Int. Symp. on Intelligent Data Analysis 2016, pp 73-85. The final publication is available at Springer via https://doi.org/10.1007/978-3-319-46349-0_7

Visual Perception of Discriminative Landmarks in Classified Time Series

T. Sobek and F. Höppner

Ostfalia University of Applied Sciences
Dept. of Computer Science, D-38302 Wolfenbüttel, Germany

Abstract Distance measures play a central role for time series data. Such measures condense two complex structures into a convenient, single number – at the cost of loosing many details. This might become a problem when the series are in general quite similar to each other and series from different classes differ only in details. This work aims at supporting an analyst in the explorative data understanding phase, where she wants to get an impression of how time series from different classes compare. Based on the interval tree of scales, we develop a visualisation that draws the attention of the analyst immediately to those details of a time series that are representative or discriminative for the class. The visualisation adopts to the human perception of a time series by adressing the persistence and distinctiveness of landmarks in the series.

1 Motivation

One can think of many different properties of time series that may or may not contribute to the similarity of two series. The literature offers a broad variety of similarity measures to address different properties, which were extensively tested in a classification setting (e.g. 1-nearest neighbour classifier) [6]. This is helpful if some black-box decision has to be made, but to gain insights into the similarity of time series, a distance measure or 1NN-classifier is almost pointless, as it provides no summary or model, let alone a visual representation of what makes a series more likely to belong to class A than B.

With this work we aim at a visual tool to highlight features in (labelled) time series that discriminate series from different classes, thereby supporting an analyst in understanding and interpreting the series. To achieve this we rely on landmarks in the series, such as minima and maxima, as these properties are also well perceived when visually inspecting series, but are dealt with counter-intuitively by prominent measures such as dynamic time warping (DTW) as we will see below. The identification of discriminative properties offers the opportunity to identify and appreciate even small features, which may not influence a distance measure to a significant extent. The difficulty is, however, to identify and match such features, as they may be affected by effects such as dilation, translation, scaling, noise, etc.

In the next section we briefly review DTW, because it is the most prominent elastic distance measure for time series; we discuss in particular why distance measures such as DTW may miss important structural properties (landmarks). In Sect. 3 we present an approach to identify, match and compare features of time series at multiple scales, which leads us directly to a visualisation of discriminative features. The method is evaluated experimentally in Sect. 4. Conclusions will be given in Sect. 5.

2 Related Work

By $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{R}^m$ we denote a time series that consists of m values indexed from 1 to m. For the sake of simplicity, we refer to the indices i as the points in time when x_i was measured. While there is considerable work about the visualisation of temporal data [1], there is only little about the visualization of a set of labelled series. Several authors, e.g. [2], use multidimensional scaling or projection techniques to visualize a set of series in a scatter plot (one dot representing one series) by means of some distance measures (such as DTW). These approaches do not aim at showing individual or distinctive properties of the series (in contrast to this work), but to give a general overview.

2.1 Similarity Measures vs Landmarks

Given two series $\mathbf{x}$ and $\mathbf{y}$, Euclidean distance $d^2(\mathbf{x}, \mathbf{y}) = \sum_i (x_i - y_i)^2$ assumes a perfect alignment of both series as only values with the same time index are compared. If the series are not aligned, x_i might be better compared with some $y_{f(i)}$ where $f : \mathbb{N} \to \mathbb{N}$ is a monotonic index mapping. With dynamic time warping (DTW) [3] the optimal warping path f, that minimizes the Euclidean distance of $\mathbf{x}$ to a warped version of $\mathbf{y}$, is determined. An example warping path f is shown in Fig. 1(left): Two series are shown along the two axes; $\mathbf{x}$ on the left, $\mathbf{y}$ at the bottom. The matrix enclosed by both series encodes the warping path: an index pair (i, j) on the warping path denotes that x_i is mapped to $y_j = y_{f(i)}$. All pairs (i, j) of the warping path, starting at index pair $(0, 0)$ and leading to (m, m), contribute to the overall DTW distance. Despite the fact that DTW is rather old, recent studies [6] still recommend it as the best measure on average over a large range of datasets.

However, value and time are treated differently in time warping approaches: while a monotone but otherwise arbitrary transformation of time is allowed, the values remain untouched during this procedure.[1] This may lead to some surprising results. In Fig. 1(middle) we have two similar series (linearly decreasing, increasing, decreasing segments), depicted in red and green. They are also shown on the x- and y-axis in the leftmost figure, together with the warping path. Both series were standardized, but their range is not identical. If we would ask a human to align both series, an alignment of the local minima and maxima would be natural, revealing the high similarity of both series as they behave

[1] In order to get meaningful results with time warping methods, the value range of both series should clearly overlap, as it may be obtained from standardization (to zero mean and unit variance).

