# shapeDTW: shape Dynamic Time Warping

Naël Farhan nael.farhan@ensae.fr
Adèle Moreau adele.moreau@ensae.fr

## 1 Introduction and contributions

In the context of time series analysis, Dynamic Time Warping (DTW) stands out as a valuable algorithm for measuring similarity between temporal sequences, exhibiting proficiency in applications such as speech recognition, video, or time series alignments. Operating as a point-to-point matching method under specific boundary and temporal consistency constraints, DTW endeavors to uncover a globally optimal alignment path, by allowing local shifts, contractions, and stretches in temporal sequences. Despite its ability to yield a global optimal solution, DTW's drawback lies in its potential to produce matches lacking local sensibility. This limitation comes from the non consideration of point-wise local structural information, indeed DTW's matching relies on checking the similarity of two points thanks to their Euclidean distance, i.e. their coordinates values. Thus, two temporal points with vastly dissimilar local structures may be erroneously paired by DTW, resulting in alignment with a poor semantic meaning.

In the paper shapeDTW: shape Dynamic Time Warping, the authors introduce the shapeDTW algorithm, a new method for measuring similarity between temporal sequences that addresses the issues of DTW and enhances its capabilities. The key concept is to take into account the point-wise local structure of the time series during the matching process. Behind this, the idea is to map points that have similar neighborhood structures, because they are more likely to be really paired. There are two major steps in the shapeDTW algorithm. First, starting from our raw temporal sequences, we need to encode the local structure of each temporal point. This done by some shape descriptor, which captures the local subsequence structural information around the concerned point. Then, the time series being converted into a sequence of descriptors, we align these two sequences using DTW. Two major advantages arises from this method. ShapeDTW gets lower alignment errors in comparison to DTW and its variants (derivative DTW and weighted DTW), and on the task of nearest neighbor classification, the shapeDTW distance substantially outperforms DTW with better performances on 64 out of 84 UCR time series datasets. At the time of the article (2016), shapeDTW stands out as the first distance measure that notably surpasses DTW within the nearest neighbor classifier scheme. Furthermore, shapeDTW offers a broadly applicable alignment framework, allowing users to create tailored shape descriptors that suit the characteristics of their domain data.

The article provided a publicly accessible code at https://github.com/jiapingz/shapeDTW. This original code being in Matlab, we also used another public code that implements the shapeDTW algorithm in Python at https://github.com/MikolajSzafraniecUPDS/shapedtw-python. The shapeDTW is already implemented in this package, but none of the experiments were reproduced. Thus, we first decided to conduct the simulated sequence-pair alignment experiment from the paper. Then, we wanted to compare the performance of shapeDTW over shapeDTW in a task of speech recognition. We contributed equally to the article's summary and each conducted one of the experiments.

# 2 Method

## 2.1 Dynamic Time Warping

First, let's reintroduce the DTW algorithm, since we will use DTW for the alignment step of the shapeDTW implementation. As stated before, DTW aims at measuring the similarity between two time series. It is applicable for both univariate and multivariate sequences, however we will introduce the univariate case. Let $\mathcal{X}$ and $\mathcal{Y}$ be two time series of length $\mathcal{L}_{\mathcal{X}}$ and $\mathcal{L}_{\mathcal{Y}}$, $\mathcal{X} = (x_1, x_2, ..., x_{\mathcal{L}_{\mathcal{X}}})^T$, $\mathcal{Y} = (y_1, y_2, ..., y_{\mathcal{L}_{\mathcal{Y}}})^T$, and $\mathcal{D}(\mathcal{X}, \mathcal{Y}) \in \mathbb{R}^{\mathcal{L}_X \times \mathcal{L}_Y}$ a pairwise distance matrix between $\mathcal{X}$ and $\mathcal{Y}$. One common pairwise distance measure is the Euclidean distance such that $\mathcal{D}(\mathcal{X}, \mathcal{Y})_{i,j} = |x_i - y_j|$. We aim at finding two sequences of indices $\alpha$ and $\beta$ of the same length $l$, which match index $\alpha(i)$ in $\mathcal{X}$ to index $\beta(i)$ in $\mathcal{Y}$, in order to minimize the total cost along the matching path $\sum_{i=1}^{l} \mathcal{D}(\mathcal{X}, \mathcal{Y})_{\alpha(i),\beta(i)}$. The alignment path $(\alpha, \beta)$ should satisfy constraints of:

- **Continuity and Monotonicity** $(\alpha(i+1), \beta(i+1)) - (\alpha(i), \beta(i)) \in \{(1,0), (1,1), (0,1)\}$
- **Boundary** $\alpha(1) = \beta(1) = 1$, $\alpha(l) = \mathcal{L}_{\mathcal{X}}$, $\beta(l) = \mathcal{L}_{\mathcal{Y}}$

Given this alignment path $(\alpha, \beta)$, we define two warping matrices $\mathcal{W}^{\mathcal{X}} \in \{0,1\}^{l \times \mathcal{L}_{\mathcal{X}}}$ and $\mathcal{W}^{\mathcal{Y}} \in \{0,1\}^{l \times \mathcal{L}_{\mathcal{Y}}}$, such that $\mathcal{W}^{\mathcal{X}}(i, \alpha(i)) = 1$, otherwise $\mathcal{W}^{\mathcal{X}}(i,j) = 0$, and similarly $\mathcal{W}^{\mathcal{Y}}(i, \beta(i)) = 1$, otherwise $\mathcal{W}^{\mathcal{X}}(i,j) = 0$. Then, the total cost along the matching path $\sum_{i=1}^{l} \mathcal{D}(\mathcal{X}, \mathcal{Y})_{\alpha(i),\beta(i)}$ is equal to $\left\| \mathcal{W}^{\mathcal{X}}.\mathcal{X} - \mathcal{W}^{\mathcal{Y}}.\mathcal{Y} \right\|_1$. Searching for the optimal temporal matching can be reformulated under an optimization problem as follows:

$$\text{argmin}_{l, \mathcal{W}^{\mathcal{X}} \in \{0,1\}^{l \times \mathcal{L}_{\mathcal{X}}}, \mathcal{W}^{\mathcal{Y}} \in \{0,1\}^{l \times \mathcal{L}_{\mathcal{Y}}}} \left\| \mathcal{W}^{\mathcal{X}}.\mathcal{X} - \mathcal{W}^{\mathcal{Y}}.\mathcal{Y} \right\|_1$$

## 2.2 Shape Dynamic Time Warping

Recall that the shapeDTW algorithm is built on two major steps, first we encode the local structure of each temporal point thanks to some shape descriptors, then we align the temporal sequences of descriptors by DTW.

Given an univariate time series $\mathcal{T} = (t_1, t_2, ..., t_L)^T$ of length $L$, we begin by extracting subsequences $s_i$ of length $l$ for each temporal point $t_i$. These subsequences will help us to capture local structure information. The subsequence $s_i$ is centered on $t_i$, and we now have $\mathcal{S} = (s_1, s_2, ..., s_L)^T$, $s_i \in \mathbb{R}^l$. We need to fill both ends of the sequence $\mathcal{T}$ by $\lfloor l/2 \rfloor$ with duplicates of $t_1 (t_L)$ so that the subsequences sampled at the ends are well defined. Next, we encode shape descriptors to express subsequences. We want similar subsequences to have similar shape descriptors. The shape descriptor of $s_i$ would naturally capture local structure information around the temporal point $t_i$. We encode each subsequence $s_i$ by a shape descriptor $d_i \in \mathbb{R}^m$. Thus, $\mathcal{S}$ is converted to a sequence of shape descriptors of the same length $\mathbf{d} = (d_1, d_2, ..., d_L)^T$, $\mathbf{d} \in \mathbb{R}^{L \times m}$. To map a subsequence $s_i \in \mathbb{R}^l$ to a shape descriptor $d_i \in \mathbb{R}^m$, we design a mapping function $\mathcal{F}(.)$, such that $d_i = \mathcal{F}(s_i)$ and $\mathbf{d} = (\mathcal{F}(s_1), \mathcal{F}(s_2), ..., \mathcal{F}(s_L))^T$. We discuss in appendix A the possible form of the mapping function. Finally, we use the DTW to align two sequences of descriptors and transfer the warping path to the original univariate time series.

