Optimal Filtering for Time Series Classification

F. Höppner

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

Abstract The application of a (smoothing) filter is common practice in applications where time series are involved. The literature on time series similarity measures, however, seems to completely ignore the possibility of applying a filter first. In this paper, we investigate to what extent the benefit obtained by more complex distance measures may be achieved by simply applying a filter to the original series (while sticking to Euclidean distance). We propose two ways of deriving an optimized filter from classified time series to adopt the similarity measure to a given application. The empirical evaluation shows not only that in many cases a substantial fraction of the performance improvement can also be achieved by filtering, but also that for certain types of time series this simple approach outperforms more complex measures.

1 Motivation

Time series comparison became almost a standard operation just like comparing ordinal or numerical values with tabular data. A broad range of different similarity measures has been proposed in the literature, ranging from simple and straightforward measures such as Euclidean distance (ED) to more complex measures that deal with temporal dilation and translation effects such as dynamic time warping (DTW [1]). Extensive comparative studies have been carried out to compare these measures over a variety of datasets [8,9].

While some scenarios may call for a highly flexible measure, such a measure may perform worse where the flexibility is not needed to solve the task. Just like with classifiers it may misuse its parameters to *overfit*, which is unlikely to happen with simpler measures. Thus, if a comparable performance can be achieved, we should go with the simpler measure (Occam's Razor). The investigation of the literature on time series similarity measures reveals a surprising fact: in this context, *smoothing* of time series is almost completely ignored. The application of a smoothing filter can be considered as a pre-processing step and is applied in many machine learning and data mining applications, but neither plays a role in the comprehensive experimental comparison of time series similarity measures [8,9], nor in a survey on clustering time series data [7] (with distances at the core of any clustering algorithm). This is not to say that the time series in the comparative studies are not pre-processed, in fact they are usually standardized before presented to the similarity measures, or various time series representations are considered, which implicitly perform smoothing (e.g., piecewise approximations) or simplify noise removal (Fourier or Wavelet transform), but filters are not explored in general. This is surprising because noise and outliers are very well perceived as problematic aspects in similarity search. The contribution of this work is to bring filtering techniques back to the conscience of the time series similarity measure community, the proposal of two approaches to derive an optimized filter for measuring time series similarity and an experimental evaluation demonstrating its high potential.

2 Definitions and Discussion

A time series $\mathbf{x}$ of length n is a series $(x_i)_{1 \leq i \leq n}$ of n measurements. The similarity of two series $\mathbf{x}$ and $\mathbf{y}$ is usually measured by means of some distance (or dissimilarity) measure $d(\mathbf{x}, \mathbf{y}) \geq 0$ where a small distance indicates high similarity. Some distance measures assume that both series $\mathbf{x}$ and $\mathbf{y}$ have the same length, but in case this is not given one of the series may be stretched to the length of the other and such measures remain applicable.

Similarity. There are two major groups of similarity measures. *Lock-step measures* directly compare corresponding values of two time series, that is, the i^{th} value of $\mathbf{x}$ with the i^{th} value of $\mathbf{y}$, as in the Euclidean distance. Simple distortions such as an offset in the recording time make two observations of the same process dissimilar under Euclidean distance. Elastic measures identify a *warping path*, a monotone transformation of time, such that series $\mathbf{x}$ corresponds best to the warped series $\mathbf{y}$ (e.g. dynamic time warping and variations thereof [1,2,3]). One can also find various modifications of these measures, e.g., the authors of [5] try to "prevent minimum distance distortion by outliers" by giving different weights to $|x_{p(i)} - y_i|$ depending on the temporal offsets $|p(i) - i|$. But it has also been pointed out in recent publications that simpler concepts may be hard to beat [9] or even outperform complex approaches [4]. Some applications seek similar *subsequences* of time series only, but this is accomplished by applying the same range of similarity measures to data from a sliding window, so the fundamental problem remains the same (only the arguments of the measure change).