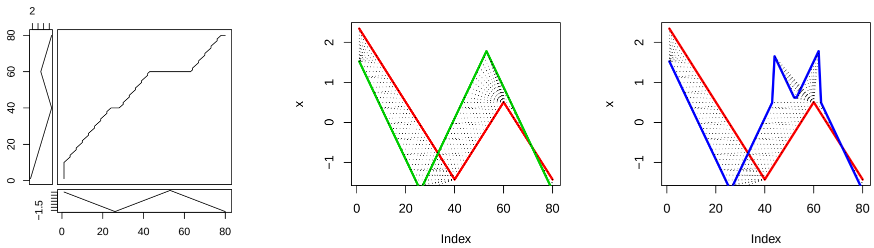

Figure 1. Behaviour of time warping distances. Left: warping path of two time series (also shown in green and red in the middle). Mid and right: examples series; dotted lines indicate the DTW assignment. Right: Pair of blue and red series; structurally different, but with the same distance as green and red series.

identically between the local extrema. The local maximum m of the red curve (near $t = 60$), however, lies below the local maximum of the green curve, so all DTW approaches assign the red maximum to *all points of the green curve above* m. (The assignment is shown in the leftmost figure and by the dotted lines in the middle.) As a consequence, if we shuffle or reorder the green data above m (cf. rightmost subfigure, blue curve), neither the assignment nor the distance changes. This is in contrast to the human perception, who would never consider the blue series being as similar to the red series as the green.

In this example, a human recognizes the red and green curve as similar because of the similarity of the segments (as suggested by the extrema). This kind of similarity includes time warping to compensate for different segment lengths, but also segmentwise value re-scaling (not done with DTW). The natural segmentation along extrema is also propagated by other authors, e.g. [5] in their landmark model. Landmarks correspond to extrema in the time series and a distance measure is defined on the sequence of landmarks rather than the original series. Landmarks are often employed for time series segmentation, but only seldomly for comparing series directly.

2.2 Interval Tree of Scales

We need not consider all extrema to grasp a time series. In the landmark model of [5], some extrema are skipped based on some a priori defined thresholds. This is typical for smoothing operations, but it is difficult to come up with such a fixed threshold, because different degrees of smoothing may be advisable for different parts of the series. Too much smoothing bears the danger of smearing out important features, too little smoothing may draw off the attention from the relevant features.

This is acknowledged by multiscale methods such as wavelets [4]. Witkin was one of the first who recognized the usefulness of a scale-space representation of time series [7]. The *scale* s denotes the degree of smoothing (variance of Gaussian

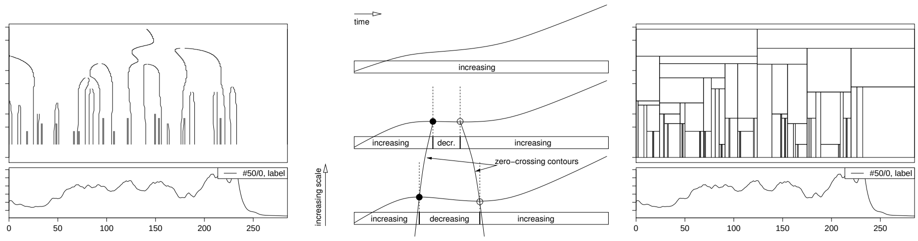

Figure 2. Left: Depending on the variance of a Gaussian smoothing filter (vertical axis, logarithmic) the number and position of zero crossings in the first derivative varies. Mid: The zero-crossings of the first derivative (extrema in the original series) vanish pairwise. Right: Interval tree of scales obtained from left figure.

filter) that is applied to the time series. The scale-space representation of a series depicts the location of extrema (or inflection points) as the scale s increases (cf. Fig. 2(left) for the time series shown at the bottom). The prominence (persistence against smoothing) of an extremum can be evaluated by following it from the original series ($s \approx 0$) to the scale s at which it vanishes (where it gets smoothed away). The scale-space can be considered as a fingerprint of the time series.

