Times series averaging and denoising from a probabilistic perspective on time-elastic kernels

Pierre-François Marteau

Times series averaging and denoising from a probabilistic perspective on time-elastic kernels

Pierre-Francois Marteau, *Member, IEEE,*
E-mail: see http://people.irisa.fr/Pierre-Francois.Marteau/

Abstract—In the light of regularized dynamic time warping kernels, this paper re-considers the concept of time elastic centroid for a set of time series. We derive a new algorithm based on a probabilistic interpretation of kernel alignment matrices. This algorithm expresses the averaging process in terms of a stochastic alignment *automata*. It uses an iterative agglomerative heuristic method for averaging the aligned samples, while also averaging the times of occurrence of the aligned samples. By comparing classification accuracies for 45 heterogeneous time series datasets obtained by first nearest centroid/medoid classifiers we show that: i) centroid-based approaches significantly outperform medoid-based approaches, ii) for the considered datasets, our algorithm that combines averaging in the sample space and along the time axes, emerges as the most significantly robust model for time-elastic averaging with a promising noise reduction capability. We also demonstrate its benefit in an isolated gesture recognition experiment and its ability to significantly reduce the size of training instance sets. Finally we highlight its denoising capability using demonstrative synthetic data: we show that it is possible to retrieve, from few noisy instances, a signal whose components are scattered in a wide spectral band.

Index Terms—Time series averaging Time elastic kernel Dynamic Time Warping Hidden Markov Model Classification Denoising.

1 Introduction

Since Maurice Fréchet's pioneering work [1] in the early 1900s, *time-elastic* matching of time series or symbolic sequences has attracted much attention from the scientific community in numerous fields such as information indexing and retrieval, pattern analysis, extraction and recognition, data mining, etc. This approach has impacted a very wide spectrum of applications addressing socio-economic issues such as the environment, industry, health, energy, defense and so on.

Among other time elastic measures, Dynamic Time Warping (DTW) was widely popularized during the 1970s with the advent of speech recognition systems [2], [3], and numerous variants that have since been proposed to match time series with a certain degree of time distortion tolerance.

The main issue addressed in this paper is time series or shape averaging in the context of a time elastic distance. Time series averaging or signal averaging is a long-standing issue that is currently becoming increasingly prevalent in the big data context; it is relevant for de-noising [4], [5], summarizing subsets of time series [6], defining significant prototypes, identifying outliers [7], performing data mining tasks (mainly exploratory data analysis such as clustering) and speeding up classification [8], as well as regression or data analysis processes in a big data context.

In this paper, we specifically tackle the question of averaging subsets of time series, not from considering the DTW measure itself as has already been largely exploited, but from the perspective of the so-called regularized DTW kernel (KDTW). From this new perspective, the estimation of a time series average or centroid can be readily addressed with a probabilistic interpretation of kernel alignment matrices allowing a precise definition of the average of a pair of time series from the expected value of local alignments of samples. The tests carried out so far demonstrate the robustness and the efficiency of this approach compared to the state of the art approach.

The structure of this paper is as follows: the introduction, the second section summarizes the most relevant related studies on time series averaging as well as DTW kernelization. In the third section, we derive a probabilistic interpretation of kernel alignment matrices evaluated on a pair of time series by establishing a parallel with a forward-backward procedure on a stochastic alignment automata. In the fourth section, we define the average of a pair of time series based on the alignment expectation of pairs of samples, and we propose an algorithm designed for the averaging of any subset of time series using a pairwise aggregating procedure. We present in the fifth section three complementary experiments to assess our approach against the state of the art, and conclude.

2 Related works

Time series averaging in the context of (multiple) time elastic distance alignments has been mainly addressed in the scope of the Dynamic Time Warping (DTW) measure [2], [3]. Although other time elastic distance measures such as the Edit Distance With Real Penalty (ERP) [9] or the Time Warp Edit Distance (TWED) [10] could be considered instead, without loss of generality, we remain focused throughout this paper on DTW and its kernelization.

• *P.-F. Marteau is with UMR CNRS IRISA, Université de Bretagne Sud, F-56000 Vannes, France.*

2.1 DTW and time elastic average of a pair of time series

A classical formulation of DTW can be given as follows. If d is a fixed positive integer, we define a time series of length T as a multidimensional sequence $v = v(i)$, such that, $\forall i \in \{1, .., T\}$, $v(i) \in \mathbb{R}^d$.

Definition 2.1. If u and v are two time series with respective lengths T_1 and T_2, an *alignment path* $\pi = (\pi_k)$ of length $p = |\pi|$ between u and v is represented by a sequence

$$\pi : \{1, \ldots, p\} \rightarrow \{1, \ldots, T_1\} \times \{1, \ldots, T_2\}$$

such that $\pi_1 = (1, 1)$, $\pi_p = (T_1, T_2)$, and (using the notation $\pi_k = (i_k, j_k)$, for all $k \in \{1, \ldots, p-1\}$, $\pi_{k+1} = (i_{k+1}, j_{k+1}) \in \{(i_k + 1, j_k), (i_k, j_k + 1), (i_k + 1, j_k + 1)\}$.

We define $\forall k\ \pi_k(1) = i_k$ and $\pi_k(2) = j_k$, as the index access functions at step k of the mapped elements in the pair of aligned time series.

In other words, a warping path defines a way to travel along both time series simultaneously from beginning to end; it cannot skip a point, but it can advance one time step along one series without advancing along the other, thereby justifying the term *time-warping*.

