Time-series averaging using constrained dynamic time warping with tolerance

Marion Morel, Catherine Achard, Richard Kulpa, Séverine Dubuisson

Time-series Averaging Using Constrained Dynamic Time Warping with Tolerance

Marion Morel[a,b], Catherine Achard[a], Richard Kulpa[b], Séverine Dubuisson[a]

[a]*Sorbonne Universités, UPMC Univ Paris 06, CNRS, UMR 7222, ISIR, F-75005, Paris, France*
[b]*M2S laboratory, Université Rennes 2, ENS Rennes, Avenue Robert Schuman, 35170 Bruz, France*

Abstract

In this paper, we propose an innovative averaging of a set of time-series based on the Dynamic Time Warping (DTW). The DTW is widely used in data mining since it provides not only a similarity measure, but also a temporal alignment of time-series. However, its use is often restricted to the case of a pair of signals. In this paper, we propose to extend its application to a set of signals by providing an average time-series that opens a wide range of applications in data mining process. Starting with an existing well-established method called DBA (for DTW Barycenter Averaging), this paper points out its limitations and suggests an alternative based on a Constrained Dynamic Time Warping. Secondly, an innovative tolerance is added to take into account the admissible variability around the average signal. This new modeling of time-series is evaluated on a classification task applied on several datasets and results show that it outperforms state of the art methods.

Keywords: Time-series averaging, Dynamic Time Warping, Local constraints, Constrained DTW Barycenter Averaging, Time-series Classification

1. Introduction

Time-series are studied in many fields and used for instance to analyze motion [10], handwriting [2], biological systems [23] and audio (speech or music) [11] signals. They differ from classical data because they are time-ordered. As a matter of fact, time-series' processing has become an important challenge for many research areas. One of the main goals is to derive a single model from a set of signals corresponding to several instances of the same physical process. This model can then be used for pattern recognition or to evaluate the differences between time-series. For example, one can measure the quality of a surgical gesture by comparing it to a set of correct ones [24]. Contrary to classical data that can be represented very simply by their mean and covariance matrix, time-series are much more difficult to process because they can have different lengths, requiring their temporal alignment as a pre-processing step.

A well-known tool for time-series' modelling is the Hidden Markov Model (HMM) [17] in which data are modeled using a Markov chain with unobserved states. Each state generates observed values according to a probability distribution that can be, for example, a multivariate Gaussian distribution for continuous values. Baum-Welch method [17] is

used to learn HMM parameters from observed unaligned time-series and the forward-backward procedure estimates the probability that an observed sequence has been generated by a HMM with a known (or learnt) set of parameters. HMM are powerful tools for solving simple time-series' recognition problems [1, 22, 5]. They however cannot evaluate the local quality of time-series, representing a gesture for example, since they only provide global probabilities to go from a state (a posture) to another one. It also implies that the alignment of a new sequence with the Markov chain is done only based on the timing between these states that does not allow a fine analysis of signals. Other models like Time delay neural networks [21] or finite state machines [7] have the same drawbacks.

For these reasons, some works have been developed based on Dynamic Time Warping (DTW). The DTW does not only provide a similarity measure between two time-series, but also a nonlinear alignment path between them. Several authors developed DTW based tools to average time-series [16, 14, 15, 13, 20]. However, these works face several limitations that will be discussed in Section 2.2. In this article, we propose a new DTW based averaging method that solves many of the limitations of the classicial DTW. Moreover, in addition to the mean value, we introduce a tolerance at each time step to model time-series' variation around the average. To the best of our knowledge, this is the first time that tolerances are introduced into DTW to model time-series. This new model of temporal data is introduced in Section 3. Results on a time-series' classification task show our approach outperforms state-of-the-art and are presented in Section 4.

2. Related Works

Before reviewing the different DTW-based methods for averaging time-series, we first give a brief overview of Dynamic Time Warping.

2.1. Dynamic Time Warping

DTW was first introduced in the 1970's for audio analysis [19, 18] before being used for general time-series analysis. DTW provides a non-linear alignment optimal path between two time-series by minimizing the cumulative distance between the signals $(x(i))_{1\leq i\leq M}$ and $(y(j))_{1\leq j\leq N}$ (time-series) to be aligned. It is a relatively straightforward process that first estimates the distance map $\mathbf{d}$ between the signals with elements $d_{i,j}$ given by:

$$d_{i,j} = (x(i) - y(j))^2 \tag{1}$$

A cumulative distance map $\mathbf{D}$ with elements $D_{i,j}$ is then computed from the distance map $\mathbf{d}$. $D_{i,j}$ represents the minimal accumulated distance to reach the point (i, j) starting from the origin $(1, 1)$. It is given by:

$$D_{i,j} = d_{i,j} + \min \begin{cases} D_{i,j-1} \\ D_{i-1,j} \\ D_{i-1,j-1} \end{cases} \quad i = 2, ..., M \quad j = 2, ..., N \tag{2}$$

considering the following initial conditions:

$$D_{1,1} = d_{1,1} \tag{3}$$

$$D_{1,j} = \sum_{p=1}^{j} d_{1,p} \quad j = 1, ..., N \tag{4}$$

$$D_{i,1} = \sum_{q=1}^{i} d_{q,1} \quad i = 1, ..., M \tag{5}$$

This cumulative distance is used to define a warping path between signals $(x(i))_{1 \leq i \leq M}$ and $(y(j))_{1 \leq j \leq N}$, noted ϕ_{xy}:

$$\phi_{xy}: \left\{ \begin{array}{ccc} [\![1; K]\!] & \longrightarrow & [\![1; M]\!] \times [\![1; N]\!] \\ k & \longmapsto & \phi_{xy}(k) = (\phi^x_{xy}(k), \phi^y_{xy}(k)) \end{array} \right. \tag{6}$$

ϕ_{xy} verifies several constraints:

- the monotonicity constraint guarantees the time ordering.
- the boundary constraints: $\phi_{xy}(1) = (1, 1)$ and $\phi_{xy}(K) = (M, N)$.
- the step size conditions: $0 \leq \phi^x_{xy}(k) - \phi^x_{xy}(k-1) \leq 1$ and $0 \leq \phi^y_{xy}(k) - \phi^y_{xy}(k-1) \leq 1, \quad \forall k \in 2...K$, see Equation 2.

This path minimizes the final cumulative distance. Its length K depends on the signals to be aligned and is determined during the DTW process.

Figure 1 presents two examples of DTW alignments between two signals with Figures 1a and 1b showing the alignments between signals and Figures 1c and 1d showing the cumulative distance maps $\mathbf{D}$ and the warping paths in green for these two examples.

DTW provides an optimal alignment of two signals. Let us now focus on the process of averaging a set of signals to obtain an average signal.

2.2. Time-series averaging

Let us first consider the simple case of two signals to average.

2.2.1. Averaging of two signals

In Section 2.1, we have introduced a method to align two time-series. The warping path ϕ_{xy} of length K contains index correspondences that align both signals $x(i)$ and $y(j)$ in a nonlinear way. This allows us to create two new aligned signals $x_K(k)$ and $y_K(k)$ with the same length K so that:

$$x_K(k) = x(\phi^x_{xy}(k)) \quad k = 1, ..., K \tag{7}$$

$$y_K(k) = y(\phi^y_{xy}(k)) \quad k = 1, ..., K \tag{8}$$

where $\phi^x_{xy}(k)$ and $\phi^y_{xy}(k)$ are defined in equation 6.