Filtering. Filters have a long tradition in signal processing to reduce or enhance certain aspects of the signal. In data analysis, filters are often applied to smooth the original series to remove noise or impute missing observations. Here, we consider (discrete time) linear time-invariant (LTI) filters only. Such a filter may be described by a vector of coefficients $\alpha = (\alpha_{-m}, \alpha_{-m+1}, \ldots, \alpha_{m-1}, \alpha_m) \in \mathbb{R}^{2m+1}$ and the application of the (discrete) filter α to a (discrete) time series $\mathbf{x}$ is defined as the convolution $(\mathbf{x} * \alpha)_i = \sum_{j=-m}^{m} \alpha_j \cdot x_{i+j}$. The convolution $\mathbf{x} * \alpha$ can be considered as a *smoothed version of* $\mathbf{x}$, but for $\mathbf{x} * \alpha$ to have the same length as $\mathbf{x}$ we need to clarify what x_{i+j} may refer to when $i + j < 1$ or $i + j > n$. Circular discrete convolution is frequently applied (index modulo time series length), but there is no justification why the last few values of $\mathbf{x}$ should influence the first few values of a smoothed $\mathbf{x}$. So instead of a circular convolution we define for an arbitrary series $\mathbf{x}$ of length n: $x_i := x_1$ if $i < 1$ and $x_i := x_n$ if $i > n$.

Benefit of Filtering. We argue that the application of filters has not tapped its full potential in the area of time series similarity measures. Such measures are used to compare series against each other with the goal of distinguishing two (or more) types of situations. Any labeled time series dataset may thus be considered as an application domain that defines (by examples) which series should be similar (same class) and which should not (different class). A filter may be tailored to a specific application by focusing on different aspects of time series, which may prove filters to be useful in a broad range of applications.

One important aspect in time series similarity is **temporal locality.** Figure 1(a) shows a simple, artificial set of time series with data from two classes having a positive peak and a negative peak, resp. The exact position of the peak, however, varies. If the peak positions do not match, we obtain the same distance between examples from the same and different classes with Euclidean distance – it is thus not helpful for discriminating series from different classes. An elastic measure, such as DTW, however, should have no problems with a correct alignment. In yet another applications the peak position may be important. In Fig. 1(b) both classes have a positive peak, but this time the exact position is relevant. This is a simple task for Euclidean distance but nearly unsolvable for DTW, which does not care about the exact temporal location. If we choose our filter *wisely*, the combination of a filter with Euclidean distance (smearing out the singleton peak in the first and leaving the data untouched in the second case) may solve both problems. Secondly, **noise** may easily distort a time series similarity measure, because time series are high dimensional data and suffer from the curse of dimensionality. The right amount of smoothing may help to identify the relevant trends in the series and reduce the impact of incidental differences. Thirdly, a filter is a **versatile preprocessor**, it can be used to approximate the slope or curvature (first or second derivative of the original signal). To distinguish classified time series it may be advantegous to inspect these transformed rather than the original series. If we manage to identify the filter that best discriminates series from different classes we increase the versatility of the measure as it can adopt automatically to a broad range of situations.

3 Optimized Filter for Time Series Similarity

The simple application of the *right* filter may solve a variety of problems for similarity measures. Although it seems like a somewhat obvious idea to apply a filter separately (before applying a distance measure), the potential impact of filtering on the discrimination of time series has not been explored before. In this section we propose ways to automatically find the filter that is best suited to distinguish time series from one another, that is, a filter that emphasizes the *important differences* (between series from different classes) and ignore or attenuate the less important ones (between series from the same class). Apparently we assume that some supervision is available by class labels. We consider two alternative approaches in the following subsections and in both cases we assume that N series $\mathbf{x}_i$, $1 \leq i \leq N$, of length n are given with labels $l_i \in L$, $|L|$ being the number of classes. By $\tilde{\mathbf{x}}_i$ we denote the filtered version of $\mathbf{x}_i$ (after applying a filter $\alpha = (\alpha_{-m}, \dots, \alpha_0, \dots, \alpha_m)$ of size $2m+1$, that is, $\tilde{\mathbf{x}} = \mathbf{x} * \alpha$). By $x_{i,j}$ we denote the j^{th} value of series $\mathbf{x}_i$. For the sake of a convenient notation, with $x_{i,j}$ we refer to $x_{i,1}$ for all $j \le 1$ and $x_{i,n}$ for all $j \ge n$.

