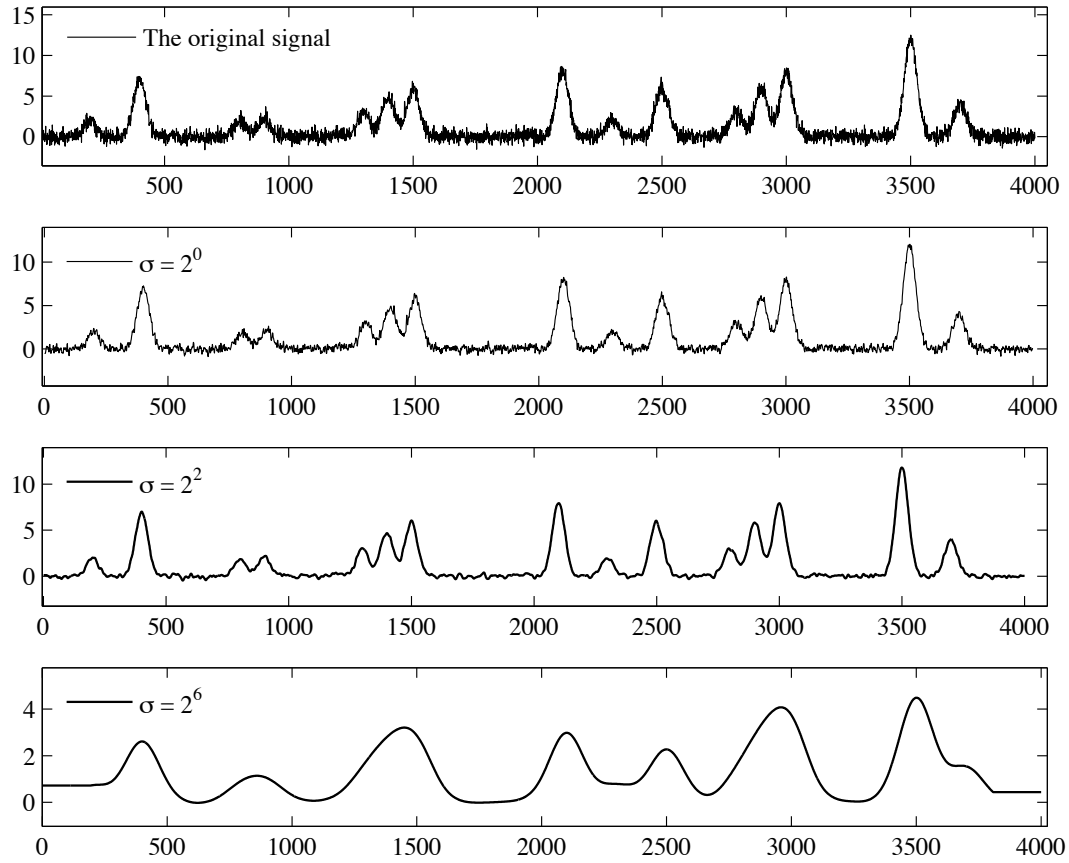

Figure 6.5: Various levels of smoothing - The first picture shows the original time series without any convolution; the second picture shows the smoothed time series with a scale of $\sigma = 2^0$, which still contains considerable noise; the third picture shows the smoothed time series with $\sigma = 2^2$, which suppresses the noise and preserves the interesting patterns; the fourth picture presents $\sigma = 2^6$, which suppresses both the noise and some of the interesting features.

idea of using MDL as a guiding principle to model various aspects of time series data has been introduced before in [79, 82], but not with the specific intent of selecting an appropriate choice of σ.

6.5.1 Minimum Description Length

We concentrate on the two-part version of the MDL principle, which states that the best landmark model LM to describe the time series $\mathbf{T}$ is the one that minimises the sum $L(LM) + L(\mathbf{T} \mid LM)$, where

- $L(LM)$ is the cost, in bits, of the landmark model derived from the given template.
- $L(\mathbf{T} \mid LM)$ is the length, in bits, of the description of the time series when encoded with the help of the landmark model LM, that is the residual information not represented by LM.

A good, detailed model that catches most features of the target dataset leads to a low cost of $L(\mathbf{T} \mid LM)$, but a good model also means a higher cost compared with a simple model. Therefore, a trade-off between model fit and its complexity is guaranteed by considering the size of the encoding. This property prevents the MDL method from overfitting.