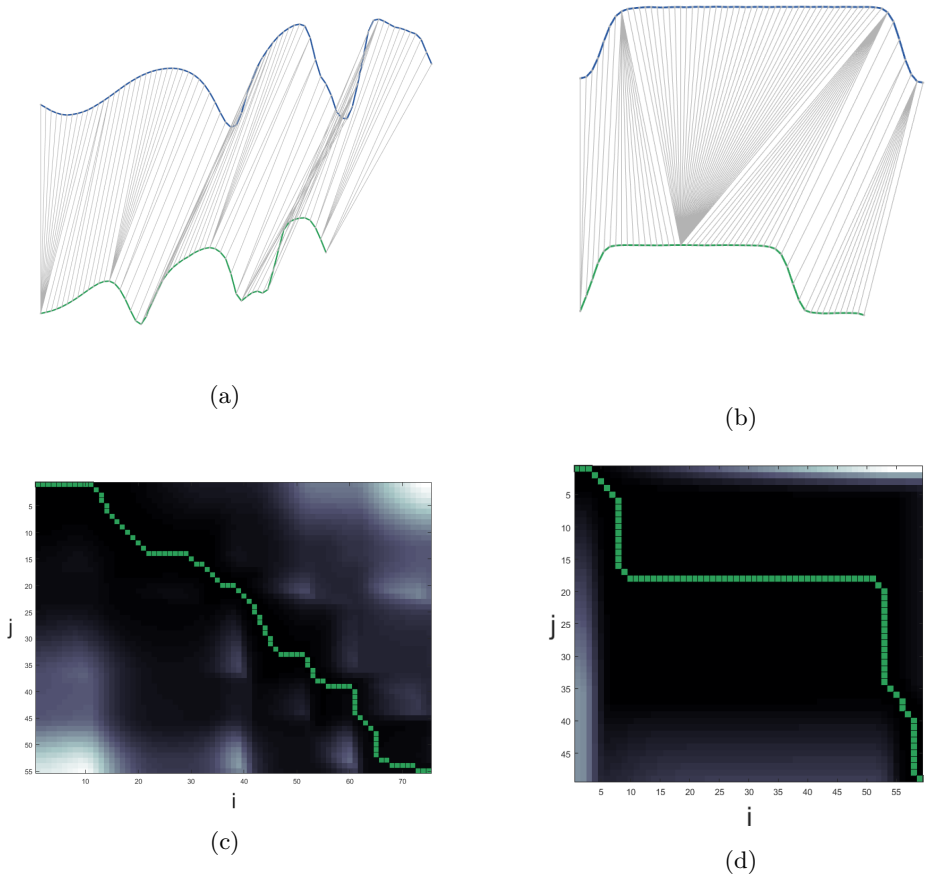
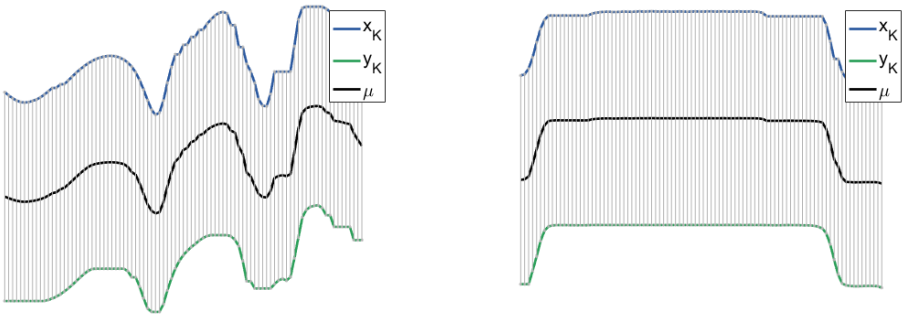


Figure 1: Illustration of the DTW process for the alignment of two pairs of signals (one per column). First row: the matching between points of two pairs of signals (in blue and green) is symbolized by grey lines. Second row: superimposition of the warping path on the cumulative distances matrix **D**. White areas of **D** correspond to largest cumulative distances whereas dark ones correspond to smallest ones. The second case of alignment ((b) and (d)) is an example of pathological path: as the green signal has a higher amplitude than the blue one, all high-amplitude values of the green signal are linked to only one value of the blue one (actually the higher one). This is characterized by the horizontal or vertical segments in the warping path (d).

The mean of signals $x(i)$ and $y(j)$ is then estimated by averaging signals $x_K(k)$ and $y_K(k)$ at each time step:

$$\mu(k) = \frac{x_K(k) + y_K(k)}{2} \quad k = 1, ..., K \tag{9}$$

By definition, the length K of the average signal $\mu(k)$ is such that $K \geq \max(M, N)$. Figure 2 illustrates this average signal on the two examples of Figure 1, based on the warping path ϕ_{xy}. $x_K(k)$ and $y_K(k)$ (respectively in blue and green) result in the average signal $\mu(k)$ (in black). Note that the length of $\mu(k)$ is $K = 91$ which is higher than the lengths of $x(i)$ and $y(j)$ (75 and 55 respectively) for the first example and $K = 92$ is higher than 59 and 49 for the second time-series.



(a) (b)

Figure 2: Illustration of the averaging method for the two pairs of signals in (a) and (b). $x_K(k)$ (in blue) and $y_K(k)$ (in green) with the same length K result from the resampling of signals $x(i)$ and $y(j)$ relatively to ϕ_{xy}. Their resulting average signal is $\mu(k)$ (in black).

2.2.2. Extension to the averaging of more than two time-series

DTW is widely used in signal processing but only for the alignment of two signals. It thus cannot directly extract an average signal from a set of time-series. Several authors have proposed solutions to tackle this issue.

The most intuitive process is the nonlinear alignment and averaging filters (NLAAF [6]): from a set of N time-series $\{x_1(k), \ldots, x_N(k)\}$, the authors propose to recursively align pairs of signals using DTW until the whole set is aligned. First, $x_1(k)$ and $x_2(k)$ are aligned and their average signal $z_1(k)$ is computed as explained in Section 2.2.1. Then the alignment and averaging of $x_3(k)$ and $z_1(k)$ gives $z_2(k)$ and so on, until every signals have been processed. One can also consider a weight during the averaging process so that every time-series has the same influence on the average. The main drawback of these kinds of methods is the succession of averaging. As illustrated before, an average signal is always longer than the two original signals: multiple averaging make

the resulting average signal much longer than the signals it represents.

The Prioritized Shape Averaging (PSA), an extension of NLAAF introduced by Niennattrakul *et al.* [13], orders the signals according to their shape similarity before averaging them. Considering a time-series dataset, the most similar signals are first detected, then aligned and averaged and so on until there is one final average time-series. Here again, the average signal becomes very long with regards to the number of time-series in the dataset.

A more global approach called DTW Barycenter Averaging has been introduced by Petitjean *et al.* in 2011 [16]. We give a short overview of this method in the next section.

2.2.3. DTW Barycenter Averaging (DBA)

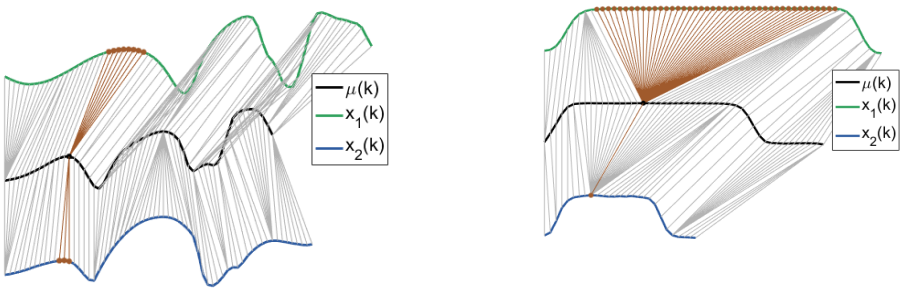
Authors in [16] proposed a fast algorithm that insures that the average signal will have a reasonable length. The main steps of the algorithms are given below:

1. Randomly choose a signal $x_0(k)$ from the dataset to initialize the average signal: $\mu(k) = x_0(k), \quad k = 1, ..., M_0$ where M_0 is the length of $x_0(k)$.
2. Iterate IT times the following steps:
   (a) Align all signals $x_l(k)$ on $\mu(k)$ and compute warping paths $\phi_{\mu x_l}$.
   (b) Update every point of the average signal $\mu(k)$ as the barycenter of points associated to it during step 2(a).