3.1 A Filter Derived from Pairwise Comparison

As a first proposal consider the following objective function

$$\text{min. } f(\alpha) = \beta \sum_{l_i = l_j} \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2 - \sum_{l_i \neq l_j} \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2 \quad \text{s.t.} \sum_{k=-m}^{m} \alpha_k = 1 \tag{1}$$

where $\sum_{l_i=l_j}$ is an abbreviation for $\sum_{1 \le i,j \le N, i \neq j, l_i = l_j}$ (same for the second sum with $\neq$ rather than $=$). The filter coefficients α are hidden in the smoothed series $\tilde{\mathbf{x}}_i$ on the right hand side of f. Distances between *filtered series* from the same class should be small (first summation), whereas distances between *filtered series* from different classes should be large (second summation). Since the second sum is subtracted, the function has to be minimized overall. The coefficient $\beta \in \mathbb{R}$ is a necessary scaling factor for the first sum, chosen to ensure that f is a convex function ($f \to \infty$ as $\|\alpha\| \to \infty$) that actually has a (global) minimum. (Without the scaling factor the second sum may dominate, turning f into a concave function and the minimum is obtained for $\|\alpha\| \to \infty$.)

Without any constraint on α there is an obvious minimum $\alpha = 0$, but this is apparently an undesired solution because all series would look identical. Here, we require the sum of all filter coefficients to be 1. This constraint ensures that the filtered series stay within the same range as the original series.

Proposition 1. *The optimal filter $\alpha = (\alpha_{-m}, \ldots, \alpha_0, \ldots, \alpha_m) \in \mathbb{R}^{2m+1}$ minimizing (1) is obtained from a linear equation system $A\alpha' = b$ with $A \in \mathbb{R}^{2m+2 \times 2m+2}$, $b = (0, \ldots, 0, 1) \in \mathbb{R}^{2m+2}$, $\alpha' = (\alpha_{-m}, \ldots, \alpha_m, \lambda)$ where*

$$A = \begin{pmatrix} M & 1 \\ 1 & 0 \end{pmatrix} \tag{2}$$

$$M = 2 \sum_{(i,j)} s_{i,j} \sum_{l=1}^{n} \pi_l(\mathbf{x}_i - \mathbf{x}_j) \pi_l(\mathbf{x}_i - \mathbf{x}_j)^\top \in \mathbb{R}^{2m+1 \times 2m+1} \tag{3}$$

$$\pi_l(\mathbf{z}) = (z_{l-m}, \ldots, z_{l+m}) \in \mathbb{R}^{2m+1} \tag{4}$$

with $s_{i,j} = \beta$ for $l_i = l_j$ and $s_{i,j} = -1$ for $l_i \neq l_j$. (The notation $\sum_{(i,j)}$ is an abbreviation for $\sum_{1 \le i,j \le N, i \neq j}$.)

Finally, we have to choose β such that f is guaranteed to be convex. This is accomplished by setting

$$\beta = 1.5 \cdot \max_{-m \le k \le m} \left\{ \frac{M^{\neq}_{k,k}}{M^{=}_{k,k}} \right\} \text{ where } M^{=} = \sum_{l_i = l_j} M_{\mathbf{x}_i, \mathbf{x}_j}, M^{\neq} = \sum_{l_i \neq l_j} M_{\mathbf{x}_i, \mathbf{x}_j}$$

where $M_{\mathbf{x},\mathbf{y}} := \sum_{l=1}^{n} \pi_l(\mathbf{x} - \mathbf{y}) \pi_l(\mathbf{x} - \mathbf{y})^\top$. [Proofs are omitted due to lack of space.] The factor of 1.5 ensures that the coefficients of the quadratic terms are strictly positive, other choices are possible as long as the factor is larger than one (but the larger the factor, the more the first sum of (1) dominates). We obtained satisfactory results with a factor of 1.5 and stick to it throughout the paper.

If we scale the resulting filter α by some scalar, the (squared) Euclidean distances scale by the same factor. While the minimization of (1) yields a unique filter, to discriminate series from different classes we will subsequently order time series by distance (to find the closest match) and this order is not affected by a factor. Therefore we divide α by its norm and arrive at a filter with $\|\alpha\| = 1$.