Zero-crossings typically vanish pairwise, three consecutive segments (e.g. increasing, decreasing, increasing) turn into a single segment (e.g. increasing), cf. Fig. 2(middle). The scale-space representation can thus be understood as a ternary tree of time series segments where the location of zero-crossings determine the temporal extent of the segment and the (dis-) appearance of zero-crossings limit the (vertical) extent or lifetime of a segment. By tracing the position of an extrema in the scale-space back to the position at $s \approx 0$ we can compensate the displacement caused by smoothing itself. We may thus construct a so-called interval tree of scales [7] (cf. Fig. 2(right)), where the lifetime of a monotone time series segment is represented by a box in the scale-space: its horizontal extent denotes the position of this segment in the series, the vertical extent denotes the stability or resistence against smoothing. Rather than choosing a single smoothing filter beforehand, such a tree represents the time series at multiple scales and allows different views or perspectives on the same series. We consider this to be advantegous for our purpose, because we do not know at which level discriminative features may occur.

3 Visualising Discriminative Features

We have seen that the distance values obtained by established methods such as DTW may be misleading when investigating structural properties of time series. We are looking for a way to visualize discriminative, structural features in sets of labelled series. When adressing the analysts perception, we should adjust the algorithmic approach to the way humans perceive time series. While the temporal alignment of DTW is objective function-driven, humans align time series by aligning landmarks and matching the corresponding segments between the landmarks. The underlying idea of this work is to use the interval tree of scales as the core for the visualisation as it encodes landmarks already (extrema or inflection points). By matching series from the same and/or from different classes we recognize which cases have which landmarks in common. By complementing the interval tree with this information it becomes a tool to not only visualize the structure of a series, but also to distinguish which parts are shared among classes and which help to distinguish classes.

3.1 Graph Representation of the Interval Tree of Scales

We consider the graphical depiction of the interval tree as a tesselation of the time-scale space, which encodes *all possible perceptions* of a time series. We represent a tile in this tesselation that covers the temporal range $[t_1, t_2]$ and the scale range $[s_1, s_2]$ by a quintuple (t_1, t_2, s_1, s_2, o) with $t_1 < t_2$, $s_1 < s_2$ and orientation $o \in \{\text{increasing}, \text{decreasing}\}$. We define a graph representation $G_\mathbf{x} = (V, E)$ of the interval tree as follows: The set of all tiles is denoted as V and makes up the set of nodes in our graph. Two tiles $v = (t_1^v, t_2^v, s_1^v, s_2^v, o^v)$ and $w = (t_1^w, t_2^w, s_1^w, s_2^w, o^w)$ are connected, that is $(v, w) \in E$, if and only if they are adjacent in time

$$t_2^v = t_1^w \tag{1}$$

We define a subset $V_S \subseteq V$ (resp. $V_E \subseteq V$) that contains all start-tiles (resp. end-tiles), that is, tiles which do not have a predecessor (resp. successor) in the graph. A path of n tiles $(v_1, \ldots, v_n)$ in $G_\mathbf{x}$ is called *perception* of series $\mathbf{x}$ if $\forall i : (v_i, v_{i+1}) \in E$, $v_1 \in V_S$ and $v_n \in V_E$. Such a *perception* represents a segmentation of the time series $\mathbf{x}$ because the time periods of the tiles v_i represent a segmentation of the time range $[1, m]$: Apparently we have $t_1^{v_1} = 1$ (because $v_1 \in V_S$), $t_2^{v_n} = m$ (because $v_n \in V_E$) and subsequent time periods touch due to (1). The sequence of orientiations is alternating between increasing and decreasing due to the properties of the original interval tree of scales. The set of all *perceptions* corresponds to all possible paths from a leftmost tile to a rightmost tile, that is, a view of the series $\mathbf{x}$ with a (possibly) different degree of smoothing within each segment.

Cf. Fig. 3: The interval tree of the series on the left (or bottom) consists of 9 (or 7) tiles. The graphs are superimposed on the interval tree. Nodes belonging to V_S (or V_E) are connected to the virtual node S (or E). From these particular graphs we find two perception for the left series ((a_0, a_1, a_2, a_4) and $(a_0, a_1, a_2, a_3, a_5, a_6)$) and four for the bottom series (e.g. $(b_0, b_2, b_4, b_7, b_8)$).

3.2 Matching Perceptions

In this section we define a distance measure for two time series $\mathbf{x}$ and $\mathbf{y}$ by means of their respective graphs $G_\mathbf{x}$ and $G_\mathbf{y}$. To compare time series we have to decide which *parts* of both series correspond and, once the assignment has been made, how well they match. With Euclidean distance or DTW these *parts* correspond to the values at individual time points. Here we assign *time periods* or segments of both series to each other and then compare the subseries of the respective time periods. The segments we consider for this assignment correspond to the tiles in the interval tree (or vertices of each graph). As we want to match the full series we have to cover all points from both series exactly once (with some possible exceptions near the beginning and end of the series) and keep the temporal order of the segments – that is, we have to match perceptions.