If δ is a distance on $\mathbb{R}^d$, the global *cost* of a warping path π is the sum of distances (or squared distances or local costs) between pairwise elements of the two time series along π, i.e.:

$$\text{cost}(\pi) = \sum_{(i_k, j_k) \in \pi} \delta(v_{i_k}, w_{j_k})$$

A common choice of distance on $\mathbb{R}^d$ is the one generated by the L^2 norm:

$$\delta(x, y) = \|x - y\|_2^2 = \sum_{l=1}^{d} (x_l - y_l)^2.$$

Definition 2.2. For a pair of finite time series X and Y, any warping path has a finite length, and thus the number of existing warping paths is finite. Hence, there exists at least one path π^* whose cost is minimal, so we can define $\text{DTW}(X, Y)$ as the minimal cost taken over all existing warping paths. Hence

$$\begin{aligned} \text{DTW}(X, Y) &= \min_{\pi} \text{cost}(\pi(X, Y)) \\ &= \text{cost}(\pi^*(X, Y)). \end{aligned} \quad (1)$$

Definition 2.3. From the DTW measure, [11] have defined the time elastic average $a(X, Y)$ of a pair of time series X and Y as the time series (a_k) whose elements are $a_k = \textit{mean}(X(\pi_k^*(1)), Y(\pi_k^*(2))),\ \forall k \in 1, \cdots, |\pi^*|$, where *mean* corresponds to the definition of the mean in Euclidean space.

2.2 Time elastic centroid of a set of time series

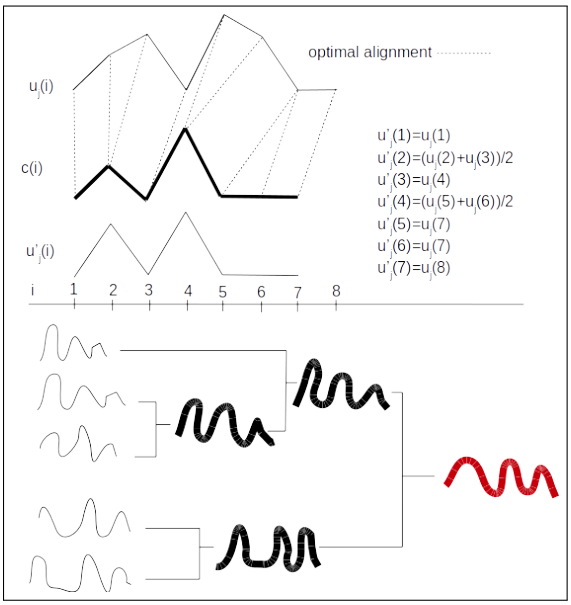
A single alignment path is required to calculate the time elastic centroid of a pair of time series (Def. 2.1). However, multiple path alignments need to be considered to evaluate the centroid of a larger set of time series. Multiple alignments have been widely studied in bioinformatics [12], and it has been shown that determining the optimal alignment of a set of sequences under the sum of all pairs (SP) score scheme is a NP-complete problem [13] [14]. The time and space complexity of this problem is $O(L^k)$, where k is the number of sequences in the set and L is the length of the sequences when using dynamic programming to search for an optimal solution [15]. This latter result applies to the estimation of the time elastic centroid of a set of k time series with respect to the DTW measure. Since the search for an optimal solution becomes rapidly intractable with increasing k, sub-optimal heuristic solutions have been subsequently proposed, most of them falling into one of the following three categories.

2.2.1 Progressive heuristics

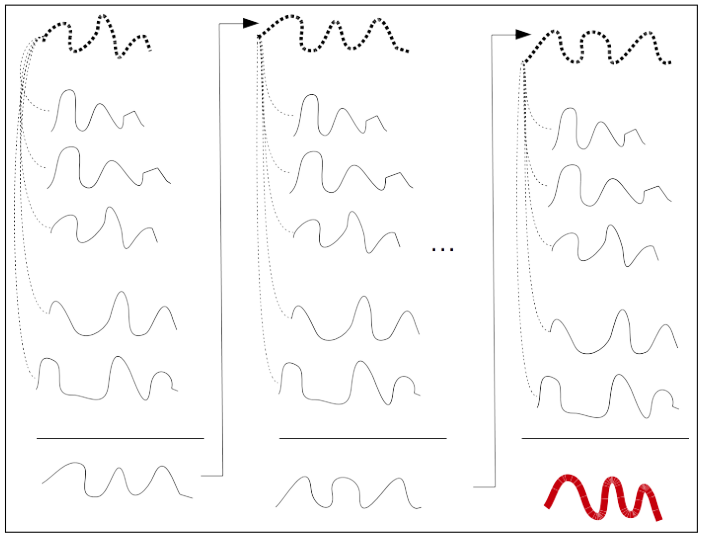
Progressive heuristic methods estimate the time elastic centroid of a set of k time series by combining pairwise centroids (Def. 2.3). This kind of approach constructs a binary tree whose leaves correspond to the time series of the data set, and whose nodes correspond to the calculation of a local pairwise centroid, such that, when the tree is complete, the root is associated with the estimated data set centroid. The proposed strategies differ in the way the tree is constructed. One popular approach consists of providing a random order for the leaves, and then constructing the binary tree up to the root using this ordering [11]. Another approach involves constructing a dendrogram (a hierarchical ascendant clustering) from the data set and then using this dendrogram to calculate pairwise centroids starting with the closest pairs of time series and progressively aggregating series that are farther away [16] as illustrated on the left of Figure 1. Note that these heuristic methods are entirely based on the calculation of a pairwise centroid, so they do not explicitly require the evaluation of a DTW centroid for more than two time series. Their degree of complexity varies linearly with the number of time series in the data set.

2.2.2 Iterative heuristics

Iterative heuristics are based on an iterated three-step process. For a given temporary centroid candidate, the first step consists of calculating the inertia, i.e. the sum of the DTW distances between the temporary centroid and each time series in the data set. The second step (Figure 1a top) evaluates the best pairwise alignment with the temporary centroid $c(i)$, of length L, for each time series $u_j(i)$ in the data set ($j \in \{1 \cdots n\}$), where i is the timestamp. A new time series of length L, $u'_j(i)$ is thus constructed that contains the contributions of all the samples of time series $u_j(i)$, but with time being possibly stretched (duplicate samples) or compressed (average of successive samples) according to the best alignment path as exemplified in Figure 1a, top left side. The third step consists in producing a new temporary centroid candidate $c(i)$ from the set $\{u'_j(i)\}$ by successively averaging (in the sense of the Euclidean



(a) Pairwise average (top) and Progressive agglomeration (bottom)



(b) Iterative agglomeration with refinement

Fig. 1. Pairwise averaging (top left), progressive hierarchical with similar first agglomeration (bottom left) v.s. iterative agglomeration (right) strategies. Final centroid approximations are presented in red bold color. Temporary estimations are presented using a bold dotted black line

centroid), the samples at every timestamp i of the $u'_j(i)$ time series. Basically, we have $c(i) = 1/n \cdot \sum_{j=1..n} u'_j(i)$.