Before we calculate $L(LM)$ and $L(\mathbf{T} \mid LM)$, we first need to discretise the landmark segment. We assume that the values t_i of the input time series $\mathbf{T}$ have been quantised to a finite number of symbols by employing the function defined below:

$$Q(t_i) = \left\lfloor (t_i - \min(\mathbf{T}))/(\max(\mathbf{T}) - \min(\mathbf{T})) \cdot N \right\rfloor - N/2$$

where N, assumed to be even, is the number of bins to use in the discretisation while $\min(\mathbf{T})$ and $\max(\mathbf{T})$ are respectively the minimum and maximum value in $\mathbf{T}$. Throughout the rest of the chapter, we assume $N = 256$, in correspondence with similar work on MDL in time series [79, 82]. One question that might arise is if such a quantisation removes meaningful information from the time series. In [82], the authors show that the effect of quantisation is rather modest on several time series from various domains.

6.5.1.1 Encoding of the Model

We will first discuss the encoding of the landmark model LM, which is derived from a given template. In the time series, the cost for encoding the landmark model is composed of two parts: the index and the model parameters. The location of any landmark segment candidate is less than the total length of the time series $\mathbf{T}$, so it can be encoded with $\log_2 n$ bits. Assuming there are m parameters for each model (for a continuous landmark model, m stands for the number of coefficients; for a discrete landmark model, m is the cost of transformations), and each parameter can be modelled with b bits, the total cost can be obtained by summing up these two parts:

$$L(LM) = k \cdot (\log_2 n + mb)$$

where k is the number of landmark segments in the time series that meet the landmark constraints.

6.5.1.2 Encoding the Data

The second part of MDL, $L(\mathbf{T} \mid LM)$, represents the residual information after subtracting the landmark model LM from the time series $\mathbf{T}$. To encode this part, we first need to introduce the notion of entropy.

Definition 12 *The* entropy *of a time series $\boldsymbol{T}$, discretised according to a set of values D, is defined as below*

$$Entropy(\boldsymbol{T}) = -\sum_{v \in D} P(t_i = v) \log_2 P(t_i = v)$$

where t_i stands for the i_{th} element in the time series $\mathbf{T}$, $P \log_2 P = 0$ in the case of $P = 0$, and $P(t_i = v)$ indicates the fraction of points in the time series which has value v.

Given the definition of entropy, we can define the description length of the second part of MDL as follows:

Definition 13 *Given a time series* $\boldsymbol{T}$ *of length* n*, the* description length *of* $L(\boldsymbol{T} \mid LM)$ *(in bits) is given by*

$$L(\boldsymbol{T} \mid LM) = n \cdot Entropy(\boldsymbol{T} \mid LM)$$

6.5.2 Smoothing Level Selection

For assessing a candidate smoothing level with parameter σ, we can simply take the corresponding smoothed landmark segments, which meet the landmark constraints, as landmark models, and obtain a residual by subtracting the obtained models from the original time series. We need two parameters (the indexes of the first and last data points of a landmark segment) to identify a landmark model. Assuming there are r interesting landmark segments under a smoothing scale σ, the landmark model cost $L(LM)$ becomes:

$$L(LM) = 2r \cdot log_2 n$$

The second MDL part $L(\mathbf{T} \mid LM)$, according to Def. 13 is represented as:

$$L(\mathbf{T} \mid LM) = n \cdot Entropy(\mathbf{T} - \sum_{i=1}^{i=r} \mathbf{Ts}_i)$$

where $\mathbf{Ts}_i$ is the i^{th} smoothed landmark segment.

For a given template and time series, we assume that the optimal degree of smoothing is the one that leads to the minimal total MDL cost.

6.6 Experiments

To show the effectiveness of the proposed method, we apply it to three different datasets, which ranges from artificial dataset to real-life datasets (traffic and ECG signals). We divide each dataset into two parts: training dataset and test dataset. The training dataset is used to detect the right smoothing scale and landmark constraints, then these obtained parameters will be applied on the test dataset.