This process is described in Algorithm 1 and illustrated in Figures 3a and 3b where both signals $x_1(k)$ and $x_2(k)$ in green and blue respectively are simultaneously aligned on $\mu(k)$ in black.

Algorithm 1 $DBA : averagingDTW$

Require: $x_0(k)$ of length M_0, $(x_l(k))_{l=1...L}$ of lengths M_l, IT
  $K = M_0$, $\mu(k) \leftarrow x_0(k), \quad k = 1, ..., K$
  for $it \in 1...IT$ **do**
    $assocTab[k] = \emptyset, k = 1...K$
    for $l \in 1...L$ **do**
      $\phi_{\mu x_l} \leftarrow DTW(\mu, x_l)$
      $p \leftarrow length(\phi_{\mu x_l})$
      while $p \geq 1$ **do**
        $(k, n) \leftarrow \phi_{\mu x_l}(p)$
        $assocTab[k] \leftarrow assocTab[k] \cup \{x_l(n)\}$
        $p \leftarrow p - 1$
      end while
    end for
    for $k \in 1...K$ **do**
      $\mu(k) \leftarrow mean(assocTab[k])$
    end for
  end for
  return $\mu(k), \quad k = 1, ..., K$



(a) (b)

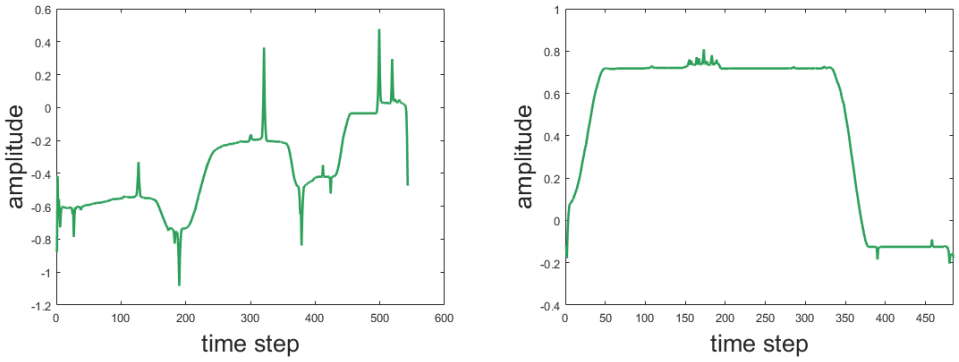
Figure 3: One iteration of DBA for 2 different sets of signals. The black signal is the average one $\mu(k)$ resulting from the previous iteration. Both blue and green signals are aligned on the black one. Each point of $\mu(k)$ is updated as the average of the points aligned on it (for example the red points for the corresponding time step).

One of the main limitations of DTW and DBA, occurs when the path "stagnates" during several indexes (one point of $x(k)$ is matched to several points of $y(k)$). This is what we call *pathological paths*, as the example of alignment of two signals with different amplitudes in Figure 1d. To solve this problem, a very simple idea is to first normalize the signals before aligning them. However, the zero-mean normalisation is not adequate when signals have different lengths and the normalization by the extrema is not robust to noise. Moreover, in case of more complex signals than the one in Figure 1b, the normalization will not be effective for all time steps. For example, if a signal has multiple local extrema, the normalization may be effective for the first extrema, but not for the others.

Another approach consists in considering both distance and derivative distance to align signals [9, 3]. Unfortunately, the alignment of the derivative is noise sensitive.

Since the DBA algorithm is based on DTW, it is also concerned by this alignment problem. Figure 4 illustrates the result of DBA averaging after $IT = 4$ iterations for the 2 sets of signals from which were extracted examples of Figure 1. The first set contains 319 time-series and the second one contains 101 time-series. Note that the high amplitudes are summarized in a single pathological point leading to an average signal whose shape is very different from original signal ones.

To avoid this problem, we developed a new algorithm, called CDBA for Constrained DTW Barycenter Averaging with several contributions. First, we propose to generalize the use of local constraints in DTW to avoid pathological paths. Second, we combine the two well-established concepts of DBA and CDTW to improve the time-series averaging. Finally, we add to the DBA averaging method a tolerance that takes into account the variability among the signals in the dataset. This innovative set of combined contribu-



(a) (b)

Figure 4: Illustration of the bad average signal obtained with DBA method with only 4 iterations on the 2 sets of signals from which were extracted examples of Figure 1.

tions provides a more accurate representation of signals than those provided by a simple averaging as classically done in literature.

3. Main contributions

Whereas DBA aligns the whole dataset on an updated average signal using DTW, CDBA is based on the same process but adds a Constrained DTW. Let us first explain the behaviour of Constrained DTW in the following section.

3.1. Constrained DTW

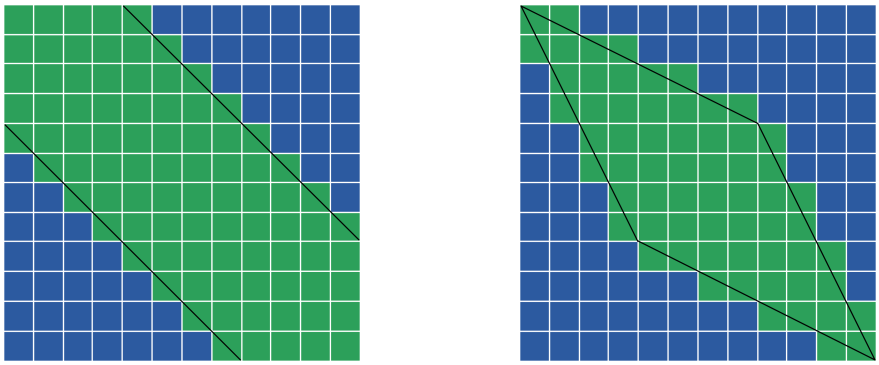
There exist mainly two ways to constrain the warping path: globally and locally.

A global constraint restricts the evolution of the warping path to a specific area to make it as diagonal as possible. Sakoe-Chiba band [19] and Itakura parallelogram [8] are the most common constraint areas and can be seen in Figure 5. These areas avoid the warping path to be vertical or horizontal for too long: it is constrained to move forward.

A local constraint consists in modifying the step conditions to penalize the vertical and horizontal paths. The penalty weight is applied to the cumulative distance computation to favor the diagonal. This weight must handle the variability of the signals to be averaged. In the extreme, the step conditions can prevent the path from being locally horizontal and vertical.

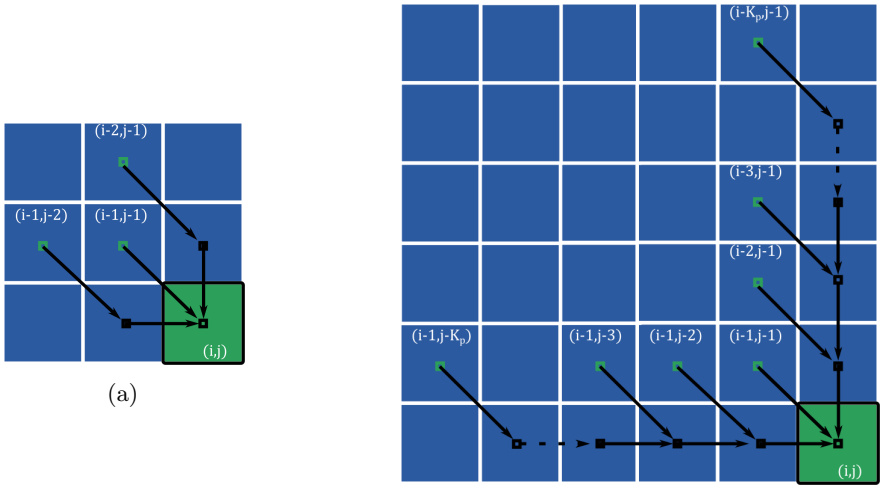
To avoid the problem of DBA introduced by vertical or horizontal path, we use a locally constrained DTW, called CDTW for Constrained Dynamic Time Warping [12].