Then, the new centroid candidate replaces the previous one and the process is iterated until the inertia is no longer reduced or the maximum number of iterations is reached. Generally, the first temporary centroid candidate is taken as the DTW medoid of the considered data set. This process is illustrated on Figure 1b. The three steps of this heuristic method were first proposed in [17]. The iterative aspect of this heuristic approach was initially introduced by [18] and refined by [6] who introduced the Dtw Barycenter Averaging (DBA) algorithm. Note that, in contrast to the progressive method, this kind of approach needs to evaluate, at each iteration, all the alignments with the current centroid candidate. The complexity of the iterative approach is higher than the progressive approach, the extra computational cost being linear with the number of iterations. More sophisticated approaches have been proposed to escape some local minima. For instance [19] have evaluated a genetic algorithm for managing a population of centroid candidates, thus improving with some success the straightforward iterative heuristic methods.

2.2.3 *Optimization approaches*

Given the entire set of time series $\mathbb{S}$ and a subset of n time series $S = \{X_j\}_{j=1\cdots n} \subseteq \mathbb{S}$, optimization approaches attempt to estimate the centroid of S from the definition of an optimization problem, which is generally expressed by Equation 2 given below:

$$c = \underset{s \in \mathbb{S}}{\operatorname{argmin}} \sum_{j=1}^{n} \mathrm{DTW}(s, X_j) \tag{2}$$

Among other works, some attempt to use this kind of direct approach for the estimation of time elastic centroid estimation was recently addressed in [20], [21] and [22].

In [20] the authors detail a Canonical Time Warp (CTW) and a Generalized version of it (GCTW) [21] that combines DTW and CCA (Canonical Correlation Analysis) for temporally aligning multi-modal motion sequences. From a least square formulation for DTW, a non-convex optimization problem is handled by means of a coordinate-descent approach that alternates between multiple temporal alignments using DTW (or a variant exploiting a set of basis functions to parameterized the warping paths) and spatial projections using CCA (or a multi-set extension of CCA). If these approaches have not been designed to explicitly propose a centroid estimation, they provide multi-alignment paths that can straightforwardly be used to compute a centroid estimate. As an extension to CTW, GCTW requires the set-up of generally "smooth" function basis that constrain the shape of the admissible alignment paths. This ensures the computational efficiency of GCTW, but in return it may induce some drawback, especially when considering the averaging of "unsmoothed" time series that may involve very "jerky" alignment paths. Hence, the choice of the function basis requires some expertise on the data.

In [22], a non-convex constrained optimization problem is derived, by integrating a temporal weighting of local sample alignments to highlight the temporal region of interest in a time series data set, thus penalizing the other temporal regions. Although the number of parameters to optimize is linear with the size and the dimensionality of the time series, the two steps gradient-based optimization process they derived is very computationally efficient and shown to outperform the state of the art approaches on some challenging scalar and multivariate data sets. However, as numerous local *optima* exist in practice, the method is not guaranteed to converge toward the best possible centroid,

which is anyway the case in all other approaches. Furthermore, their approach, due to combinatorial explosion, cannot be adapted for time elastic kernels like the one addressed in this paper and described in section 2.4.

2.3 Discussion and motivation

According to the state of the art in time elastic centroid estimation, an exact centroid, if it exists, can be calculated by solving a NP-complete problem whose complexity is exponential with the number of time series to be averaged. Heuristic methods with increasing time complexity have been proposed since the early 2000s. Simple pairwise progressive aggregation is a less complex approach, but which suffers from its dependence on initial conditions. Iterative aggregation is reputed to be more efficient, but entails a higher computational cost. It could be combined with ensemble methods or soft optimization such as genetic algorithms. The non-convex optimization approach has the merit of directly addressing the mathematical formulation of the centroid problem in a time elastic distance context. This approach nevertheless involves a higher complexity and must deal with a relatively large set of parameters to be optimized (the weights and the sample of the centroid). Its scalability could be questioned, specifically for high dimensional multivariate time series.

It should also be mentioned that some criticisms of these heuristic methods have been made in [23]. Among other drawbacks, the fact that DTW is not a metric could explain the occurrence of unwanted behavior such as centroid drift outside the time series cluster to be averaged. We should also bear in mind that keeping a single best alignment can increase the dependence of the solution on the initial conditions. It may also increase the aggregating order of the time series proposed by the chosen method, or potentially enhance the convergence rate.

In this study, we do not directly address the issue of time elastic centroid estimation from the DTW perspective, but rather from the point of view of the regularized dynamic time warping kernel (KDTW) [24]. Although this perspective allows us to consider centroid estimation as a preimage problem, which is in itself another optimization perspective, we rather show that the KDTW alignment matrices computation can be described as the result of applying a forward-backward algorithm on a stochastic alignment automata. This probabilistic interpretation of the pairwise alignment of time series leads us to propose a robust averaging scheme for any set of time series that interpolate jointly along the time axis and in the sample space. Furthermore, this scheme significantly outperforms the current state of the art method, as shown by our experiments.