Given two univariate time series $\mathcal{X} = (x_1, x_2, ..., x_{\mathcal{L}_{\mathcal{X}}})^T$, $\mathcal{X} \in \mathbb{R}^{\mathcal{L}_{\mathcal{X}}}$ and $\mathcal{Y} = (y_1, y_2, ..., y_{\mathcal{L}_{\mathcal{Y}}})^T$, $\mathcal{Y} \in \mathbb{R}^{\mathcal{L}_{\mathcal{Y}}}$, let $\mathbf{d}^{\mathcal{X}} = (d_1^{\mathcal{X}}, d_2^{\mathcal{X}}, ..., d_{\mathcal{L}_{\mathcal{X}}}^{\mathcal{X}})^T$, $d_i^{\mathcal{X}} \in \mathbb{R}^m$, $\mathbf{d}^{\mathcal{X}} \in \mathbb{R}^{\mathcal{L}_{\mathcal{X}} \times m}$ and $\mathbf{d}^{\mathcal{Y}} = (d_1^{\mathcal{Y}}, d_2^{\mathcal{Y}}, ..., d_{\mathcal{L}_{\mathcal{Y}}}^{\mathcal{Y}})^T$, $d_i^{\mathcal{Y}} \in \mathbb{R}^m$, $\mathbf{d}^{\mathcal{Y}} \in \mathbb{R}^{\mathcal{L}_{\mathcal{Y}} \times m}$ be their sequences of shape descriptors. The alignment problem of shapeDTW is equivalent to solving the optimization problem:

$$\arg\min_{l,\tilde{\mathcal{W}}^{\mathcal{X}}\in\{0,1\}^{l\times\mathcal{L}_{\mathcal{X}}},\tilde{\mathcal{W}}^{\mathcal{Y}}\in\{0,1\}^{l\times\mathcal{L}_{\mathcal{Y}}}} \left\|\tilde{\mathcal{W}}^{\mathcal{X}}.\mathcal{X}-\tilde{\mathcal{W}}^{\mathcal{Y}}.\mathcal{Y}\right\|_{1,2}$$

Where $\tilde{\mathcal{W}}^{\mathcal{X}}$ and $\tilde{\mathcal{W}}^{\mathcal{Y}}$ are warping matrices of $\mathbf{d}^{\mathcal{X}}$ and $\mathbf{d}^{\mathcal{Y}}$, and $\|.\|_{1,2}$ is the ${}_1/{}_2$-norm of matrix.

The primary distinction between DTW and shapeDTW lies in their similarity measurement approach. While DTW gauges the similarity between $x_i$ and $y_j$ based on their Euclidean distance, shapeDTW employs the Euclidean distance between their shape descriptors, specifically $\left\|d_i^{\mathcal{X}}-d_j^{\mathcal{Y}}\right\|_2$, as the basis for similarity assessment.

# 3 Data

## 3.1 Simulated data

First, we want to look at how shapeDTW behaves on simulated data. To do so, we used some time series from the OSULeaf dataset which is made of 442 leaves outlines (each of length 427). Then, to simulate data in order to compute metrics, for a time series, we created an algorithm to create an aligned one : first, we apply a smooth scale point-wise to every point of the original series, and then for a proportion $\alpha$ of points we stretch that value 1, 2 or 3 times (uniform draw). Note that, if we take $\alpha = 10\%$ for instance, on average, the simulated series will be $2\alpha = 20\%$ longer. Therefore, we can create a time series which share similar shapes and scale from the original one and different length. In Figure 1, we plot some scaling vectors as well as an example of a simulated aligned series.