Suppose we have two similar series, where one time series $\mathbf{x}$ contains a certain landmark while the other $\mathbf{y}$ does not. When perceiving all the details (path through tiles near the bottom of the interval tree), both series do not match structurally in the number of tiles (because $\mathbf{x}$ has an additional landmark). To perceive them as similar, we have to switch to a coarser scale for $\mathbf{x}$ in the temporal region where the additional landmark resides. That is, a structural comparison of time series corresponds to finding the right path through both graphs $G_{\mathbf{x}}$ and $G_{\mathbf{y}}$ such that the tiles in both sequences correspond to each other. (In the example case of Fig. 3 we have to find one perception (out of 2) for the left series and one perception (out of 4) for the bottom series that correspond best, e.g. (a_0, a_1, a_2, a_4) and (b_1, b_4, b_7, b_8).)

However, given two arbitrary paths $p_{\mathbf{x}} = (v_1^{\mathbf{x}}, \ldots, v_k^{\mathbf{x}})$ from $G_{\mathbf{x}}$ and $p_{\mathbf{y}} = (w_1^{\mathbf{y}}, \ldots, w_l^{\mathbf{y}})$ from $G_{\mathbf{y}}$, the tiles $v_i^{\mathbf{x}}$ and $w_i^{\mathbf{y}}$ may not be directly comparable: as we intend to perform a structural match, segments of different type must not be aligned, e.g., we do not assign increasing $w_1^{\mathbf{y}}$ to decreasing $v_1^{\mathbf{x}}$. (Fig. 3: a_0 cannot be associated with b_0.) Either we have to switch to a different level of abstraction again (that is, different perceptions) – or we allow to skip short segments near the beginning and the end of the series. We express this by an alignment $\delta \in \mathbb{N}$ such that tile $w_{1+\delta}^{\mathbf{y}}$ is assigned to $v_1^{\mathbf{x}}$ with $o^{v_1} = o^{w_{1+\delta}}$. (Ex. from Fig. 3: to match $p_{\mathbf{x}} = (a_0, a_1, a_2, a_4)$ to $p_{\mathbf{y}} = (b_0, b_2, b_4, b_7, b_8)$ we need to associate the first node of $p_{\mathbf{x}}$ with the second node of $p_{\mathbf{y}}$, that is, set the offset $\delta = 1$ leading to a comparison of $a_0 \sim b_2, a_1 \sim b_4, \ldots$)

Eventually we compare the shape of the series within a segment. Given a tile $v = (t_1^v, t_2^v, s_1^v, s_2^v, o^v) \in V$ we denote the subseries of $\mathbf{x}$ that the tile refers to by $\mathbf{x}|_v := (x_{t_1^v}, x_{t_1^v+1}, \ldots, x_{t_2^v})$. Given two paths (plus an alignment δ) we evaluate how well the assigned segments match each other by means of a dissimilarity measure d' to compare $\mathbf{x}|_{v_i}$ against $\mathbf{y}|_{w_{i+\delta}}$. As the assigned segments need not be of the same length, the distance $d'(v_i^{\mathbf{x}}, w_{i+\delta}^{\mathbf{y}})$ must cope with segments of different lengths. This might be achieved by stretching one series to the length of the other and apply Euclidean distance afterwards. Other choices will be discussed below.

Thus, among all possible matches of perceptions of time series $\mathbf{x}$ and $\mathbf{y}$, choose the one that minimizes the sum of distances of the corresponding segments:

$$d(\mathbf{x}, \mathbf{y}) = \min_{(v_1^{\mathbf{x}},\ldots,v_k^{\mathbf{x}}) \in G_{\mathbf{x}}, (w_1^{\mathbf{y}},\ldots,w_l^{\mathbf{y}}) \in G_{\mathbf{y}}, \delta} \sum_{i=\max\{1,1-\delta\}}^{\min\{k,l+\delta\}} d'(v_i^{\mathbf{x}}, w_{i+\delta}^{\mathbf{y}}) \tag{2}$$

The search for the minimal distance (and thus the best match) can be formalized as a weighted shortest path problem in the graph $\mathcal{G}_{\mathbf{x},\mathbf{y}} = (V_{\mathbf{x},\mathbf{y}}, E_{\mathbf{x},\mathbf{y}})$ with