2.4 Time elastic kernels and their regularization

The **Dynamic Time Warping** (DTW) distance between two time series X_1^p and Y_1^q of lengths p and q respectively, [2], [3] as defined in Equation 1 can be recursively evaluated as

$$d_{dtw}(X_1^p, Y_1^q) = d_E^2(X(p), Y(q)) + \text{Min} \begin{cases} d_{dtw}(X_1^{p-1}, Y_1^q) \\ d_{dtw}(X_1^{p-1}, Y_1^{q-1}) \\ d_{dtw}(X_1^p, Y_1^{q-1}) \end{cases} \quad (3)$$

where $d_E(X(p), Y(q))$ is the Euclidean distance (sometimes the square of the Euclidean distance is preferred) defined on $\mathbb{R}^k$ between the two positions in sequences X_1^p and Y_1^q taken at times p and q, respectively.

Apart from the fact that the triangular inequality does not hold for the DTW distance measure, it is furthermore not possible to define a positive definite kernel directly from this distance. Hence, the optimization problem, which is inherent to the learning of a kernel machine, is no longer convex and could be a source of limitation due to the emergence of local minima.

Regularized DTW: seminal work by [25], prolonged recently by [24] lead us to propose new guidelines to ensure that kernels constructed from elastic measures such as DTW are positive definite. A simple instance of such a regularized kernel, derived from [24], can be expressed as a convolution kernel, which makes use of two recursive terms:

$$\begin{aligned} &\text{KDTW}(X_1^p, Y_1^q) = K_{dtw}(X_1^p, Y_1^q) + K'_{dtw}(X_1^p, Y_1^q) \\ &K_{dtw}(X_1^p, Y_1^q) = \\ &\quad \tfrac{1}{3} e^{-\nu d_E^2(X(p), Y(q))} \cdot \Big(h(p-1,q) K_{dtw}(X_1^{p-1}, Y_1^q) \\ &\qquad + h(p-1,q-1) K_{dtw}(X_1^{p-1}, Y_1^{q-1}) \\ &\qquad + h(p,q-1) K_{dtw}(X_1^p, Y_1^{q-1}) \Big) \\ &K'_{dtw}(X(p), Y(q)) = \\ &\quad \tfrac{1}{3} \cdot \Big(h(p-1,q) K'_{dtw}(X_1^{p-1}, Y_1^q) e^{-\nu d_E^2(X(p), Y(q))} \\ &\quad + \Delta_{p,q} h(p-1,q-1) K'_{dtw}(X_1^{p-1}, Y_1^{q-1}) e^{-\nu d_E^2(X(p), Y(q))} \\ &\quad + h(p,q-1) K'_{dtw}(X_1^p, Y_1^{q-1}) e^{-\nu d_E^2(X(p), Y(q))} \Big) \end{aligned} \quad (4)$$

where $\Delta_{p,q}$ is the Kronecker symbol, $\nu \in \mathbb{R}^+$ is a *stiffness* parameter which weights the local contributions, i.e. the distances between locally aligned positions, $d_E(.,.)$ is a distance defined on $\mathbb{R}^k$, and h is a symmetric binary non negative function, usually in $\{0, 1\}$, used to define a symmetric corridor around the main diagonal to limit the "time elasticity" of the kernel. For the remaining of the paper we will not consider any corridor, hence $h(.,.) = 1$ everywhere.

The initialization is simply $K_{dtw}(X(0), Y(0)) = K'_{dtw}(X(0), Y(0)) = 1$.

The main idea behind this regularization is to replace the operators $\min$ and $\max$ (which prevent symmetrization of the kernel) by a summation operator. This allows us to consider the best possible alignment, as well as all the best (or nearly the best) paths by summing their overall cost. The parameter ν is used to check what is termed as nearly-the-best alignment, thus penalizing alignments that are too far away from the optimal ones. This parameter can be easily optimized through a cross-validation.

For each alignment path, KDTW evaluates the product of local alignment costs $e^{-\nu d_E^2(X(p), Y(q))} \leq 1$ occurring along the path. This product can be very small depending on the size of the time series and the selected value for ν. This is the source for a diagonal dominance problem in the Gram matrix. But, above all, this requires to balance the choice of the ν value according to the lengths of the matched time

series. This is the main (and probably the only) limitation of the KDTW kernel: the selectivity or bandwidth of the local alignment kernels needs to be adjusted according to the lengths of the matched time series.

3 Stochastic alignment process

To introduce a probabilistic paradigm to the time elastic averaging of time series, we first consider the pairwise alignment process as the output of a stochastic automata. The stochastic alignment process that we propose finds its roots in the forward-backward algorithm defined for the learning of Hidden Markov Models (HMM) [26] and in the parallel between HMM and DTW that is proposed in [27], [28] and in a more distant way in [29]. However we differ from these founding works (and others) in the following

1) we do not construct a parallel with DTW, but with its kernelized variant KDTW
2) [28] only consider an optimal alignment path (exploiting the Viterbi algorithm) while we consider the whole set of possible alignments (as in [27])
3) [27] construct an asymmetric classical left-right HMM (one time series plays the role of the observation sequence, while the other plays the role of the state sequence). With a similar idea [29] proposes a generative mixture model along a discrete time grid axis with local and global time warp capability. We construct instead an alignment process, that sticks on the DTW recursive definition without any other hypothesis on the structure of the automata, and for which the two aligned time series play the role of the observation sequence, and the set of states corresponds to the set of all possible sample pairs alignments.

3.1 pairwise alignment of time series as a Markov model

Let $o_1^n = o_1 o_2 \cdots o_n$ and $o'^{n'}_1 = o'_1 o'_2 \cdots o'_{n'}$ be two discrete time series (observations) of length n and n' respectively. To align this two time series, we define a stochastic alignment automata as follows. First we consider the set of state variables $\mathcal{S} = \{S_{1,1}, S_{1,2}, \cdots, S_{n,n'}\}$. Each $S_{i,j}$ characterizes the alignment between observed samples o_i and o'_j. The posterior probability for all state variables, $S_{i,j}$, given the sequences of observations o_1^n and $o'^{n'}_1$ is $P(S_{i,j}|o_1^n; o'^{n'}_1)$.