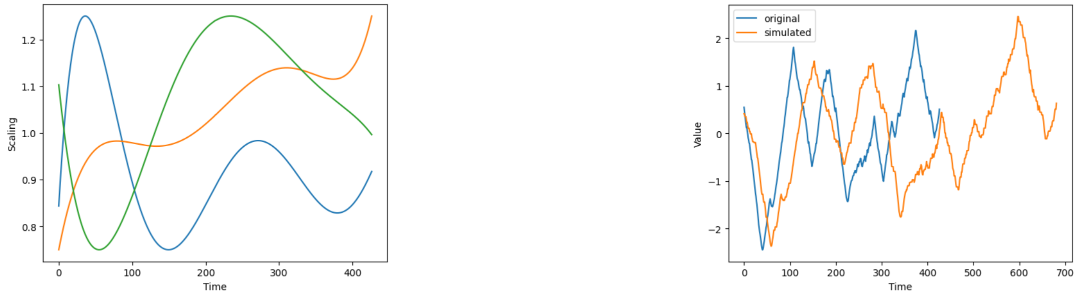
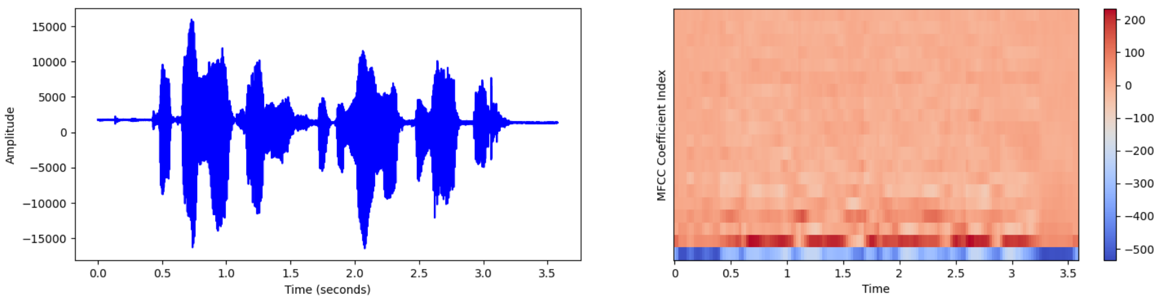


Figure 1: Examples of scaling vectors in $[0.75, 1.25]$ and a simulated aligned series

## 3.2 Surrey Audio-Visual Expressed Emotion database

The DTW being used a lot in the field of speech recognition, as a second experiment, we would like to compare the performance of DTW and shapeDTW for such a task. We work with the Surrey Audio-Visual Expressed Emotion (SAVEE) database. It was created as a foundational step for the development of an automatic emotion recognition system. This database comprises recordings from four native English male speakers (identified as DC, JE, JK, KL), encompassing 7 distinct emotions and totaling 480 British English utterances. The sentences were carefully selected from the standard TIMIT corpus, ensuring phonetic balance for each emotion. The Figure (a) below shows for instance an audio signal from the database where the speaker JE says "If people were more generous, they would be no need for welfare".



(a) Audio signal of the Sentence *d10* pronounced by Speaker JE

(b) MFCC Features from Sentence *d10* pronounced by Speaker JE

After an in-depth study of the data, we decided not to keep the KL speaker in our database. As the audio of the sentences he pronounces was very different from the other 3 speakers, the DTW and shapeDTW distances almost never manage to recognize the associated sentence. This seems to be due to poor word articulation. What's more, since calculation times were rather long, it made sense to reduce the size of the database.