This solution consists in locally preventing horizontal and vertical displacements. For that, some sideways displacements are artificially added to make the path as diagonal as possible. A first solution, presented in Figure 6a, consists in considering three local displacements. This configuration leads to the same accessible area than the global constraint proposed by Itakura (see Figure 5b). This process can be extended to a larger number of possible displacements around the diagonal. Let us call K_p the number of



(a) Sakoe-Chiba Band (b) Itakura Parallelogram

Figure 5: Common global constraints for the warping path in the literature. Comparing two signals, the path's research space is restricted to the green area.



(a) (b)

Figure 6: Local constrained applied during the creation of the warping path. The figure illustrates the allowed values used to compute the new $D_{i,j}$. (a) With a constraint slope $K_p = 2$; (b) the general case of local constrained admissible steps with any K_p.

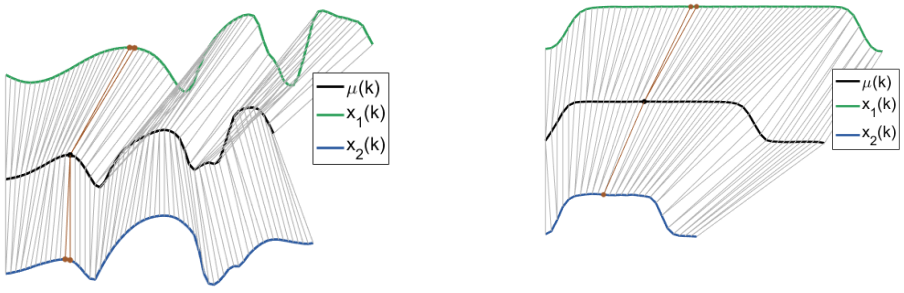
elementary steps of the longest local displacement (for example $K_p = 2$ in Figure 6a). The cumulative distance map $\mathbf{D}$ corresponding to Figure 6a is given by:

$$D_{i,j} = \min \begin{cases} D_{i-1,j-2} + d_{i,j-1} + d_{i,j} \\ D_{i-1,j-1} + d_{i,j} \\ D_{i-2,j-1} + d_{i-1,j} + d_{i,j} \end{cases} \tag{10}$$

whereas in the general case (Figure 6b), it is given by:

$$D_{i,j} = \min \begin{cases} D_{i-1,j-K_p} + \sum_{k=1}^{K_p} d_{i,j-(K_p-k)} \\ D_{i-1,j-(K_p-1)} + \sum_{k=1}^{K_p-1} d_{i,j-(K_p-1-k)} \\ \vdots \\ D_{i-1,j-2} + d_{i,j-1} + d_{i,j} \\ D_{i-1,j-1} + d_{i,j} \\ D_{(i-2,j-1} + d_{i-1,j} + d_{i,j} \\ \vdots \\ D_{i-(K_p-1),j-1} + \sum_{k=1}^{K_p-1} d_{i-(K_p-1-k),j} \\ D_{i-K_p,j-1} + \sum_{k=1}^{K_p} d_{i-(K_p-k),j} \end{cases} \tag{11}$$

As can be seen in Figure 7, using CDTW avoids pathological paths and leads to a better alignment between signals.

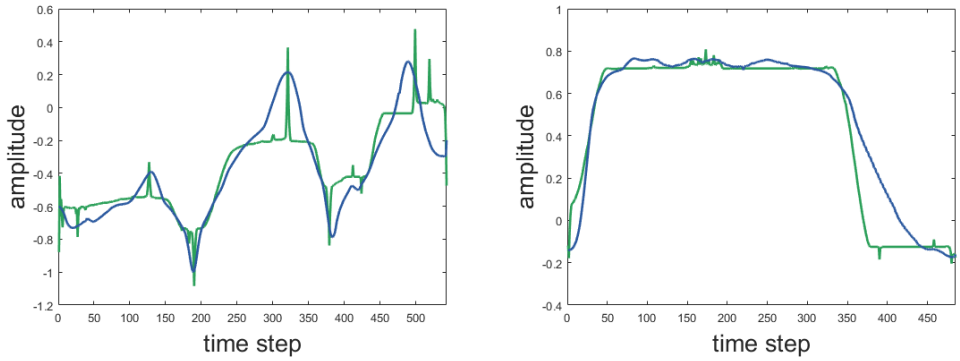


(a) (b)

Figure 7: One iteration of CDBA for two different sets of signals. The black signal is the average $\mu(k)$. Both blue and green signals are aligned on the black one with CDTW. Each point of $\mu(k)$ is updated as the average of the red points.

3.2. CDBA

Based on the work of Petitjean *et al.* [16], we propose to average a set of time-series by using a CDTW instead of a DTW. The CDTW, as explained in Section 3.1, provides a constrained warping path between two signals. Figure 8 compares two average signals obtained by DBA and CDBA after 4 iterations for the set of signals presented in Figure 4. Even if CDBA may associate several points of a signal to only one point of the other one as in DBA, Figure 8 shows that the average signal obtained with CDBA does not have the singular points of the one obtained with DBA. Moreover, it kept the shape of initial signals.



(a) (b)

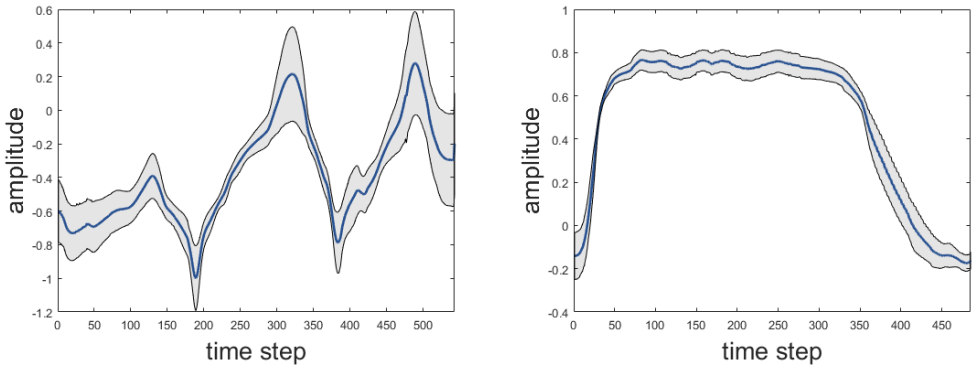
Figure 8: Illustration of the average signal obtained with DBA (in green) and CDBA (in blue) after 4 iterations for the 2 sets of signals of Figure 4. The average signal obtained with CDBA does not have the singular points of DBA.

As a set of points is badly represented by its simple mean and can be better modeled by a Gaussian which takes into account the deviation around this mean, we propose to introduce a tolerance in the representation of time-series. To the best of our knowledge, it is the first time that tolerance is introduced in the temporal signal modelling. This concept is presented in the following section.

3.3. Introducing tolerance in the modeling

To better represent a set of signals, we added to its modelling the tolerance that depicts admissible values around the average signal. This tolerance is computed for each time step of the average signal. It corresponds to the standard deviation of the sets of signals aligned on each point of the average time-series. The tolerance of the CDBA at time step k is thus computed as the standard deviation of the points aligned on this time step k of the average time-series. Algorithm 2 summarizes the proposed process to model a set of signals using CDBA.

Figure 9 depicts the average signals and tolerances for both sets of signals. The tolerance is depicted as $\pm(1 \times \sigma)$, where σ is the standard deviation among the learning set (see Algorithm 2 for its computation). Note that using tolerance with the DBA is possible but not relevant because of the pathological paths that would make the whole variability restricted to the extrema points.



(a) (b)

Figure 9: Average signal with CDBA for two different sets of signals. The tolerance corresponds to the gray area around the average signal, computed as $\pm(1 \times \sigma)$.