The transitions probabilities between states are driven by a tensor $\mathbf{A} = [a_{ij;kl}]$, where $a_{ij;kl} = P(S_{k,l}|S_{i,j}), \forall (k,l)$ and $(i,j) \in \{1 \cdots n\} \times \{1 \cdots n'\}$. $\mathbf{A}$ can be defined accordingly to the standard DTW definition, namely

$$a_{ij;kl} = \begin{cases} \frac{1}{3} \text{ IF } \begin{cases} (k = i \text{ AND } l = j+1) \\ \text{OR } (k = i+1 \text{ AND } l = j+1) \\ \text{OR } (k = i+1 \text{ AND } l = j) \end{cases} \\ 0 \text{ OTHERWISE.} \end{cases} \quad (5)$$

The $1/3$ factor ensures that the transition matrix equivalent to $\mathbf{A}$ is stochastic, basically

$$\forall i,j \ \sum_{kl} a_{ij;kl} = 1 \quad (6)$$

Notice that any tensor $\mathbf{A}$ satisfying equation (6) could be considered at this level instead of the previous DTW surrogate tensor.

Furthermore, each state is observable through the so-called emission probabilities which are defined by a set of functions $\mathbf{B} = [b_{ij}(u,v)]$, where $b_{ij}(u,v) = P(o_u, o'_v|S_{ij})$, $\forall (u,v)$ and $(i,j) \in \{1 \cdots n\} \times \{1 \cdots n'\}$. The b_{ij} functions are estimated as follows

$$b_{ij}(u,v) = \begin{cases} \kappa(o_u, o'_v) \text{ IF } u = i \text{ AND } v = j \\ 0 \text{ OTHERWISE.} \end{cases} \quad (7)$$

where $0 \leq \kappa(o_u, o'_v) \leq 1$ is any density kernel or discrete distribution measure (e.g. $\kappa(.,.) \propto e^{-\nu d_E^2(.,.)}$ for KDTW).

Here we differ from the classical HMM: for our construction, the states are not truly hidden since the knowledge of the local observation pair (a local alignment (o_u, o'_v)) fully determines the state (S_{uv}). However, the main difference lies in the nature of the observation sequence itself. Unlike HMM, our observation consists of a pair of subsequences that are not traveled necessarily synchronously, but according to the structure of the transition tensor $\mathbf{A}$. For instance, given the DTW tensor described by equation (5), from a current state associated to the alignment (o_u, o'_v), three possible alignments can be reached at the next transition: (o_{u+1}, o'_v), (o_u, o'_{v+1}) or (o_{u+1}, o'_{v+1}).

Finally let $\mathbf{u}$ be the initial state probability vector defined by $\forall (i,j) \in \{1 \cdots n\} \times \{1 \cdots n'\}$, $\mathbf{u}_{ij} = 1$ if $i = j = 1$, 0 otherwise.

Thereby, the stochastic alignment automata is fully specified by the triplet $\theta = (\mathbf{A}, \mathbf{B}, \mathbf{u})$, where $\mathbf{A}$ only depends on the lengths n and n' of the observations, and $\mathbf{B}$ depends on the complete pair of observations $o_{1:n}$ and $o'_{1:n'}$.

3.2 Forward-backward alignment algorithm

We derive the forward-backward alignment algorithm for our stochastic alignment automata from its classical derivation that was defined for Hidden Markov Models [26].

For all $S \in \mathcal{S}$, the posterior probability $P(S|o_{1:n}; o'_{1:n'}; \theta)$ is decomposed into forward/backward recursions as follows:

$$\begin{aligned} P(S|o_1^n; o'^{n'}_1; \theta) &= \frac{P(o_1^n; o'^{n'}_1; S|\theta)}{P(o_1^n; o'^{n'}_1; |\theta)} \\ &= \frac{P(o_1^t; o_t^n; o'^{t'}_1; o'^{n'}_{t'}; S|\theta)}{P(o_1^n; o'^{n'}_1|\theta)} \\ &= \frac{P(o_t^n; o'^{n'}_{t'}|S; \theta) P(S; o_1^t; o'^{t'}_1|\theta)}{P(o_1^n; o'^{n'}_1|\theta)} \end{aligned} \quad (8)$$

The last equation results from the application of the Bayes rule and the conditional independence of $o_t^n; o'^{n'}_{t'}$ and $o_1^t; o'^{t'}_1$ given S, θ.

Let $\alpha_{t,t'} = P(o_1^t; o'^{t'}_1; S_{t,t'}|\theta)$ be the probability of the alignment of the pair of partial observation sequences $(o_1^t, o'^{t'}_1)$ produced by all possible state sequences that end at

state $S_{t,t'}$. $\alpha_{t,t'}$ can be recursively evaluated as the forward procedure

$$\begin{cases} \alpha_{11} = u_{11}\kappa(o_1, o'_1) \\ \alpha_{tt'} = \kappa(o_t, o'_{t'}) \sum\limits_{u,v \in \mathcal{F}_{t,t'}} \alpha_{uv} a_{uv;tt'} \end{cases} \quad (9)$$

where $\mathcal{F}_{t,t'}$ is the subset of states allowing to reach the state $S_{t,t'}$ in a single transition. For the DTW tensor $\mathbf{A}$ (Eq. 5), we have $\mathcal{F}_{t,t'} = \{S_{t-1,t'}, S_{t,t'-1}, S_{t-1,t'-1}\}$.
Notice that in this case, $\alpha_{n,n'} = K'_{dtw}(o_1^n, o'^{n'}_1)$.

Similarly let $\beta_{t,t'} = P(o_t^n; o'^{n'}_{t'}|S;\theta)$ be the probability of the alignment of the pair of partial sequences $(o_t^n, o'^{n'}_{t'})$ given starting state $S_{t,t'}$. $\beta_{t,t'}$ can be recursively evaluated as the backward procedure

$$\begin{cases} \beta_{nm} = 1 \\ \beta_{tt'} = \sum\limits_{u,v \in \mathcal{B}_{t,t'}} \beta_{uv} a_{tt';uv} \kappa(o_t, o'_{t'}) \end{cases} \quad (10)$$

where $\mathcal{B}_{t,t'}$ is the subset of states that can be reached from the state $S_{t,t'}$ in a single transition. For the DTW tensor $\mathbf{A}$ (Eq. 5), we have $\mathcal{B}_{t,t'} = \{S_{t+1,t'}, S_{t,t'+1}, S_{t+1,t'+1}\}$.