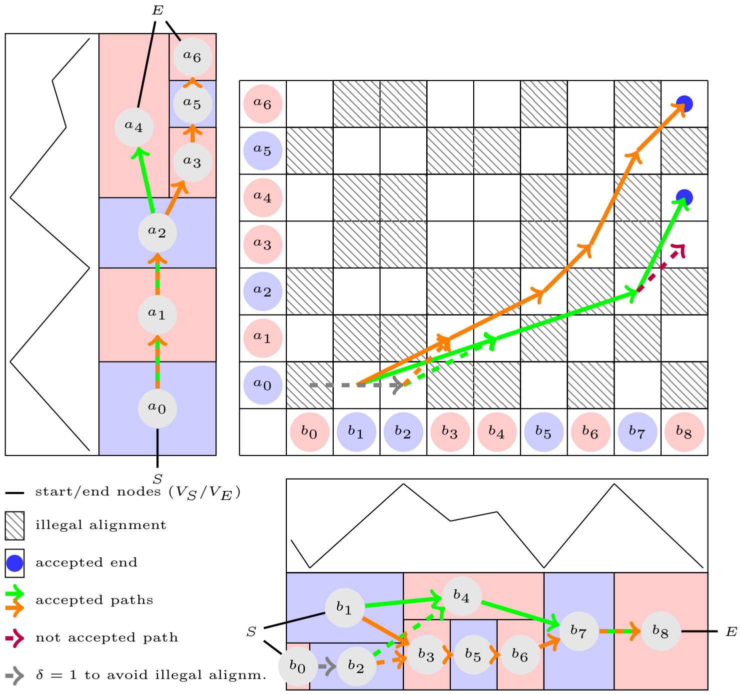

Figure 3. Two series (left and bottom), represented by their interval tree and respective graph (slope is color-coded). The matrix in the center encodes the assignment of nodes from both graphs, a match of perceptions from both series is thus a path from the bottom left to top right edge of the matrix.

$V_{\mathbf{x},\mathbf{y}} = V_{\mathbf{x}} \times V_{\mathbf{y}}$ and $E_{\mathbf{x},\mathbf{y}} \subseteq V_{\mathbf{x},\mathbf{y}}^2$ where $\forall v, v' \in V_{\mathbf{x}}, \forall w, w' \in V_{\mathbf{y}}$: $((v, w), (v', w')) \in E_{\mathbf{x},\mathbf{y}} \Leftrightarrow (v, v') \in E_{\mathbf{x}} \wedge (w, w') \in E_{\mathbf{y}} \wedge o^v = o^w$ from an arbitrary start node $(v, w) \in V_{\mathbf{x}}^S \times V_{\mathbf{y}}^S$ to an end node $(v', w') \in V_{\mathbf{x}}^E \times V_{\mathbf{y}}^E$. The search for an optimal δ is reformulated by extending the set of start/end tiles by their adjacent tiles. The edge weights are given by the distance d' among the segments. Standard methods such as the Dijkstra algorithm might be used, but a more efficient solution via dynamic programming is advisable and has been implemented for this work. Similar to DTW we have a matrix of assignments (of tiles rather than points) where we seek for the minimal cost path from a start position (bottom left) to end position (top right) as illustrated by the matrix in Fig. 3. This step has complexity $O(n \cdot m)$ with n and m being the number of nodes in the resp. graph (rather than number of points as in DTW).

3.3 Discriminative Features

Determining the distance between two time series includes the identification of the best-matching perceptions. If two series share many low-level features the corresponding tiles are likely to be included in the optimal assignment, whereas series from different classes may have to retract to segments on a coarser scale (with fewer landmarks) as the details of one series have no counterparts in the other. We complement the interval tree of some series $\mathbf{x}$ of class c with this information: We count (when comparing a series to all other) how often segments were involved in the best match of series from the same as well as from different classes. Based on these numbers we highlight tiles that get primarily matched to series of the same class (or other classes). In the visualization, increasing segments will be coloured in blue, decreasing in red, but the opacity is determined by the entropy of the distribution "same class vs other class" weighted by the total number count (a distribution 5:0 is less relevant than 25:0). Furthermore, we normalize both counts to 100 to avoid a bias towards the *other classes* in multiclass problems (otherwise we expect only a fraction of $\frac{1}{k}$ cases from class c and $\frac{k-1}{k}$ cases from other classes (bias towards 'other class')).