# 4 Results

## 4.1 Experiment 1: Leaves outlines and simulated data

This first experiment was made to see how shapeDTW behaves against classic DTW when time series have many local features, such as leaves outlines. To highlight qualitative weaknesses that may occur with the use of DTW, we can look at the example down below, which shows two similar leaves and alignments results for DTW and shapeDTW. On two areas where there are important change in the time series, DTW struggles to correctly align two similar shapes.

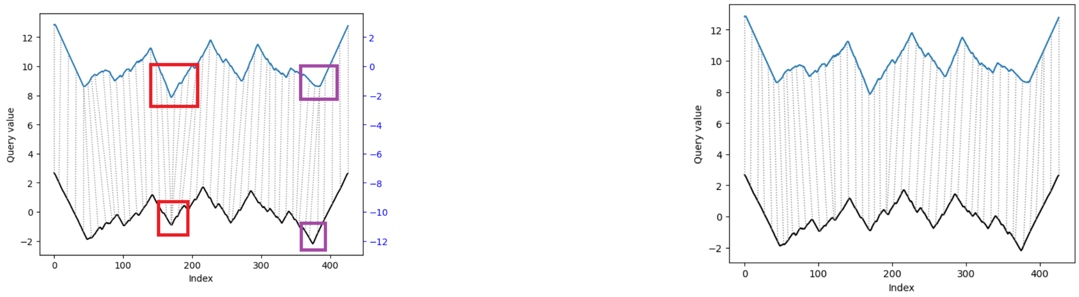


Figure 3: Alignment path for DTW on the left, and shapeDTW with a Derivative descriptor (subsequence length of 25) on the right

Furthermore, we introduce simulated aligned series to compute our metric, the Mean Absolute Deviation (Appendix B), between the path given by DTW/shapeDTW and the ground truth alignment based on our simulation. For some values of the stretching pourcentage $\alpha$, we compute the MAD over the 442 series (only one simulation per series), which gives the results in Table 1.

We notice that shapeDTW outperforms DTW well for most of the first values of $\alpha$ as it is able to well align corresponding shapes. However, when $\alpha$ gets higher, its performance gets worse than DTW, because more data points are added, this might be due to the fact that we keep the

| $\alpha$ (%) | 2.5 | 5 | 7.5 | 10 | 12.5 | 15 | 17.5 | 20 |
|---|---|---|---|---|---|---|---|---|
| $MAD_{\text{DTW}}$ | 1041 | 1310 | 1494 | 1645 | 1729 | 1946 | 2121 | 2156 |
| $MAD_{\text{shapeDTW}}$ | 586 | 968 | 1265 | 1487 | 1686 | 2020 | 2282 | 2690 |

Table 1: Mean Absolute Deviation for values of $\alpha$, shapeDTW has Derivative descriptor and $l = 25$

subsequence length constant and a higher one might yield better performance. Also, for a high stretching percentage, shapes between the two series can differ more, therefore using shapeDTW might become useless.

### 4.2 Experiment 2: Speech recognition

The aim of this experiment is to perform audio comparison thanks to DTW and shapeDTW. We want to compare a sentence spoken by one actor with all the sentences spoken by another person, then extract the most similar sentence using DTW or shapeDTW. This comparison is carried out for all sentences and for all actors, in order to obtain a rate of correctly and incorrectly recognized sentences. We expect a better performance of the shapeDTW thanks to its ability to take local neighborhood information into account.

The audio comparison process consists of two distinct stages. First, the MFCC (Mel-Frequency Cepstral Coefficient) calculation part which extracts features from the audio sample, providing information about the rate of change in spectral bands. This step is crucial as it helps in identifying the content of the audio sample and facilitates a unique identification. The MFCCs of an audio signal are a small set of features (usually about 10–20) which describe the overall shape of the spectral envelope. We use the librosa library to extract the MFCCs and an example can be found on Figure (b). Subsequently, the chosen distance metric (DTW or shapeDTW) is employed to compare two MFCC feature matrices, assessing their similarity. It lets us determine the similarity of two samples even in cases where the samples are not perfectly aligned or their speeds differ. For both distances, we use the step pattern "symmetric2", and shapeDTW uses a subsequence width of 30 and the Derivative shape descriptor.