3.2 A Filter Derived from Groupwise Comparisons

If we assume that all time series from the same class label are similar, we may consider the groupwise (or classwise) means as prototypical time series for their respective class. We define the *mean smoothed series* $\bar{\tilde{x}}_l$ for a class label l as:

$$\bar{\tilde{\mathbf{x}}}_l = \frac{\sum_{1 \le i \le n, l_i = l} \tilde{\mathbf{x}}_i}{\sum_{1 \le i \le n, l_i = l} 1}$$

The rationale for finding a filter is then that the distance of any series $\mathbf{x}$ with label l to the mean series $\bar{\tilde{\mathbf{x}}}_l$ of its own class should be small, but distances to the mean series $\bar{\tilde{\mathbf{x}}}_k$ of other classes, $k \neq l$, should be large. This time we enforce a unit length constraint on the filter to allow for filter types whose coefficients sum up to zero.

$$\text{max. } f(\alpha) = \frac{\sum_{l,k \in L} \|\bar{\tilde{\mathbf{x}}}_l - \bar{\tilde{\mathbf{x}}}_k\|^2}{\sum_{1 \le i \le N} \|\tilde{\mathbf{x}}_i - \bar{\tilde{\mathbf{x}}}_{l_i}\|^2} \qquad \text{s.t. } \|\alpha\| = 1 \tag{5}$$

This objective function has to be maximized (subject to the unit length constraint): The nominator has to be maximized (distances between class means), the denominator has to be minimized (distance of individual series to its own class mean).

Proposition 2. *The optimal filter $\alpha = (\alpha_{-m}, \ldots, \alpha_0, \ldots, \alpha_m) \in \mathbb{R}^{2m+1}$ maximizing (5) is the eigenvector with the largest eigenvalue of the matrix $Q^{-1}P \in \mathbb{R}^{2m+1 \times 2m+1}$, where*

$$P = \sum_{l,k \in L} M_{\bar{\mathbf{x}}_l, \bar{\mathbf{y}}_k}, \quad Q = \sum_{1 \le i \le n} M_{\mathbf{x}_i, \bar{\mathbf{x}}_{l_i}}, \quad M_{\mathbf{x},\mathbf{y}} = \sum_{l=1}^{n} \pi_l(\mathbf{x} - \mathbf{y}) \pi_l(\mathbf{x} - \mathbf{y})^\top$$

and $\pi_l(\mathbf{z}) = (z_{l-m}, \ldots, z_{l+m}) \in \mathbb{R}^{2m+1}$.

[Again, the proof is omitted due to lack of space.]

3.3 Computational Complexity

Lock-step measures such as Euclidean distance are computationally inexpensive ($O(n)$ for the comparison of two series of length n). The nature of most elastic measures, such as dynamic time warping, calls for a quadratic complexity $O(n^2)$. The advantage of the filter approaches is that we can spend some computational effort beforehand to determine the optimal filter and then stick to a lock-step measure, taking advantage of the low linear complexity $O(n)$ when comparing time series. The computation of the optimal filter involves the pairwise combination of time series in both corollaries. While proposition 1 requires to combine all series with the same or different label ($O(N^2)$), with proposition 2 we combine only mean series for each class label which drastically reduces the computational effort ($O(|L|^2)$ with $|L|$ being much smaller than N).

4 Experimental Evaluation

Time series similarity measures are typically evaluated against each other by examining their performance in a one-nearest-neighbor (1-NN) classification task (cf. [8,9]). A dataset is subdivided into train and test data and for each series from the test dataset, the closest series from the training set is sought. The class label from the closest match is then used for class prediction. The accuracy (or error rate) reports the number of correct (or incorrect) predictions. We report cross-validated results in Table 1 and (in contrast to the typical cross-validation for classifiers) use only one fold for training and $k-1$ for testing. As ED and DTW are the most prominent representatives of lock-step and elastic measures we compare them to three types of filters: (a) filter constraint "**s**um=1" (ED-FS) as defined by Proposition 1, (b) **f**ilter constraint "**n**orm=1" (ED-FN) as defined by Proposition 2, and (c) a standard **G**aussian **f**ilter (ED-FG). The filter is *always determined using the training data only*, then the filter is applied to the test and training data and the 1-NN classifier is carried out with Euclidean distance.