Apart from counting, upon matching all series to one series $\mathbf{x}$, we may collect for each of its tiles all segments from other series that were assigned to it. From these segments we can derive secondary features such as maximal difference (in value), slope, curvature, variance, etc. and construct a classical dataset (with a fixed number of attributes) for each tile and feed it into a standard classifier to identify features that help to distinguish the classes. In this work we consider only the maximal difference (difference between end points of a tile) and report its utility for distinguishing classes by means of the weighted accuracy[2].

Among all possible visualisations (one per time series) we automatically select one per class that offers a perception where the discriminative features are pronounced best and present them to the analyst. Each tile in the interval tree is annotated with the (weighted) class distribution (same vs other class) and, below, the (weighted) accuracy (only if above 65%). The bottom left tile in Fig. 5(l) reads as follows: this tile is matched (in the optimal assignement) with 62% (28%) of series from the same (other) class(es) and a classifier on the height difference alone yields 66% (weighted) accuracy.

3.4 Segment Distance

For equation (2) we employ a distance measure $d'(v_i^{\mathbf{x}}, w_j^{\mathbf{y}})$ that is used to compare segments. By scaling the segments and applying Euclidean distance, (2) becomes quite similar to DTW, only that we do not allow arbitrary warping but linear warping between extrema. However, as we have showcased in Sect. 2.1, Euclidean distance and DTW miss a possibility of vertical scaling. We could use Pearson correlation instead, because its built-in normalization compensates for different ranges in both segments. Here, to better focus on structural distances and get most of the interval tree, we treat the value range identical to the temporal range, that is, rescale and shift both series in both dimensions such that their start and

[2] As before, we weigh the cases such that the total weight of series from the same class and series from a different class becomes identical to get accuracies independent of the number of classes.

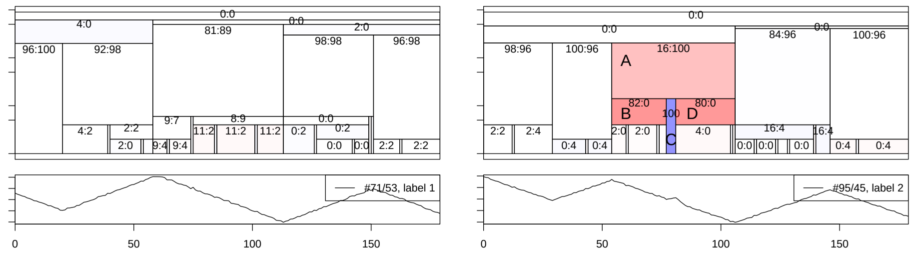

Figure 4. Series from the right class have a small bump imputed. The tile marked A corresponds to a perception of the series where the bump has been smoothed away; it matches 100% of the *series from other classes*, while the tiles B-D occur mainly for series of the same class. This turns the tiles A-D into interesting visualization features.

end points coincide (in time and value). Then we apply Euclidean distance to the rescaled series (denoted by $d_{ES}(\mathbf{x},\mathbf{y})$ in (3)) and thereby capture differences in the shape of the segments. But there are more aspects than just shape if we want to adopt to the visual perception: The duration of the segments, the difference in value (range) and in particular the importance, that is, the persistence against smoothing. We therefore penalize d_{ES} by additional factors:

$$\underbrace{d'(v^{\mathbf{x}}, w^{\mathbf{y}})}_{\text{segment distance}} = \underbrace{d_{ES}(\mathbf{x}|_v, \mathbf{y}|_w)}_{\text{shape distance}} \cdot \underbrace{f(\Delta t_v, \Delta t_w)}_{\text{delta in duration}} \cdot \underbrace{f(\Delta y_v, \Delta y_w)}_{\text{delta in height}} \cdot \underbrace{f(\Delta s_v, \Delta s_w)}_{\text{delta in importance}} \tag{3}$$

where for some tile u of series $\mathbf{z} = (z_1, \ldots, z_n)$ we define $\Delta t_u = |t_2^u - t_1^u|$, $\Delta y_u = |z_{t_2^u} - z_{t_1^u}|$, $\Delta s_u = |s_2^u - s_1^u|$ and $f(x,y) = \frac{\max\{x,y\}}{\min\{x,y\}}$. We thus penalize d_{ES} by a factor of 2 if one segment is twice as long, tall, or important (persistent) as the other.

4 Experimental Evaluation

4.1 Sanity Check

We illustrate the approach by using an example similar to that of Fig. 1 in the introduction. All series consist of five linear segments with varying duration that were standardized during preprocessing. Half of the series have a small bump in the third segment (downward slope), which is somewhat more prominent than the bumps that were introduced by Gaussian noise. Standard 1-NN classifiers based on Euclidean or DTW distance have difficulties with this simple setting, they both reach only accuracies close to 50%. The influence of the small bump on the distance value will be rather small, a conventional distance measure will have a hard time in distinguishing both classes.