Hence from Eq. 8, we get

$$P(S_{t,t'}|o_1^n; o'^{n'}_1; \theta) = \frac{\alpha_{t,t'}\beta_{t,t'}}{P(o_1^n; o'^{n'}_1; |\theta)} \quad (11)$$

Any tensor $\mathbf{A}$ satisfying equation (6) is not eligible: for the $\alpha_{t,t'}$ and $\beta_{t,t'}$ recursions to be calculable, one have to impose *linearity*. Basically $\alpha_{t,t'}$ cannot depends on any $\alpha_{u,v'}$ that is not previously evaluated. The constraint we need to impose is that the time stamps are locally increasing, i.e. if $\alpha_{t,t'}$ depends on any $\alpha_{u,v'}$, then necessarily $[(t < u \text{ and } t' \leq v') \text{ or } (t \leq u \text{ and } t' < v')]$. The same applies for the $\beta_{t,t'}$ recursion.

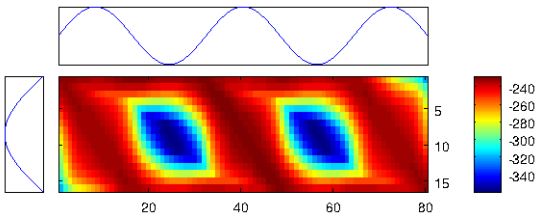


Fig. 2. Forward Backward matrix (logarithmic values) for the alignment of a positive halfwave with a sinus wave. The dark red color represents high probability states, while dark blue color represents low probability states.

As an example, Figure 2 presents the Forward Backward (FB) matrix ($FB(t,t') = P(S_{t,t'}|o_{1:n}; o'_{1:n'}; \theta)$) corresponding to the alignment of a positive halfwave with a sinus wave. The three areas of likely alignment paths are clearly identified in dark red colors.

3.3 Parallel with KDTW

A direct parallel exists between KDTW and the previous Markov process. It follows from the forward equation (eq. 9) that

$$\begin{aligned} K_{dtw}(X(k), Y(l)) &= \sum_{i,j} a_{ij,kl} b_{kl} K_{dtw}(X(i), Y(j)) \\ &= \kappa(X(k), Y(l)) \sum_{i,j} a_{ij,kl} K_{dtw}(X(i), Y(j)) \end{aligned} \quad (12)$$

where $\mathbf{A} = [a_{ij,kl}]$ is defined in equation (5), and $\mathbf{B} = [b_{kl}]$, defined in equation (7), is such that $b_{kl} = e^{-\nu d_E^2(x(k),y(l))}$. Hence, the KDTW recursion coincides with the forard recursion (Eq. 8). Similarly, we can assimilate the backward recursion (eq. 10) to the K_{dtw} evaluation of the pair of time series obtained by inverting X and Y along the time axis. Hence, the forward-backward matrix elements (eq. 11) can be directly expressed in terms K_{dtw} recursions.

Furthermore, the corridor function $h()$ that occurs in the KDTW recursion (Eq. 4) modifies directly the structure of the transition tensor $\mathbf{A}$ by setting $a_{ij,kl} = 0$ whenever $h(i,j) = 0$ or $h(k,l) = 0$. neighbor states may be affected also by the normalization that is required to maintain the $\mathbf{A}$ stochastic.

3.4 Time elastic averaging of a pair of time series

Let us introduce the marginal probability of subset $S_{t,\bullet} = \{S_{t1}, S_{t2}, \cdots, S_{tn'}\}$ given the observations o and o', namely that sample o_t is aligned with the samples of series $o'^{n'}_1$

$$P(S_{t,\bullet}) = \sum_{t'} P(S_{t,t'}|o_1^n; o'^{n'}_1; \theta) \quad (13)$$

we define a normalized posterior probability as

$$\begin{aligned} &\hat{P}(S_{t,t'}|o_1^n; o'^{n'}_1; \theta) = \\ &\quad P(S_{t,t'}|o_1^n; o'^{n'}_1; \theta) / P(S_{t,\bullet}|o_{1:n}; o'_{1:n'}; \theta) \end{aligned} \quad (14)$$

$\hat{P}(S_{t,t'}|o_1^n; o'^{n'}_1; \theta)$ quantifies the probability that samples o_t and $o'_{t'}$ are aligned by the automata given that sample o_t is aligned with one of the samples of $o'^{n'}_1$.

The normalized posterior probabilities, equation (14), are at the basis of our procedure for averaging a set of time series.

Let o_1^n and $o'^{n'}_1$ two time series. The pairwise average estimate of o and o' is defined as the pair (c, τ) where c is a time series of length $n = \texttt{max}(n, n')$ and τ the time stamps associated to c.

$$\begin{aligned} &\forall i \leq n \\ &\quad c_1(i) = \sum_{j=1}^{n'} o'_j \hat{P}(S_{ij}|o_1^n; o'^{n'}_1) \\ &\quad \tau_1(i) = \sum_{j=1}^{n'} j \hat{P}(S_{ij}|o_1^n; o'^{n'}_1) \\ &\forall i \leq n' \\ &\quad c_2(i) = \sum_{j=1}^{n} o_j \hat{P}(S_{ij}|o_1^n; o'^{n'}_1) \\ &\quad \tau_2(i) = \sum_{j=1}^{n} j \hat{P}(S_{ij}|o_1^n; o'^{n'}_1) \\ &\forall i \leq \texttt{min}(n, n') \\ &\quad c(i) = \frac{1}{1+\jmath_{i \leq \texttt{min}(n,n')}} (c_1(i) + c_2(i)) \\ &\quad \tau(i) = \frac{1}{1+\jmath_{i \leq \texttt{min}(n,n')}} (\tau_1(i) + \tau_2(i)) \end{aligned} \quad (15)$$