Algorithm 2 $CDBA : averagingCDTW$

Require: $x_0(k)$ of length M_0, $(x_l(k))_{l=1...L}$ of lengths M_l, IT, K_p
  $M_0 = K$, $\mu(k) \leftarrow x_0(k), \quad k = 1...K$
  for $it \in 1...IT$ **do**
    $assocTab[k] = \emptyset, k = 1...K$
    for $l \in 1...L$ **do**
      $\phi_{\mu x_l} \leftarrow CDTW(\mu, x_l, K_p)$
      $p \leftarrow length(\phi_{\mu x_l})$
      while $p \geq 1$ **do**
        $(k, n) \leftarrow \phi_{\mu x_l}(p)$
        $assocTab[k] \leftarrow assocTab[k] \cup \{x_l(n)\}$
        $p \leftarrow p - 1$
      end while
    end for
    for $k \in 1...K$ **do**
      $\mu(k) \leftarrow mean(assocTab[k])$
      $\sigma(k) \leftarrow std(assocTab[k])$
    end for
  end for
  return $\mu(k), \sigma(k), \quad k = 1...K$

In the following section, we validate this new time-series modelling on a classification task.

4. Application to time-series classification

4.1. Database

We used the UCR time-series Classification Archive dataset [4], also used in [13], [14], [15] and [16]. The characteristics of this dataset, containing 20 databases, are presented in Table 1. The 20 different databases contains various types of data, such as images (*Adiac*, *FaceAll*, *FaceFour*, *FISH*, *OSULeaf*, *SwedishLeaf*, *Yoga*), motions (*GunPoint*), ECG data (*ECG200*) or simulated data (*CBF*, *TwoPatterns*). The number of classes varies between 2 to 50 according to the database. Moreover, the length of all signals inside a database is fixed. It varies from 60 to 637 time steps depending on the database.

Database	**Nb cl.**	**Nb ex.**	**Length**
50words	50	905	270
Adiac	37	781	176
Beef	5	60	470
CBF	3	925	128
Coffee	3	56	286
ECG200	2	200	96
FISH	7	350	463
FaceAll	14	2570	131
FaceFour	4	112	350
GunPoint	2	200	150
Lighting2	2	121	637
Lighting7	7	143	319
OliveOil	4	60	570
OSULeaf	6	442	427
SwedishLeaf	15	1125	128
syntheticControl	6	600	60
Trace	4	200	275
TwoPatterns	4	5000	128
Wafer	2	7174	152
Yoga	2	3300	426

Table 1: Characteristics of the databases in UCR time-series Classification Archive [4]. For each database in first column, second column provides the number of classes it contains, the third one the number of signals and the fourth on the number of time step of signals (all signals inside a database have the same length).

4.2. Procedure

The test procedure is the same or all databases of the archive. Every data class $\mathcal{C}_c$ is represented by its average signal $\mu_c(k)$ and its tolerance $\sigma_c(k)$ obtained from the training signals. Each test signal $x(k)$ is aligned on $\mu_c(k)$ using DTW or CDTW depending on

whether $\mu_c(k)$ and $\sigma_c(k)$ are estimated by DBA or CDBA. The alignment leads to the warping path $\phi_{x\mu_c} = (\phi^x_{x\mu_c}, \phi^{\mu_c}_{x\mu_c})$ (see Equation 9). This path is used to compute the log-probability of signal $x(k)$ to belong to class $\mathcal{C}_c$ given by:

$$P(x \in \mathcal{C}_c) = \frac{1}{K} \sum_{k=1}^{K} \ln \left(\frac{1}{\sqrt{2\pi}\sigma_c(\phi^{\mu_c}_{x\mu_c}(k))} \exp \left\{ -\frac{(x(\phi^x_{x\mu_c}(k)) - \mu_c(\phi^{\mu_c}_{x\mu_c}(k)))^2}{2\sigma_c(\phi^{\mu_c}_{x\mu_c}(k))^2} \right\} \right) \quad (12)$$

The classification step is carried out looking for the class $\mathcal{C}_c$ that maximizes the log-probability $P(x \in \mathcal{C}_c)$.

Moreover, a 10-fold cross validation has been used. For that, each database was randomly divided into 10 subsets: each subset was then independently used for testing and the 9 reminders for learning, until all have been considered for testing.

4.3. Results

To validate our approach, we classify every test signals using the following different processes:

- DBA: we minimize the cumulative distance estimated by DTW between the average signal $\mu_c(k)$ obtained by DBA and test sample $x(k)$.
- CDBA without tolerance: we minimize the cumulative distance estimated by CDTW between the average signal $\mu_c(k)$ obtained by CDBA and test sample $x(k)$.
- CDBA with tolerance: we maximize Equation 12 with $(\mu_c(k), \sigma_c(k))$ estimated by CDBA and $\phi_{x\mu_c(k)}$ obtained by CDTW.

4.3.1. Improvements induced by the constraint (CDBA)

The classification results without tolerance are given in Table 2. The number of iterations to compute both DBA and CDBA is set to $IT = 10$ (the number of iterations has been fixed arbitrary: the algorithm converges quickly, and we could have chosen any value of $IT > 5$) and during the CDTW alignment process, the slope constraint K_p varies from 2 to 11.

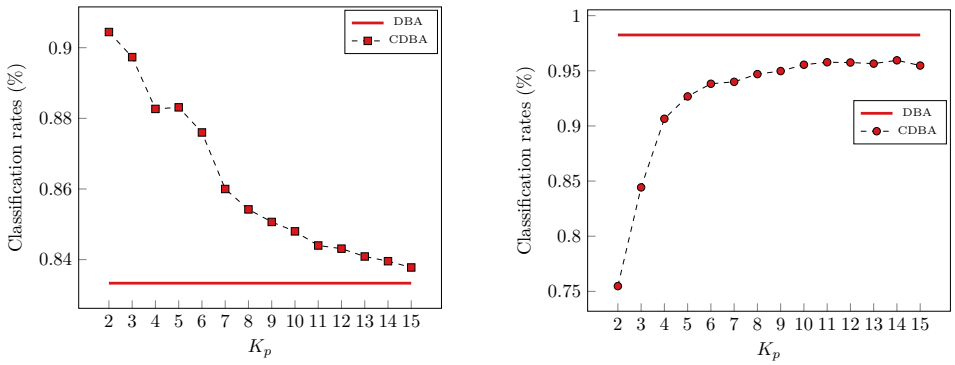
As can be seen on Table 2, the best CDBA classification results (last column) are greater or equal than DBA's ones, except for one database: *TwoPatterns*. A closer look to this particular databases shows that the corresponding signals have important delays that cannot be corrected by CDBA, even when using a high slope $K_p = 11$. However, we can notice that the higher K_p, the best CDBA results. Figure 10b presents the classification rates according to K_p for the database *TwoPatterns*. The rates increase with K_p and become closer to DBA results.

We can also remark that for many cases, high values of K_p lead to results close to DBA's ones. This is quite logical, because fewer constraints are then introduced (let us recall that CDBA is a constrained version of DBA).

Finally, an important conclusion is that the optimal value for K_p depends on the databases and more particularly, on the delay between signals inside the databases. A low value for K_p is adapted to slightly delayed signals as for example the database *FaceAll* (see Figure 10a) where the best recognition rate is obtained with $K_p = 2$. A high value is more suitable for very delayed signals (see Figure 10b).