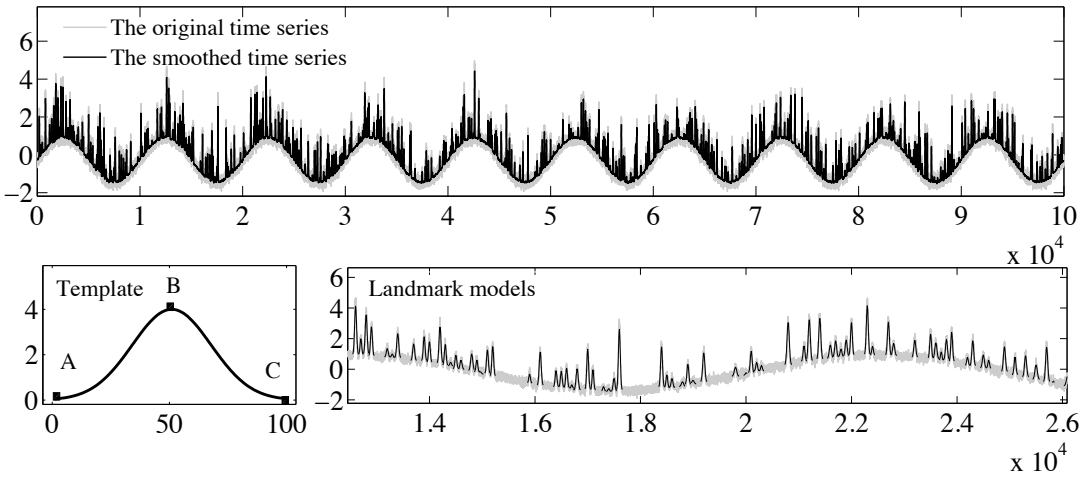

Figure 6.6: Predefined pattern detection from an artificial dataset - The grey curve in the top picture is an original artificial dataset composed of 100,000 data points; the black curve in the top picture is the time series smoothed with the smoothing scale $\sigma = 2^3$; the black curve in the bottom left picture is an instance of the selected template, which is a Gaussian function, marked with landmarks A, B, C; the black peaks in the bottom right picture are the detected patterns in a fragment of the whole dataset.

6.6.1 Artificial Dataset

We begin with testing the method on an artificial dataset, which is obtained by combining Gaussian peaks, random high-frequency noise and a slowly fluctuating baseline. The Gaussian peaks are the interesting patterns we want to detect. Shown as the grey curve in the bottom right picture of Fig. 6.6, the artificial dataset is composed of 100,000 data points, including 481 useful Gaussian peaks. We take the first 20,000 data points as the training dataset, which includes 104 interesting patterns, and take the remaining 80,000 data points as the testing dataset, which includes 377 interesting patterns.

As we have a function to describe the pattern of interest, we use a continuous template (Gaussian function) here, an instance of which is shown as the bottom left picture of Fig. 6.6. The template can be marked with 5 landmarks, in which A and C are the begin and end points; B is the data point that has the maximum

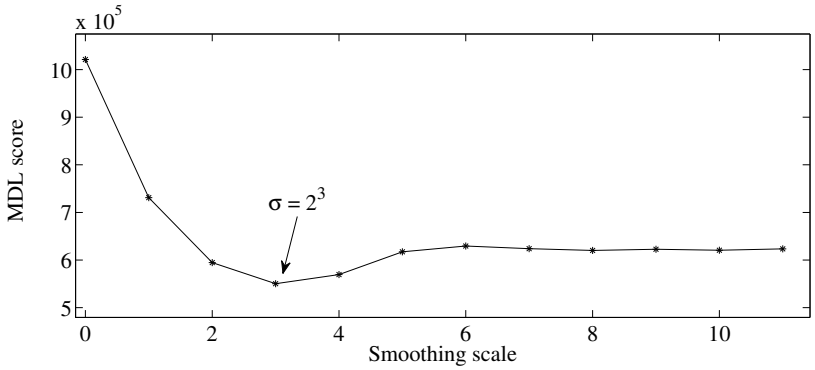

Figure 6.7: The smoothing scale for an artificial dataset - This picture shows a scatter plot between smoothing scale and MDL score, in which the fourth smoothing scale corresponds to the minimal MDL score.

magnitude in the template. We utilise the first-derivative method to extract landmarks from the artificial dataset. Because the first-derivative method is sensitive to noise, we cannot apply it directly to the raw dataset. We first smooth the raw dataset with the convolution method mentioned in Section 6.5.1, and then transform the smoothed dataset to a landmark sequence.