The visualisation derived from the pairwise comparison is shown in Fig. 4. While nothing of interest shows up for the class where the bump is absent, the visualisation for the other class clearly reveals the relevant features. Note that the noise introduces local extrema, which vanish quickly as the scale increases, the small bump survives somewhat longer. Some series from the same class match not only the bump but also the noise, but they are small in number. While all of the series from the other class match this segment on a coarse scale (tile A), most of the series from this class subdivide tile A into three subtiles B, C, D. This is easily recognizable from the visualisation, which therefore supports a human in interpreting and understanding the differences of both classes.

4.2 Series from the UCR repository

In this section we discuss the results on some datasets from the UCR time series repository. All series were standardized in advance. We show a diversive subset from these datasets: motion capture (Gunpoint), shape (Plane, Fish), and mass spectrometry (Coffee). For many other datasets of the same type the visualisation led to comparable results. The visualisation provides less informative results if series from different classes hardly share common properties, but in this case a sophisticated search for discriminative features is not necessary anyway.

Fig. 5 shows a series of class 1 from the Coffee dataset. Series from both classes are quite similar in shape, the visualization draws our intention directly to some interesting differences. While 73% of the series from class 1 exhibited a small, local extremum (tile B), most of the series from class 0 do not have this feature (tile A). A similar observation can be made at tile C and its subnodes. The number 89 in tile C denotes a (weighted) accuracy of 89% obtained from the absolute difference in value alone (between start and end points of this tile).

Fig. 6 displays a series of class 1 from the Gunpoint dataset. For this dataset, we did not use the zero crossings of the first but the second derivative, that is, the interval tree of scales characterizes inflection points rather than extrema. The series record the hand position of subjects drawing a gun, aiming, and returning it to the holster – or performing the same motion without a gun. The visualisation shows the relevance of small features when drawing and returning the gun from/to the holster, which are absent with series from the other class.

Fig. 7 depicts a series of class 2 from the Fish dataset. We immediately recognize that the small local extrema in the first large increasing and the first large decreasing segment (in [0,100] and [120,220], resp.) are characteristic for the majority of the series from class 2. The upper left tile describes the segment of all series from the beginning to the first minimum that persists over all (shown) scales. The difference in value (from start to end of the segment) is, according to the value 85 shown in the tile, sufficient to achieve a 85% (weighted) accuracy in predicting this class vs. any other class.

Finally, Fig. 8 shows an example from class 6 of the Plane dataset. Again, the visualisation emphasizes the discriminative features well: there is a characteristic series of small local extrema near $t \approx 70$ for class 6. Most series from other classes share the coarse decreasing and increasing segments only, the local extrema are discriminators for class 6 rather than just noise.

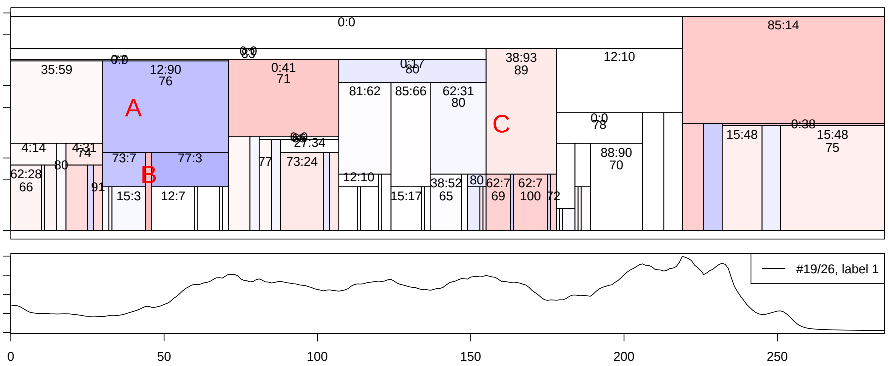

Figure 5. Extrema of coffee data set, class 1.