Filter Width. Both proposals for determining an optimized smoothing filter require the filter size m to be given a priori. We ran experiments with varying the filter size to observe the effect on the classification rate. There are differences in the performance, but the variation is quite limited. For all experiments in this paper, however, we have consistently chosen $2m$ to be 10% of the time series length but not larger than $m = 7$.

4.1 Artificial Data

As a sanity check, we revisit the two datasets from section 2 that are particularly difficult for either DTW or ED: In Fig. 1(a) the series from two groups differ in the orientation of a single peak (up vs down), but the exact location of the peak varies. A small amount of warping compensates for the offset in the peak position, but for ED two series from the same group are (most of the time) almost as similar as two series from different groups. Smoothing the series blurs the local peak, which makes it easier to detect the similarity between peaks of the same direction even if no warping is possible. As we can see from Table 1 the filter approaches perform (equally) well (about 94% accuracy).

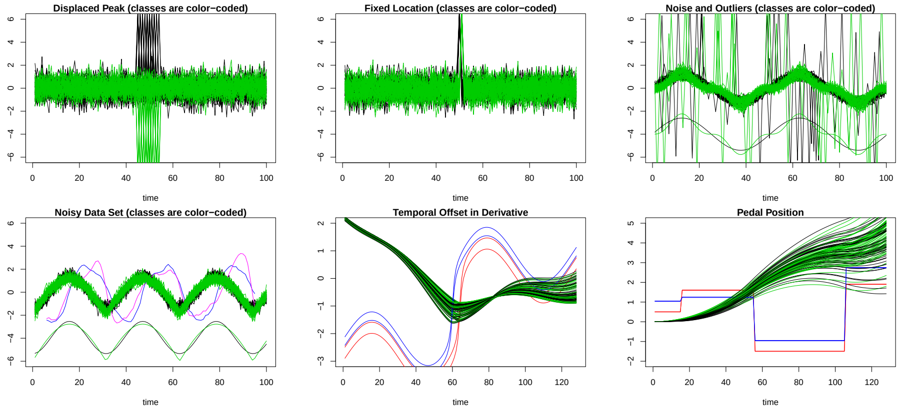

Figure 1. Some data sets used. Top row from left to right (a)-(c), bottom row (d)-(f). All datasets consist of 100 time series. Class information is color-coded (green/black).

The second example is shown in Fig. 1(b): the peak orientation is always identical (up), but the exact location of the peak makes the difference between both classes. As expected DTW is not able to perceive a difference between both classes, but for ED this is a very simple task (100% accuracy). The proposed approaches manage to adopt automatically to this situation, FS and FN perform very well (close to 100%, cf. Table 1), but apparently the standard Gaussian filter cannot help here, its performance is close to the poor DTW performance.

Response to Noise. A second set of examples is shown in Fig. 1(c-d), where the dataset (c) consists of sinusoidal curves plus Gaussian noise and a few outliers. One example from each class (without noise and outliers) is shown near the bottom (being not part of the dataset). As we can see from Table 1, DTW performance drops below 60%; ED performs much better, but all the smoothing approaches outperform ED. The second example (Fig. 1(d)) consists of noisy series, which do not contain any outliers. Again, one example from each class without any noise is shown at the bottom. No warping is necessary, but the noise seems to prevent DTW from performing as well as ED. The most prominent differences of the two classes are the rounded versus angular minima. When applying a smoothing filter we risk to smear out the angular edge and to loose an important feature for class discrimination. But actually the FN and FS filter manage to keep an accuracy close to 100%. The chosen filter does not only denoise but delivers series where the peak positions are displaced for series from different classes (examples of smoothed series shown in blue and pink).

4.2 Results for Data from the UCR Time Series Repository

We also report results on datasets from the UCR time series repository [6] in Table 1. All datasets were used as they were provided, no changes were made to them. We are interested in how much of the performance increase of elastic

Table 1. Mean accuracy and standard deviation of cross-validated 1-NN-classifier (no. of folds in column #, for UCR data the same number of folds was used as in [9]). Euclidean distance (ED) with: Gaussian filter (ED-FG), filter obtained from sum constraint (ED-FS), filter obtained from norm constraint (ED-FN).