Database	DBA	CDBA with K_p varying between 2 to 11 (%)										
	(%)	2	3	4	5	6	7	8	9	10	11	K_p opt
50Words	61,55	69,61	66,74	65,53	63,20	63,65	63,87	63,09	62,54	63,87	62,65	69,61
Adiac	46,10	48,66	47,50	47,25	46,48	46,22	46,48	46,35	46,35	46,74	46,48	48,66
Beef	25,00	25,00	26,67	28,33	30,00	30,00	30,00	26,67	26,67	25,00	26,67	30,00
CBF	96,45	94,09	96,13	95,27	94,19	95,38	96,13	96,13	96,45	96,56	96,56	96,56
Coffee	96,43	94,64	94,64	94,64	96,43	96,43	96,43	96,43	96,43	96,43	98,21	98,21
ECG	71,00	72,50	71,50	72,50	73,50	75,00	74,00	72,00	72,50	73,50	74,00	74,00
FISH	64,00	79,43	75,43	74,29	74,57	74,57	74,00	72,29	73,14	72,00	72,29	79,43
FaceAll	83,33	90,44	89,73	88,27	88,31	87,60	86,00	85,42	85,07	84,80	84,40	90,44
FaceFour	87,50	87,50	84,82	84,82	85,71	85,71	85,71	86,61	86,61	85,71	86,61	87,50
GunPoint	69,00	54,50	68,00	70,00	70,50	70,00	70,00	69,50	70,00	70,00	71,50	71,50
Lighting2	60,33	61,16	69,42	69,42	64,46	62,81	67,77	66,11	64,46	61,16	62,81	69,42
Lighting7	68,53	75,52	82,52	81,12	83,22	82,52	79,72	83,22	83,22	81,82	80,42	83,22
OliveOil	86,67	86,67	86,67	86,67	86,67	86,67	86,67	85,00	85,00	85,00	85,00	86,67
OSULeaf	45,25	45,70	43,21	42,31	38,69	40,05	40,50	39,59	37,78	36,88	38,69	45,70
SwedishLeaf	69,42	70,31	68,36	68,98	69,96	70,40	70,84	70,31	69,87	70,04	70,31	70,84
SyntheticControl	98,67	98,83	98,67	99,33	99,17	99,17	99,17	99,50	99,33	99,33	99,33	99,50
Trace	99,50	97,00	97,50	97,50	98,50	98,50	98,50	99,00	99,00	99,50	99,00	99,50
TwoPatterns	97,66	76,72	82,90	90,12	91,64	92,90	93,38	94,04	94,72	95,32	95,32	95,32
Wafer	41,14	66,43	65,01	69,36	73,42	72,26	72,93	69,44	67,66	63,51	61,96	73,42
Yoga	53,52	56,64	55,03	53,67	53,61	53,06	53,03	53,03	52,88	53,00	52,27	56,64
AVERAGE	**71,05**	**72,67**	**73,52**	**73,97**	**74,11**	**74,14**	**74,26**	**73,69**	**73,48**	**73,01**	**73,22**	**76,36**

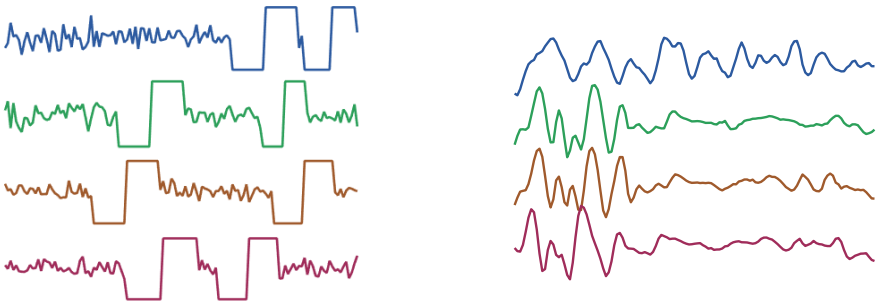
Table 2: Time-series' classification results obtained by DBA and CDBA with a slope constraint K_p varying from 2 to 11. The last column gives the best CDBA results for each database.



(a) *FaceAll* (b) *TwoPatterns*

Figure 10: Classification rates according to K_p for two datasets: (a) *FaceAll* that contains slightly delayed signals and (b) *TwoPatterns* that contains highly delayed signals. Note that the optimal K_p value depends on delays between time-series inside the same class.

To better understand this influence of K_p on the recognition rates, let us compare in the rest of this section the behavior of CDTW and DTW for the two particular databases (*FaceAll* and *TwoPatterns*).



(a) *TwoPatterns* (b) *FaceAll*

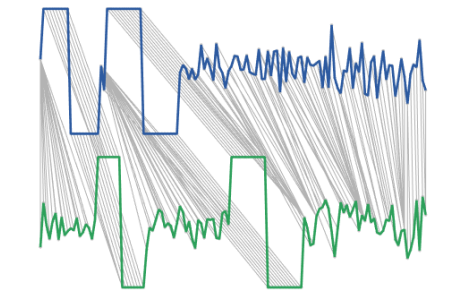
Figure 11: Signals used to compare CDTW and DTW behaviors. (a) Four signals of the same class in the database *TwoPatterns*. (b) One signal of class $\mathcal{C}_1$ (in blue), the three others belong to class $\mathcal{C}_2$ (in green, brown and red) in the dataset *Lighting7*.

Figure 11a illustrates four examples of time-series belonging to the same class $\mathcal{C}_1$ of the database *TwoPatterns*. Note that some of them may be very delayed as for example for the first and the second signals. Their alignments, obtained by DTW and CDTW with a slope constraint $K_p = 2$, are presented in Figures 12a and 12b respectively. This is important to notice that the CDTW-based algorithm is not able to manage the delay due to its limited slope. Actually, the alignment is incorrect and the signals' shapes are badly aligned (Figure 12b). On the contrary, as DTW-based algorithm allows vertical and horizontal displacements in the warping path, it can manage the delay and then correctly align signals (Figure 12a).

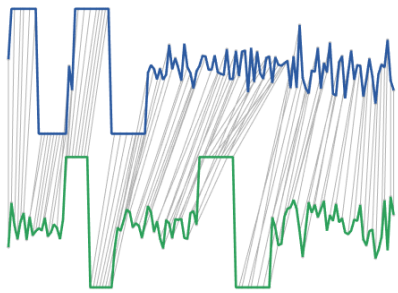
These phenomena occur several times during the average computation. Thus, while the average computed by DTW-based algorithm on 271 time-series keeps the signals' shape (Figure 12c), a poorer average, whose shape less corresponds to the original signals' ones, is given by CDTW-based algorithm (Figure 12d). However these bad cases occurred only with large delays, on 2 databases out of 20 in our study. Let us now consider other signals and highlight the interest of the CDTW-based method.

Figure 11b illustrates one time-series from a class $\mathcal{C}_1$ (in blue) and three other ones belonging to another class $\mathcal{C}_2$ (in green, brown and red) of the *FaceAll* dataset. We are now interested in the behavior of DTW and CDTW-based methods when aligning signals belonging to different classes.

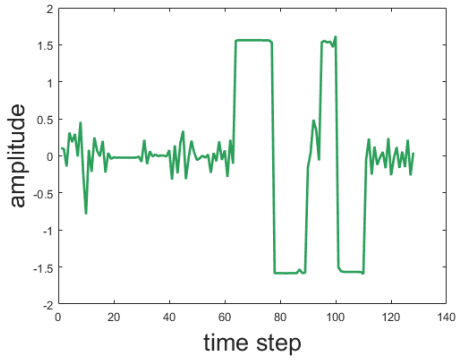
As the warping path of the DTW is not slope-constrained, it can contain vertical and horizontal local displacements to obtain signals as similar as possible. Figure 13a shows the DTW-based alignment for which one point of the first signal can match several points of the other one (see the gray lines symbolizing the matching possibilities). In Figure 13c,