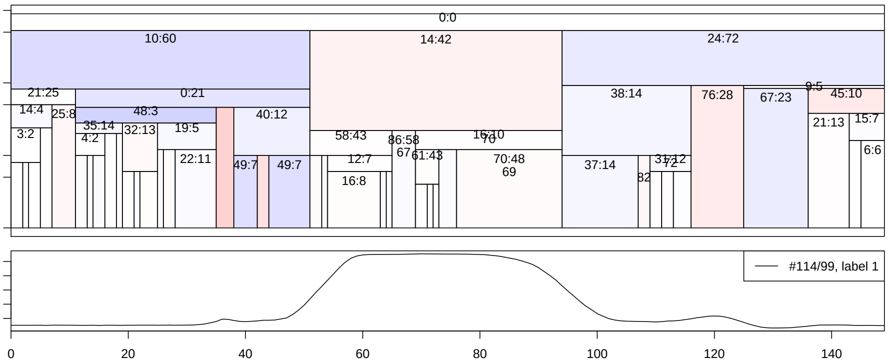

Figure 6. Inflection points of gunpoint data set, class 1.

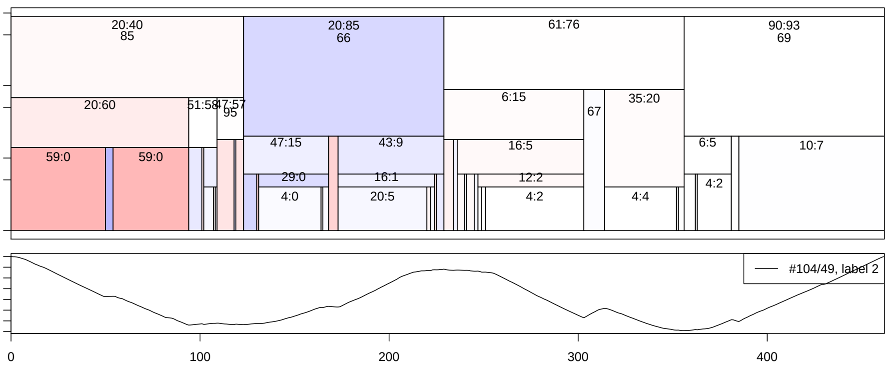

Figure 7. Extrema of fish data set, class 2.

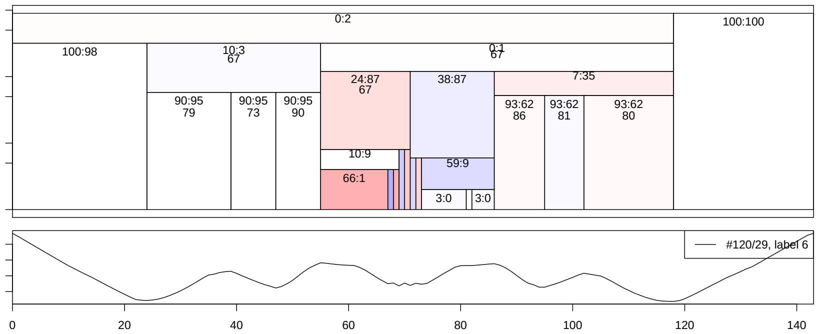

Figure 8. Extrema of plane data set, class 6.

5 Conclusion

In this preliminary work we have proposed a visual tool that helps an analyst to review and explore classified time series. The classification itself was not the focus of this work, but to support the data understanding and the assessment of structural differences to simplify, e.g. subsequent preprocessing steps. We have demonstrated that classification based on distance measures may hold some counterintuitive intricacies, therefore the explicit goal of the visualisation was to correspond to the visual perception of the series. In the experiments the method successfully delivered insights for the class-wise distinction of time series. A possible direction for future work may be the use of the annotated tiles for relevance-weighted distances (cf. relevance feedback in information retrieval).

References

1. W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011.
2. A. B. Alencar, F. V. Paulovich, R. Minghim, M. G. D. A. Filho, and M. C. F. D. Oliveira. Similarity-Based Visualization of Time Series Collections: An Application to Analysis of Streamflows. *International Conference Information Visualisation*, pages 280–286, jul 2008.
3. D. J. Berndt and J. Clifford. Finding Patterns in Time Series: A Dynamic Programming Approach. In *Advances in Knowledge Discovery and Data Mining*, pages 229–248. MIT Press, 1996.
4. S. G. Mallat. *A Wavelet Tour of Signal Processing*. Elsevier Ltd, 2001.
5. C.-S. Perng, H. Wang, S. R. Zhang, and D. S. Parker. Landmarks: a new model for similarity-based pattern querying in time series databases. In *Int. Conf. Data Engineering*, 2000.
6. X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, feb 2012.
7. A. P. Witkin. Scale Space Filtering. In *Artifical Intelligence (IJCAI)*, pages 1019–1022, Karlsruhe, Germany, 1983.