<table>
<tr><th></th><th>#</th><th>ED</th><th>DTW</th><th>ED-FG</th><th>ED-FN</th><th>ED-FS</th></tr>
<tr><td>Fig. 1(a)</td><td>11</td><td>0.68|0.06</td><td>1.00|0.00</td><td>0.93|0.04</td><td>0.94|0.03</td><td>0.94|0.03</td></tr>
<tr><td>Fig. 1(b)</td><td>11</td><td>1.00|0.00</td><td>0.50|0.05</td><td>0.56|0.04</td><td>0.99|0.01</td><td>0.99|0.01</td></tr>
<tr><td>Fig. 1(c)</td><td>11</td><td>0.72|0.15</td><td>0.58|0.07</td><td>0.84|0.08</td><td>0.84|0.04</td><td>0.81|0.08</td></tr>
<tr><td>Fig. 1(d)</td><td>11</td><td>0.90|0.06</td><td>0.84|0.08</td><td>0.96|0.02</td><td>0.98|0.02</td><td>0.98|0.02</td></tr>
<tr><td>Fig. 1(e)</td><td>11</td><td>0.78|0.09</td><td>0.62|0.09</td><td>0.78|0.08</td><td>0.98|0.05</td><td>0.95|0.08</td></tr>
<tr><td>Fig. 1(f)</td><td>11</td><td>0.78|0.08</td><td>0.61|0.07</td><td>0.78|0.08</td><td>0.97|0.04</td><td>0.97|0.04</td></tr>
<tr><td>ECG200</td><td>5</td><td>0.85|0.03</td><td>0.77|0.03</td><td>0.83|0.03</td><td>0.83|0.03</td><td>0.82|0.04</td></tr>
<tr><td>ECGFiveDays</td><td>32</td><td>0.86|0.04</td><td>0.80|0.04</td><td>0.92|0.03</td><td>0.90|0.06</td><td>0.90|0.04</td></tr>
<tr><td>FISH</td><td>5</td><td>0.73|0.03</td><td>0.69|0.03</td><td>0.72|0.03</td><td>0.72|0.04</td><td>0.68|0.04</td></tr>
<tr><td>GunPoint</td><td>5</td><td>0.87|0.03</td><td>0.85|0.03</td><td>0.86|0.03</td><td>0.87|0.04</td><td>0.88|0.04</td></tr>
<tr><td>OliveOil</td><td>2</td><td>0.86|0.05</td><td>0.86|0.04</td><td>0.85|0.05</td><td>0.86|0.05</td><td>0.84|0.05</td></tr>
<tr><td>Beef</td><td>2</td><td>0.46|0.07</td><td>0.46|0.07</td><td>0.45|0.07</td><td>0.44|0.07</td><td>0.45|0.07</td></tr>
<tr><td>Adiac</td><td>5</td><td>0.54|0.02</td><td>0.54|0.02</td><td>0.60|0.02</td><td>0.42|0.16</td><td>0.52|0.02</td></tr>
<tr><td>Coffee</td><td>2</td><td>0.82|0.07</td><td>0.84|0.08</td><td>0.80|0.07</td><td>0.97|0.03</td><td>0.86|0.09</td></tr>
<tr><td>50words</td><td>5</td><td>0.59|0.02</td><td>0.62|0.02</td><td>0.60|0.02</td><td>0.42|0.16</td><td>0.53|0.02</td></tr>
<tr><td>SwedishLeaf</td><td>5</td><td>0.70|0.02</td><td>0.75|0.01</td><td>0.72|0.02</td><td>0.70|0.01</td><td>0.69|0.02</td></tr>
<tr><td>CBF</td><td>12</td><td>0.94|0.02</td><td>0.99|0.00</td><td>0.98|0.01</td><td>0.99|0.01</td><td>0.99|0.01</td></tr>
<tr><td>OSULeaf</td><td>5</td><td>0.52|0.03</td><td>0.58|0.04</td><td>0.53|0.02</td><td>0.48|0.05</td><td>0.51|0.03</td></tr>
<tr><td>FaceFour</td><td>5</td><td>0.80|0.05</td><td>0.87|0.04</td><td>0.81|0.05</td><td>0.82|0.05</td><td>0.82|0.05</td></tr>
<tr><td>Lighting7</td><td>2</td><td>0.61|0.05</td><td>0.71|0.04</td><td>0.65|0.04</td><td>0.68|0.04</td><td>0.68|0.05</td></tr>
<tr><td>Lighting2</td><td>5</td><td>0.67|0.05</td><td>0.78|0.06</td><td>0.72|0.05</td><td>0.68|0.08</td><td>0.72|0.05</td></tr>
<tr><td>synth.control</td><td>5</td><td>0.86|0.02</td><td>0.99|0.00</td><td>0.99|0.00</td><td>0.96|0.01</td><td>0.96|0.00</td></tr>
<tr><td>Trace</td><td>5</td><td>0.64|0.03</td><td>0.98|0.02</td><td>0.59|0.04</td><td>0.63|0.04</td><td>0.63|0.04</td></tr>
</table>