We calculate MDL scores based on a smoothing scale candidate set $\{2^0, 2^1, 2^2, \ldots, 2^{11}\}$, and choose the scale with the minimal MDL score as the right smoothing scale, which is 2^3 in this case, shown as Fig. 6.7. The landmark constraints that succeed in identifying all the 104 interesting patterns in the training dataset are chosen as the target landmark constraints, as follows:

- The length of landmark subsequences should be 3.
- The first and last landmarks should be valley points.
- The second landmark should be a peak point.
- The peak magnitude should be no less than 0.15.

Based on the obtained smoothing scale and landmark constraints, we detect 380 landmark segments, 377 of which are true peaks, and the remaining 3 landmark segments are caused by noise. The precision of the continuous landmark model is thus 99.2%, and the recall is 100%.

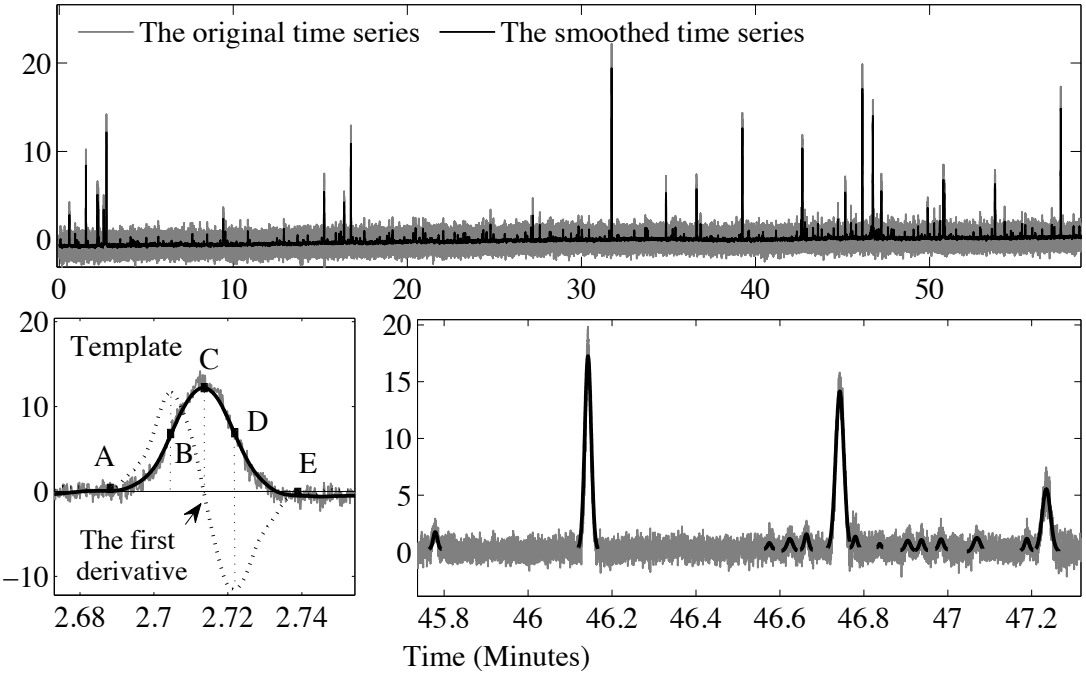

Figure 6.8: Traffic event detection from a traffic dataset - The grey curve in the top picture is the raw strain signal collected from one strain sensor installed on a highway bridge; the black curve in the top picture is the strain signal smoothed with the smoothing scale $\sigma = 2^4$; the black curve in the bottom left picture is the selected discrete template, marked with landmarks A, B, C, D and E; the black peaks in the bottom right picture are landmark models.

6.6.2 Real-life Traffic Dataset

In this section, we apply a discrete template to a real-life traffic dataset collected at the Hollandse Brug. We select a piece of strain signal at 100 Hz of 1 hour at 3:00 AM to detect traffic events, which is composed of 360,000 data points. Using the video record of this period, we label the traffic events as cars or trucks. We take the first 60,000 data points as the training dataset, including 23 cars and 6 trucks, and the remaining 300,000 data points as the test dataset, including 150 car events and 14 truck events.