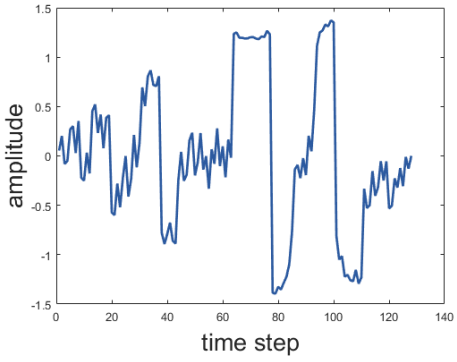
(a) *DTW*



(b) *CDTW*

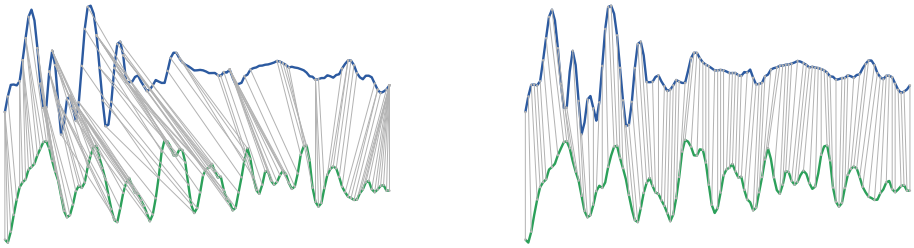


(c) *DBA*

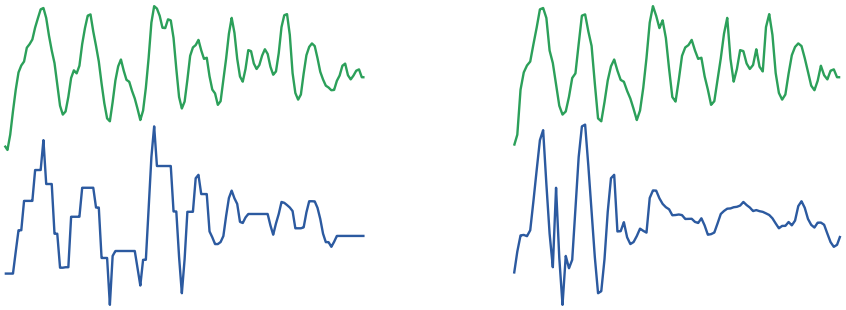


(d) *CDBA*

Figure 12: Case of two delayed signals in *TwoPatterns* database. (a) The signals aligned with DTW and (b) the signals aligned with CDTW. In both cases, the matching between points is symbolized by grey lines. In second row are given the resulting signal averages obtained with respectively DBA (c) and CDBA (d) processes.



(a) DTW (b) $CDTW$



(c) DTW (d) $CDTW$

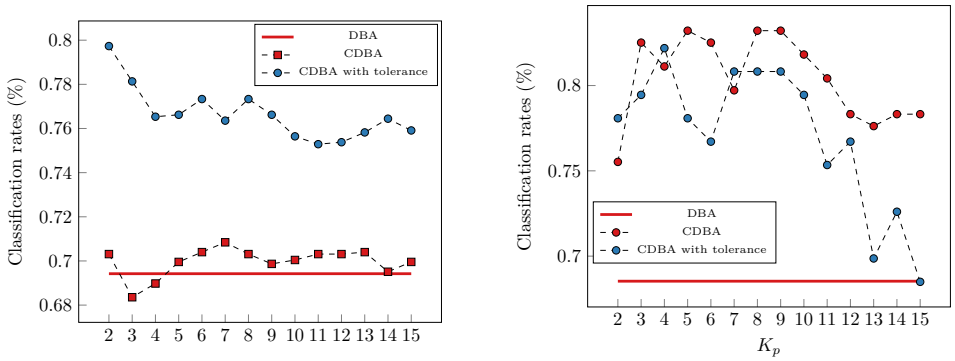
Figure 13: Illustration of the DTW and CDTW based alignments of signals belonging to different classes. First row: alignment with DTW (a) and CDTW (b). Second row: resulting wrapped time-series obtained by aligning signals with paths obtained by DTW (c) or CDTW (d).

the warped signals are plotted in the temporal baseline of the warping path. They are very similar and may lead to an incorrect classification. On the contrary, because of its slope limitation, CDTW does not face the same issue and the warped signals are very different (see Figures 13b and 13). This explains the higher classification results of CDBA in comparison with DBA on the *FaceAll* dataset and on a large number of databases (19 out of 20).

4.3.2. Improvements induced by the tolerancing

In this section, we compare classification results obtained with DBA, CDBA without tolerance and CDBA with tolerance. For CDBA-based methods, results have been optimized according to K_p for each database. Note that ideally, this value has to be optimized on a validation dataset. This was not done here on all databases because the number of examples was often too small to divide the learning dataset. The next paragraph presents a protocol for K_p optimization for the largest databases (*TwoPatterns*, *Wafer* and *Yoga*). Classification rates are shown in Table 3. For comparison purpose, Hidden Markov Models (HMM) were also tested, with the number of states optimized between 1 and 14. The state density probability function is represented using a gaussian, as for the proposed model. Moreover, the EM algorithm was used to estimate the parameter of the HMM. A first conclusion is that CDBA with tolerance gives the highest average classification results (81.13% with tolerance and 76.36% without). Thus, CDBA with tolerance increases classification results by more than 10% compared to DBA's ones.

For example, we can see in Figure 14a, for the *SwedishLeaf* database, that for all K_p values, classification results for CDBA are always higher with tolerance. In some particular cases, as for the *Lighting7* database (see Figure 14b), introducing the tolerance slightly decreases the classification results. However, we have noticed that only one class of this dataset is badly classified, and we hypothesize that this could be due to the distribution of the examples in this class, that is probably not Gaussian. The CDBA classification approach outperforms widely the HMM.



(a) *SwedishLeaf* (b) *Lighting7*

Figure 14: Classification rates according to the tolerance for two datasets.

Database	**HMM** (%)	**DBA** (%)	**CDBA** (%)	**CDBA+tol** (%)
50Words	10,39	61,55	69,61	73,26
Adiac	13,06	46,10	48,66	54,67
Beef	36,67	25,00	30,00	40,00
CBF	78,28	96,45	96,56	96,99
Coffee	55,36	96,43	98,21	100,00
ECG	64,00	71,00	75,00	76,50
FISH	27,71	64,00	79,43	88,29
FaceAll	39,60	83,33	90,44	95,24
FaceFour	43,75	87,50	87,50	96,43
GunPoint	67,50	69,00	71,50	88,00
Lighting2	67,77	60,33	69,42	80,99
Lighting7	46,85	68,53	83,22	82,52
OliveOil	33,33	86,67	86,67	91,67
OSULeaf	28,51	45,25	45,70	54,98
SwedishLeaf	36,80	69,42	70,84	79,73
SyntheticControl	87,00	98,67	99,50	98,50
Trace	81,50	99,50	99,50	99,00
TwoPatterns	53,36	97,66	95,32	96,02
Wafer	68,36	41,14	73,42	68,73
Yoga	54,52	53,52	56,64	61,00
AVERAGE	**49,72**	**71,05**	**76,36**	**81,13**

Table 3: Time-series' classification rates obtained with DBA and CDBA without and with tolerance. CDBA results are obtained after optimization of the slope constraint K_p on each database. CDBA with tolerance gives the highest average classification results.

4.3.3. Optimization of the constraint parameter K_p

As mentioned above, for the largest databases (*TwoPatterns*, *Wafer* and *Yoga*), K_p can be optimized accordingly to a validation set. For time measurement reasons, the $k-fold$ cross-validation was not considered here, and the initial training set was split into two subsets: one is used to compute the model (average and eventually the tolerance), the other one for the constraint parameter's optimization. This explain why the results are slightly different from those of the table 2. This optimization is made for both classification tasks, with and without tolerance. Results of this study are presented in Tables 4 and 5. We can note that for most of the cases, the optimized constraint parameter (last column) actually corresponds to the best K_p value. If not, it is very close to it.