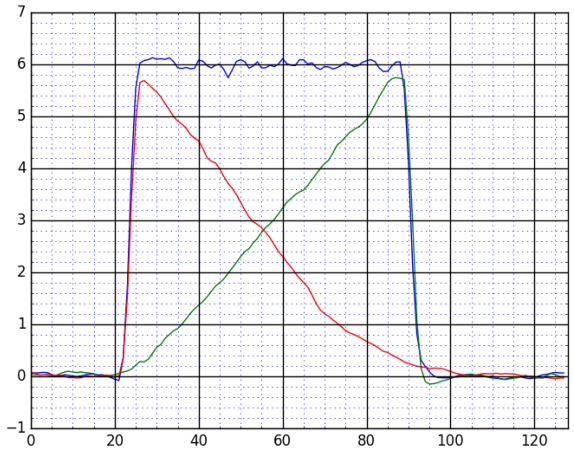


Fig. 3. Centroids obtained for the CBF data set. For the three shapes, the expected start (24) and end (88) time stamps (hence the expected shape duration of 64 frames) are correctly extracted

where $\mathbb{I}_{i\leq \min(n,n')}$ is equal to 1 if $i \leq \min(n, n')$, 0 otherwise.

To average a set of time series, the pairwise averaging principle described by equation (15) can be used to design a pairwise progressive agglomeration algorithm, as depicted in figure (1-a). We do not develop furthermore this algorithm and adopt instead an iterative approach that is described hereinafter.

3.5 Time elastic centroid estimate of a set of time series

The normalized posterior marginals, equation (14), are also at the basis of our procedure for averaging a set of time series.

Similarly to the pairwise averaging described in equation (15), let $O = \{{}^k o_1^n\}_{k=1\cdots N}$ be a set of time series and o_1^n a reference time series (o_1^n can be initially setup as the medoid of set O). The centroid estimate of O is defined as the pair (c, τ) where c is a time series of length n and τ is the sequence of time stamps associated to the samples of c

$$\begin{aligned} c(i) &= \frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{n_k} {}^k o_j \hat{P}(S_{ij}|o_1^n; {}^k o_1^n) \\ \tau(i) &= \frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{n_k} j \hat{P}(S_{ij}|o_1^n; {}^k o_1^n) \end{aligned} \quad (16)$$

Obviously, (c, τ) is a non uniformly sampled time series for which $\tau(i)$ is the time of occurrence of observation $c(i)$. $\tau(i)$ could be understood as the expected time of occurrence of the expected observation $c(i)$. A uniform re-sampling can straightforwardly be used to get back to a uniformly sampled time series.

The proposed iterative agglomerative algorithm (cf. Fig. 1-b), called TEKA (Time Elastic Kernel Averaging), that provides a refinement of the centroid estimation at each iteration until reaching a (local) optimum is presented in algorithm (1).

As an example, figure (3) presents the time elastic centroid estimates obtained, using algorithm (1) with $K = K_{dtw}$, for the Cylinder Bell Funnel synthetic data set [30]. The three shapes are characterized with a starting time stamp uniformly drawn in $[16; 32]$ and an end time stamp uniformly drawn in $[32; 96]$, which leads to expected start and end time stamps of 24 and 88 respectively, and a shape duration of 64 samples. Figure (3) clearly shows that, from a subset of 300 time series (100 for each category), the algorithm has correctly recovered the start and end shape events (hence the expected shape duration) for all three shapes.

Algorithm 1 Iterative Time Elastic Kernel Averaging (TEKA) of a set of time series

Let K be a similarity time elastic kernel for time series satisfying eq. (12)
Let O be a set of time series of d dimensional samples
Let c be an initial centroid estimate (e.g. the medoid of O) of length n
Let τ and τ_0 be two sequences of time stamps of length n initialized with zero values
Let $MeanK_0 = 0$ and $MeanK$ be two double values;
repeat
    $c_0 = c$, $\tau_0 = \tau$, $MeanK_0 = MeanK$;
    Evaluate c and τ according to Eq. (16)
    //Average similarity between c and O elements
    $MeanK = \frac{1}{|O|}\sum_{o\in O} K(c, o)$
until $MeanK > MeanK_0$
(c_0, τ_0) is the centroid estimation
Uniformly re-sample c_0 using the time stamps τ_0

The figures presented in Table 1 compare the centroid estimates provided by the iterated DBA [19], CTW [20] and TEKA algorithms. For the experiment, the DBA and TEK algorithms were iterated at most 10 times. The centroid estimates provided by the TEKA algorithm are much smoother than the ones provided by DBA or CTW. This denoisng property, expected from any averaging algorithm, will be address in a dedicated experiment (c.f. subsection 4.3).

3.6 Role of parameter ν

In practice, the selectivity or bandwidth of the local alignment kernels (that is controlled by parameter ν) has to be adapted according to the the lengths of the time series. If the time series are long, then nu should be reduced to maintain the calculability of the forward-backward matrices, and the local selectivity decreases. Hence, more alignment paths are likely and more sample pairs participate to the calculation of the average such that local details are filtered out by the averaging. Conversely if the time series are short, ν can be increased, hence fewer sample pairs participate to the calculation of the average, and details can be preserved.