Following a similar MDL-based procedure as in the previous experiment, the smoothing scale is determined as $2^4 = 16$, shown as Fig. 6.9. The black curve in

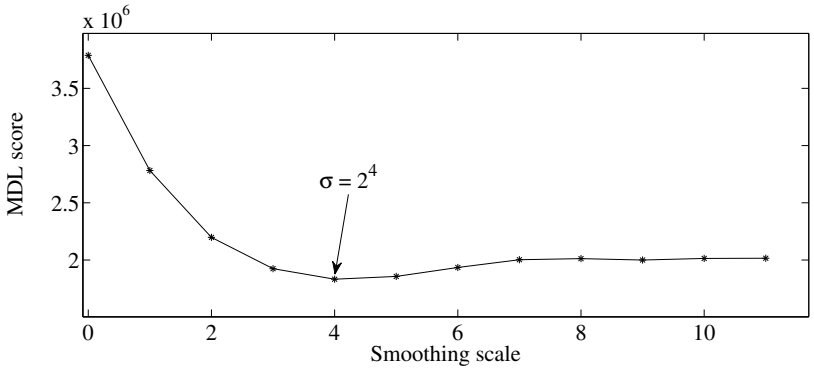

Figure 6.9: The smoothing scale for a traffic dataset - This picture shows a scatter plot of MDL score as a function of smoothing scale, in which the fifth smoothing scale corresponds to the minimal MDL score.

the top picture of Fig. 6.8 is the smoothed curve under this optimal smoothing scale. We select the 6 trucks in the smoothed training dataset for template construction. In order to select the most representative sequence to act as the template, we actually employ MDL as a model selection framework. The truck data that leads to the minimal MDL score (on the training data) is chosen as the discrete template, shown as the peak in the bottom left picture of Fig. 6.8.

The testing dataset is smoothed with a $\sigma = 16$ Gaussian kernel and the landmark constraints are set as:

- The length of landmark subsequences should be 5.
- The first and last landmarks should be valley points.
- The second and fourth landmarks should be inflection points.
- The third landmark should be a peak point.
- The peak magnitude should be no less than 0.44.

Based on the obtained smoothing scale and landmark constraints, we manage to catch 170 landmark segments in the test data: 14 of which correspond to truck events (as validated through the video of the bridge), 150 of which correspond to car events, and the remaining 6 landmark segments are caused by noise.

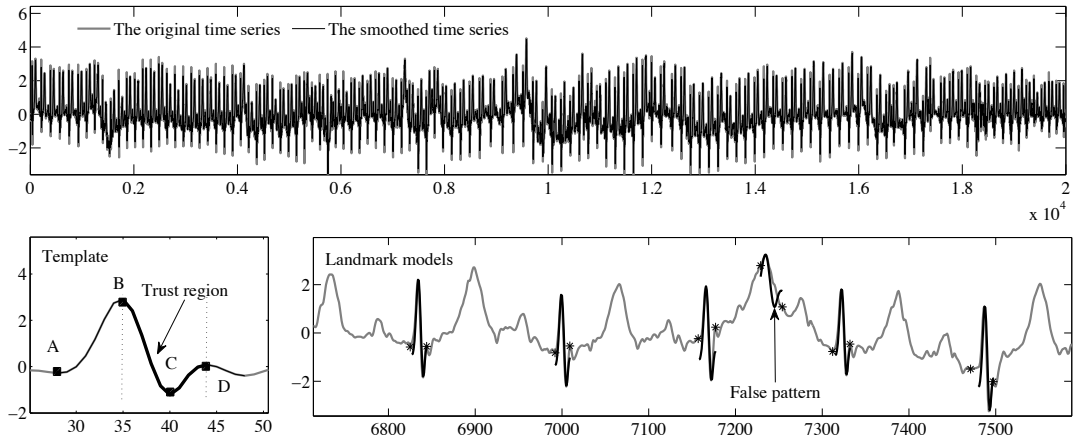

Figure 6.10: The QRS complex detection from an ECG dataset - The grey curve in the top picture is an ECG dataset taken from a real patient, the black curve is a smoothed time series; the curve in the bottom left picture is a discrete template of the QRS complex, which is composed of a R wave and a S wave, and can be marked with four landmarks A, B, C and D; the thick black segment between landmarks B and D is chosen as the trust region; the black peaks in the bottom right picture are landmark models, one of which is a false pattern.