measures can be achieved by filtering (when sticking to a lock-step measure). By scanning through the Table we can see that quite often a substantial fraction of the performance increase obtained by switching from ED to DTW is also obtained when switching to a filter approach – in various cases the achieved accuracy is identical. To our surprise, we can also identify cases where filtering actually outperforms DTW (e.g. Coffee and Adiac), which is a remarkable result.

4.3 Accumulated Signals

Finally, we consider a situation where the FS/FN filter approaches perform particularly well. Many real series record a physical property reacting to some external input, for example, the temperature during a heating period, the rise and fall of the water level as inlets or outlets are opened, the distance covered when driving a car, etc. What is actually changing is some input variable (power of heating element, valve position, throttle control), but instead of capturing this parameter directly, some other accumulated physical property is measured (temperature, water level, covered distance).

The following examples are artificially generated but were created to reproduce a real case.[1] In the dataset of Fig. 1(e) all the series appear very similar and a discrimination between the classes seems close to impossible. The series correspond to an accumulation of some physical property, whose derivative is very similar for all examples, but differs in the temporal location of a steep increase near $t = 60$ (two examples from each class in red/blue). As the integrated values are actually measured, this difference is hardly recognized in the original series. But both filters, FS and FN, manage to transform the raw data such that the accuracy increases dramatically (cf. Table 1).

[1] Only the artificial data can be shared: `public.ostfalia.de/~hoeppnef/tsfilter`

For the second example in Fig. 1(f) we have a similar situation, the time series look very similar across the two classes. One may think of a gas pedal position as the actually controlled input (two examples from each class in red/blue), which influences speed, but only the mileage is measured. Again, both filters manage to identify filters that separate both classes very well. In both cases, a Gaussian filter does not help (data is not noisy) and the elastic DTW performs worst.

5 Conclusions

When seeking for the best similarity measure for a specific application, among equally well performing solutions we should prefer the simplest approach (Occam's Razor). Filtering time series and measuring their Euclidean distance is one of the most simple things one can possibly think of, however, this option has not received much attention in the literature. Two approaches have been proposed to derive such a filter from training data and the experimental results have shown that they turn Euclidean distance into a much more versatile tool, as it can adapt to specific properties of the time series. For various datasets a substantial increase in the performance has been observed and for a specific class of problems (discrimination of series that represent accumulating physical properties) this approach outperforms elastic measures. Together with the simplicity of the Euclidean measure, which has a computational advantage over complex elastic measures, this approach is a worthwhile alternative to existing measures. How to identify an optimal filter in combination with elastic measures remains an open question for future work.

References

1. 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. MITP, 1996.
2. L. Chen and R. Ng. On the marriage of Lp-norms and edit distance. In *Int. Conf. on Very Large Databases*, pages 792–803, 2004.
3. L. Chen, M. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. *Proc. ACM SIGMOD*, pages 491–502, 2005.
4. F. Höppner. Less is more: Similarity of time series under linear transformations. In *Proc. SIAM Int. Conf. Data Mining (SDM)*, pages 560–568, 2014.
5. Y.-S. Jeong, M. K. Jeong, and O. a. Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, Sept. 2011.
6. E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR Time Series Classification/Clustering Homepage, 2011.
7. W. T. Liao. Clustering of time series data – a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.
8. J. Serrà and J. L. Arcos. An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems*, 67:305–314, Sept. 2014.
9. 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.