| Original Speaker | Comparison Speakers | DTW Correct Matches | DTW Wrong Matches | DTW Success Rate | shapeDTW Correct Matches | shapeDTW Wrong Matches | shapeDTW Success Rate |
|---|---|---|---|---|---|---|---|
| DC | JE | 70 | 50 | 59.6% | 73 | 47 | 64.6% |
| | JK | 73 | 47 | | 82 | 38 | |
| JE | DC | 69 | 51 | 53.7% | 73 | 47 | 61.7% |
| | JK | 60 | 60 | | 75 | 45 | |
| JK | DC | 53 | 67 | 38.3% | 62 | 58 | 47.5% |
| | JE | 39 | 81 | | 52 | 68 | |

Table 2: Number of sentences correctly or wrongly matched for each speaker with DTW and shapeDTW

We can read Table 2 as follows: for each of the 120 sentences pronounced by DC, when comparing with the sentences pronounced by JE thanks to DTW, the most similar sentence is the right one in 70 cases over 120. The rate of success is the percentage of sentences well-classified for a speaker when comparing to the two others. We notice a higher success rate when using shapeDTW. For instance, the success rate when the original speaker is JE goes from 53.7% with DTW to 61.7% with shapeDTW. Finally, we demonstrate that incorporating local neighborhood information (shapeDTW) does benefit the speech recognition task.

# A Shape Descriptors

In this section, we introduce several shape descriptor mapping functions $\mathcal{F}(.)$, that inherently encode local shape information. We choose the length of subsequences $l$ as any positive integers ($l \geq 1$), which does not affect the definition of shape descriptors.

**Raw-Subsequence** No transformation is applied to $s_i$, i.e. $d_i = \mathcal{I}(s_i) = s_i$.

**Piecewise aggregate approximation (PAA)** The subsequence $s_i$ is split into $m$ equal-lenghted and disjoint intervals. For each interval we calculate the mean value of temporal points falling into it, and a vector of these mean values is our shape descriptor $d_i = PAA(s_i)$.

**Discrete Wavelet Transform (DWT)** DWT is applied to the whole subsequence $s_i$ and wavelet coefficients are bound into the form of vector, which we use as a shape descriptor $d_i = DWT(s_i)$.

**Slope** We split subsequence into $m$ disjoint intervals and fit a line according to points falling within each interval. By concatenating the slopes of the fitted lines from all intervals, we obtain a vector and use it as a shape descriptor $d_i = Slope(s_i)$, which is invariant to y-shift.

**Derivative** We use the first order derivative of given subsequence, which is invariant to y-shift.

**HOG1D** It is the Histogram of Oriented Gradients (HOG) for 1-D time series, it uses concatenated gradient histograms to represent shapes of temporal sequences. It is also invariant to y-shift.

**Compound descriptor** Instead of using only one shape descriptor, we can fuse two or more complementary shape descriptors, and bound them into a form of single vector $d = (d_A, \gamma d_B)$. We can use weights to compensate for differences in average descriptor values.

# B Alignment quality evaluation

The *mean absolute deviation* is used to compare the proximity between alignment paths. Given a reference sequence $\mathcal{X}$, a target sequence $\mathcal{Y}$ and two alignment paths $\alpha$, $\beta$ between them, the *mean absolute deviation* between $\alpha$ and $\beta$ is calculated by $\delta(\alpha, \beta) = A(\alpha, \beta)/\mathcal{L}_{\mathcal{X}}$, where $A(\alpha, \beta)$ is the area between $\alpha$ and $\beta$ and $\mathcal{L}_{\mathcal{X}}$ is the length of the reference sequence $\mathcal{X}$.

In practice, to compute the area, we count the number of cells which fall between the two alignments.