The precision of the continuous landmark model is thus 96.5%, and the recall is 100%.

Our algorithm is quite fast. To give an indication of the efficiency on this relatively large dataset, the total running time was 4.47 seconds on the test set (2.51 seconds for landmark subsequence selection and 1.96 seconds for the landmark models).

6.6.3 ECG Signal

The electrocardiogram (ECG) signal is used to measure the electrical activity of the (human) heart [83]. A single heart beat is typically composed of 5 deflections, called the P, Q, R, S and T wave, in which the Q, R and S waves are often considered together as the QRS complex, because they are closely linked. Note that not every QRS complex contains all the three wave elements, and any combination of

these waves can also be referred to as a QRS complex [84]. Accurately recognising the QRS complex and distinguishing them from the other noise sources such as P and T waves is a critical technology for many clinical instruments [85].

In this section, we choose an ECG dataset of 20,000 data points from [86], which is collected at a frequency of 250 Hz. We take 2,000 data points as the training dataset, which consists of 13 QRS complexes, and the remaining 18,000 data points as the test dataset, containing 106 QRS complexes. Since there is only little noise in the original time series, the optimal smoothing scale comes out as $\sigma = 2^0$ (see also the top picture of Fig. 6.10).

We take the 13 QRS complexes in the training dataset for template construction. The curve in the bottom left picture of Fig. 6.10 is selected as the discrete template. The template can be represented using four landmarks, which are extracted with the first-derivative method. The landmarks B and D are the least sensitive to disturbances, so the landmark segment between these two landmarks is selected as the trust region, shown as the bottom left picture in Fig. 6.10. Based on the training dataset and smoothing scale, the landmark constraints are set as:

- The length of each landmark subsequence should be 4.
- The first and third landmarks should be valley points.
- The second and the fourth landmarks should be peak points.
- The magnitude of the second landmark should be the highest one in the landmark subsequence.
- The magnitude of the third landmark should be the lowest one in the landmark subsequence.
- The temporal difference between the last and the first landmark should be less than 25.
- The magnitude difference between the second and the third landmark should be less than 2.

Based on the obtained smoothing scale and landmark constraints set above, we manage to catch 107 landmark segments: 103 of which are true QRS complexes, 4 of which are false QRS complexes, and 3 true QRS complexes are missing. The precision of the continuous landmark model is thus 96.3%, and the recall is 97.2%. Fig. 6.10 shows one instance of such a false detection, between time points 7,200 and 7,300.

This figure also demonstrates the purpose of landmark models. Note that the detection of predefined patterns (the core of our work) produces a list of consecutive landmarks for each instance of the template detected. When visualising an instance in the actual data, pointing out the landmarks in question is of limited interest. By means of the landmark models, the matching of the template to the actual data can be determined (including unreliable segments of the data), such that the transformed instance of the template can be overlain on the time series, as is demonstrated in the figures in this section.

6.7 Related Work

In this chapter, we have presented three concepts that have been extensively used in image matching fields: templates [87, 88], landmarks [88, 89] and trust regions (or trust features) [80].

Template matching can be used for face detection [90], duplicate document detection [91] and motion classification [92]. The concept of template has been introduced to time series to detect specific patterns or shapes [31, 33, 69, 93, 94]. Frank et al. [94] propose Geometric Template Matching (GeTeM) which uses time-delay embeddings for building models from segments of time series and compares the reconstructed dynamical systems in terms of their state space as well as their dynamics. In [93], a novel and flexible approach is proposed based on segmental semi-Markov models. In [31, 33, 69], meaningful templates are constructed with shape-based averaging algorithms, such as Prioritized Shape Averaging (PSA) [69] and Accurate Shape Averaging (ASA) [31]. Wei et al. propose the Atomic Wedgie method "that exploits the commonality among the predefined patterns

to allow monitoring at higher bandwidths, while maintaining a guarantee of no false dismissals" [68]. Most of the proposed methods are mainly designed for full sequence matching, which are ineffective in detecting predefined patterns from streaming time series.