Database	**DBA** (%)	**CDBA** (%)							
		2	3	4	5	6	7	8	$K_p = (K_p)_{opt}$
FaceAll	77.28	84.20	82.84	81.42	80.06	78.46	79.17	78.64	82.84
TwoPatterns	96.85	71.63	83.30	86.93	88.75	90,00	90.40	90.38	90.4
Wafer	51.10	70.72	72.71	69.89	92.29	71.90	83.49	86.20	92.30

Table 4: Classification rates without tolerance

Database	**DBA** (%)	**CDBA** (%)							
		2	3	4	5	6	7	8	$K_p = (K_p)_{opt}$
FaceAll	71.538	80.06	78.82	76.92	75.68	75.39	75.15	73.61	80.06
TwoPatterns	97.13	74.55	84.03	87.90	88.95	90.23	90.80	90.30	90.80
Wafer	72.05	76.48	72.99	64.59	80.13	80.39	76.09	76.35	76.48

Table 5: Classification rates with tolerance

4.3.4. Computation times

Table 6 gives the computation times for the classification of a time-series belonging to the database *50Words* (270 time steps as shown in table 1) obtained with DBA or CDBA, with or without using the tolerance. DTW algorithms were written in C++ and imported in Matlab R2016a as MEX-files in a Intel Core i7-4790 CPU 3.60 GHz. We can see, for CDBA, that the computation time increases with K_p, but stay reasonable.

Tolerance	**DBA**	**CDBA**						
		2	3	4	5	6	7	8
Without	0.083	0.099	0.115	0.140	0.175	0.200	0.251	0.272
With	0.107	0.119	0.146	0.165	0.192	0.229	0.257	0.294

Table 6: Computational time (in seconds) of the classification task with and without tolerance.

DBA and CDBA with $K_p = 2$ get almost the same computational time because both of them provide the same number of local displacements (3) at each timestep. We also note that these computation times increase with K_p, *i.e.* with the addition of local displacements.

4.3.5. Summary and discussion

Results showed that our new methodology globally outperforms DBA.

On the one hand, depending on the signal's shape, it can provide lower recognition rates than those of DBA. This can be explained by the slope limitation of the CDTW: it cannot correctly handle the alignment of two signals having a similar length and very different timings because of its vertical and horizontal displacement constraints. If we consider for example a signal delayed from another one, CDTW may fail to properly align these two signals because of the delay that cannot be overtaken by CDTW that is slope-limited (see database *TwoPatterns*). The DBA, on the contrary, is not affected in this case as it is not slope-limited. Thus, an alternative is to increase the slope-constraint K_p.

On the other hand, CDBA often outperforms DBA (see classification results for databases *50Words*, *Adiac*, *Beef*, *FaceAll*, *FISH*, *Lighting2*, *Lighting7*, *OSULeaf*, *SwedishLeaf*, *Wafer*...). Actually, DBA leads to an average signal with multiple singular points introduced by pathological paths as shown previously. A consequence is that a lot of signals (even those of other classes) can be aligned using DTW on this average signal. This obviously leads to false recognitions that explain the poor classification rate. The slope constraints introduced by CDBA lead to an average signal without singular point. Each class is thus better modeled and confusions are avoided during recognition.

Finally, we can see that the tolerance added to CDBA average plays an important role and often increases the classification rates.

5. Conclusion

In this paper, we proposed a modeling of time-series based on CDTW that outperforms the well-established DBA process in a classification task. It can be applied to any time-series, even with different intra-class temporal lengths. Our process relies on an average signal that preserves the shape and the length of the signals it represents, unlike the methods proposed in the literature. When most of the works focusing on time-series averaging are based on DTW, we proposed to use CDTW to avoid the problem of pathological warping paths. Moreover, we added to this averaging a tolerance at each time step that models time-series' variations around it.

Our process was validated in a classification task on the UCR time-series Classification Archive containing 20 databases containing different kinds of signals, of various lengths, shapes and classes. Our CDBA algorithm obtained most of the time higher global recognition rates than DBA ones, and even much higher when a tolerance based on the training data is added. However, for some databases, the DBA method provides higher results because of the slope limitation constraint of CDBA. This is the main limitation of our method since it can lead to a wrong alignment when both input signals have very different timings. To increase the performance of our method, a dedicated study of the automatic computation of the slope constraint parameter depending on the signals shapes must be done in future works. The influence of the initialisation step that fixes the length of the average signal on the recognition rates of DBA and CDBA must also be studied.

This research opens a wide range of perspectives in data mining process. Our approach is very generic and can be adapted to multiple case studies. It could also be generalized to

higher dimensions, considering covariance instead of standard deviation for the tolerance computation.

Acknowledgement

This study was partially supported by the funding of ENS Paris-Saclay.

References

[1] Achard, C., Qu, X., Mokhber, A., and Milgram, M. (2008). A novel approach for recognition of human actions with semi-global features. *Machine Vision and Applications*, 19(1):27–34.

[2] Babu, V., Prasanth, L., Sharma, R., Rao, G. V., and Bharath, A. (2007). Hmm-based online handwriting recognition system for telugu symbols. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 01*, pages 63–67. IEEE Computer Society.

[3] Burns, A.-M. (2013). *On the Relevance of Using Virtual Humans for Motor Skills Teaching: a case study on Karate gestures*. PhD thesis, Rennes 2.

[4] Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., and Batista, G. (2015). The ucr time series classification archive.

[5] Gales, M. and Young, S. (2007). The Application of hidden markov models in speech recognition. *Foundations and Trends® in Signal Processing*, 1(3):195–304.

[6] Gupta, L., Molfese, D., Tammana, R., and Simos, P. (1996). Nonlinear alignment and averaging for estimating the evoked potential. *IEEE Transactions on Biomedical Engineering*, 43(4):348–356.

[7] Hong, P., Turk, M., and Huang, T. (2000). Gesture modeling and recognition using finite state machines. In *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 410–415. IEEE Comput. Soc.

[8] Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):67–72.

[9] Keogh, E. and Pazzani, M. (2001). Derivative dynamic time warping. In *In First SIAM International Conference on Data Mining (SDM'2001)*.

[10] Mokhber, A., Achard, C., and Milgram, M. (2008). Recognition of human behavior by space-time silhouette characterization. *Pattern Recognition Letters*, 29(1):81–89.

[11] Muda, L., Begam, M., and Elamvazuthi, I. (2010). Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *CoRR*, 2(3).

[12] Müller, M. (2007). *Information retrieval for music and motion: with 26 tables*. Springer, Berlin.

[13] Niennattrakul, V. and Ratanamahatana, C. A. (2009). Shape averaging under time warping. In *6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 626–629. IEEE.

[14] Petitjean, F., Forestier, G., Webb, G. I., Nicholson, A. E., Chen, Y., and Keogh, E. (2016). Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. *Knowledge and Information Systems*, 47(1):1–26.

[15] Petitjean, F. and Gançarski, P. (2012). Summarizing a set of time series by averaging: From Steiner sequence to compact multiple alignment. *Theoretical Computer Science*, 414(1):76–91.

[16] Petitjean, F., Ketterlin, A., and Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693.

[17] Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[18] Rabiner, L. R. (1978). Considerations in dynamic time warping algorithms for discrete word recognition. *The Journal of the Acoustical Society of America*, 63(1):575–582.

[19] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

[20] Seto, S., Zhang, W., and Zhou, Y. (2015). Multivariate time series classification using dynamic time warping template selection for human activity recognition. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1399–1406. IEEE.

[21] Sigal, L., Sclaroff, S., and Athitsos, V. (2004). Skin color-based video segmentation under time-varying illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):862–877.

[22] Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., and Kitamura, T. (2000). Speech parameter generation algorithms for HMM-based speech synthesis. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 1315–1318. IEEE.

[23] Tomasi, G., van den Berg, F., and Andersson, C. (2004). Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. *Journal of Chemometrics*, 18(5):231–241.

[24] Zia, A., Sharma, Y., Bettadapura, V., Sarin, E. L., Ploetz, T., Clements, M. A., and Essa, I. (2016). Automated video-based assessment of surgical skills for training and evaluation in medical schools. *International Journal of Computer Assisted Radiology and Surgery*, 11(9):1623–1636.