Landmarks can be used to break time series into meaningful segments, and a template can be featured by a vector of landmarks. Landmarks are also referred to as key-points [74], break-points [75] and change-points [76]. Perng et al. [73] propose a feature-based technique called the landmark model, which uses landmarks instead of the raw data for processing. A two-level representation [74] is proposed to recognise gestures, using both local and global features. In practice, the reliability of each landmark varies with its location. To the best of our knowledge, this hasn't been mentioned in the literature.

It has been pointed out by researchers that some unspecified portions of streaming time series should be ignored [81, 95] to achieve a better result, which means some data points have nothing to do with predefined patterns, and should be filtered out. Ye and Keogh [96] propose a new time series primitive, time series shapelets, for time series classification. The shapelets are informally defined as the subsequences that are in some sense maximally representative of a class. This method is interpretable and accurate in classifying static time series [97], but is ineffective in handling streaming time series. Inspired by these works, we introduce trust region (trust feature) into streaming time series to obtain a more reliable landmark model.

A number of representation methods have been developed in the literature to reduce the dimensionality of time series, such as Discrete Fourier Transform (DFT) [25], Single Value Decomposition (SVD) [98], Discrete Wavelet Transform (DWT) [99]. There are also some researchers who employ symbolic representations, such as Symbolic Aggregate approximation (SAX) [70] and bit-level approximation [71]. Features extracted from time series carry summarized information of the time series [27, 100], which can represent the original time series concisely [101], and are less sensitive to noise [102], so the feature extraction operation can also be used to reduce dimensionality (reduce the size of the data), such as Amplitude-Level Features (ALF) [103], characteristic-based clustering (CBC) [100]. Some

representations are based on piecewise techniques, such as Piecewise Linear Approximation (PLA) [75], Piecewise Aggregate Approximation (PAA) [72], Adaptive Piecewise Constant Approximation (APCA), Derivative Time Series Segment Approximation (DSA) [101, 104] and Piecewise Vector Quantized Approximation (PVQA) [105, 106]. Some representations aim to keep both local and global information about the original time series, such as Multi-resolution Vector Quantization (MVQ) approximation [107] and multi-resolution PAA (MPAA) [108].

Next to the representation methods, a number of similarity measures have been proposed [24], of which the Euclidean Distance (ED) [25, 26] is the most common [27, 28]. However, when shifting and temporal distortions exist in the given subsequences, the ED is proven to be ineffective [28]. To handle stretching and compression along the temporal dimension, Dynamic Time Warping (DTW) [30] was proposed, which achieves an optimal temporal alignment through detecting the shortest warping path in a distance matrix [24, 31, 32, 33]. Finding the shortest warping path is a non-trivial problem, whose computation complexity can reach $O(n^2)$, where n is the number of data points. To speed up the computation of DTW, some lower bounding constraints, like LB_Keogh [32, 36] and the Ratanamahatana-Keogh Band [37], have been introduced to prune expensive computations, which can reduce the complexity to $O(n)$. There are also some other edit-based methods proposed to handle outliers and noise [24], such as Longest Common Subsequence (LCSS) [109], Edit Distance with Real Penalty (ERP) [110] and Edit Distance on Real sequence (EDR) [111]. However, most of the proposed methods focus mainly on temporal deformations [93], which are inadequate in dealing with shifting and scaling in the amplitude dimension [28]. Consequently, Spatial Assembling Distance (SpADe) [28] is proposed to handle shifting and scaling in both the temporal and amplitude dimensions.

6.8 Conclusion

Predefined pattern detection from streaming time series is a quite challenging topic, because it is not only sensitive to noise, but also sensitive to temporal and

magnitude deformations. A number of representation and similarity measure methods have been proposed to approximate interesting subsequences, but most of them are mainly designed for full sequence matching, and are ineffective when the disturbances mentioned above exist. Based on MDL, we smooth the streaming time series with a reasonable scale, and construct a template from the smoothed training time series. The template stands for the pattern of interest that we want to extract from the streaming time series. To carry out this task, we proposed a three-stage representation method, which first transfers the time series into a landmark sequence, and then utilizes the constraints within a template landmark sequence to select promising landmark subsequences of interest patterns, finally employing the trust region (or trust feature) to model candidate patterns. Most of the existing feature-based methods just focus on the quality of models, and pay little attention to the reliability of candidate patterns. Our landmark model overcomes this problem by transferring the template in both temporal and magnitude dimensions according to trust regions (or trust features).