3.7 Computational complexity

TEKA has intrinsically the same algorithmic complexity than the DBA algorithm, basically $O(L^2)$ for each pairwise averaging, where L is the average length of the time series. Nevertheless, computationally speaking, TEKA algorithm is slightly more costly mainly because of two reasons:

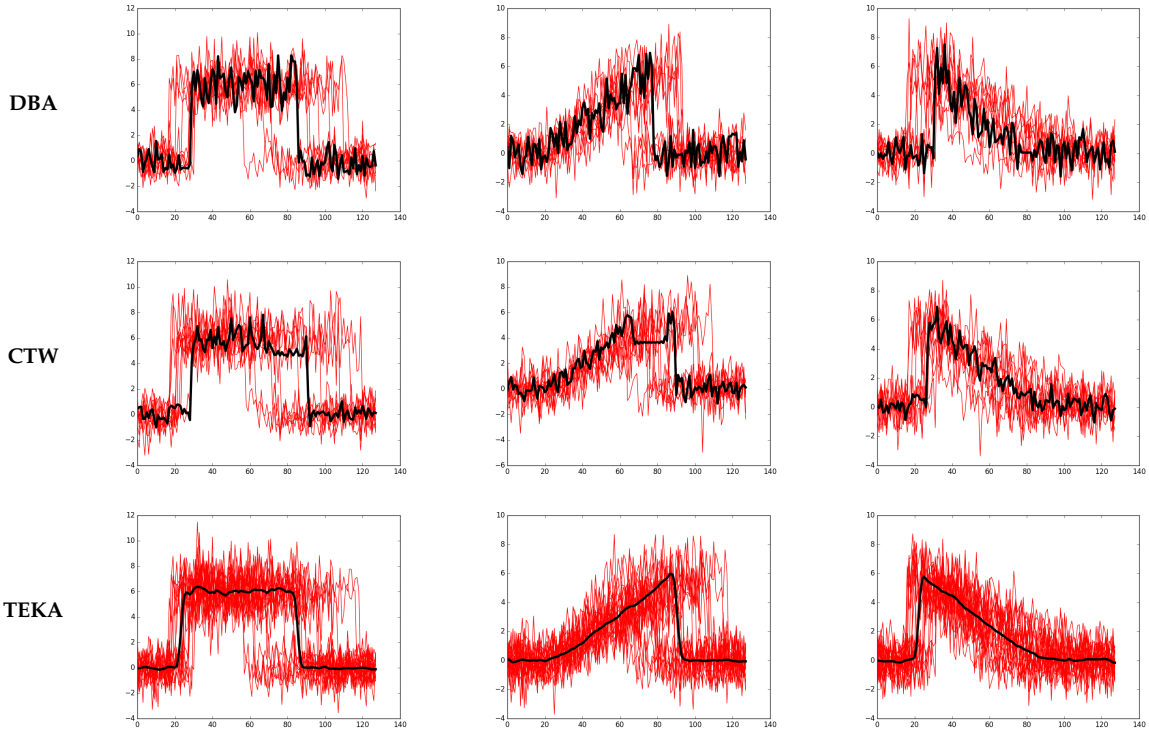


TABLE 1
Centroid estimation for the three categories of the CBF dataset and for the three tested algorithms: DBA (top), CTW (center) TEKA (bottom). The centroid estimations are indicated as a bold black line superimposed on top of the time series (in light red) that are averaged.

- the FB matrix induces a factor three in complexity because of the reverse alignment and the multiplication term by term of the forward and backward matrices.
- the exponential terms that enter into the computation of KDTW (Equation 4) are costly, basically $O(M(n)n^{1/2})$, where $M(n)$ is the cost of the floating point multiplication, and n is the number of digits. This induces another factor 2 or 3 depending on the chosen floating point precision.

The overall algorithmic cost for averaging a set of N time series of average length L with an average number of iterations I is, for the two algorithms, $O(I \cdot N \cdot L^2)$.

Some optimization are indeed possible, in particular replacing the exponential function by another local kernel easier to compute is an important source of algorithmic simplification. We do not address further this issue in this paper and let it stands as a perspective.

4 Experiments

The two first proposed experiments aim at demonstrating the benefits of using time elastic centroids in a data reduction paradigm: 1NN classification for the first one, and isolated gesture recognition for the second one using 1NN and SVM classifiers in conjunction with the KDTW kernel. The third experiment explores the noise reduction angle brought by time elastic centroids.

4.1 1-NN classification

The purpose of this experiment is to evaluate the effectiveness of the proposed time elastic averaging method (TEKA) against a triple baseline. The first baseline allow us to compare centroid-based with medoid-based approaches. The second and third baselines are provided by the DBA [19] and CTW [20] algorithms (thanks to the implementation proposed by the authors), currently considered as a state of the art method to average a set of sequences consistently with DTW. We have tested the CTW averaging with a 1NN-DTW (CTW1) and a 1NN-KDTW (CTW2) classifier to highlight the impact of the selected similarity measure.

For this purpose, we empirically evaluate the effectiveness of the methods using a first nearest centroid/medoid (1-NC) classification task on a set of time series derived from widely diverse fields of application. The task consists of representing each category contained in a training data set by estimating its medoid or centroid and then evaluating the error rate of a 1-NC classifier on an independent testing data set. Hence, the classification rule consists of assigning to the tested time series the category which corresponds to the closest (or most similar) medoid or centroid according to the DTW measure for DTW medoid, DBA and CTW centroids (CTW1) or to KDTW measure for KDTW medoid, CTW (CTW2) and TEKA centroids.

In [8] a generalized k-NC task is described. The authors demonstrate that by selecting the appropriate number k of centroids (using DBA and k-means), they achieve, without loss, a 70% speed-up in average, compared to the original k-

Near Neighbor task. Although, in general, the classification accuracies is improved when several centroids are used to represent the TRAIN datasets, our main purpose is to highlight and amplify the discrimination between time series averaging methods: this is why we stick here with the 1-NC task.

A collection of 45 heterogeneous data sets is used to assess the proposed algorithms. The collection includes synthetic and real data sets, as well as univariate and multivariate time series. These data sets are distributed as follows:

- 42 of these data sets are available at the UCR repository [31]. Basically, we used all the data sets except for *StarLightCurves*, *Non-Invasive Fetal ECG Thorax1* and *Non-Invasive Fetal ECG Thorax2*. Although these last three data sets are still tractable, their computational cost is high because of their size and the length of the time series they contain. All these data sets are composed of scalar time series.
- One data set, uWaveGestureLibrary_3D was constructed from the uWaveGestureLibrary_X—Y—Z scalar data sets to compose a new set of multivariate (3D) time series.
- One data set, CharTrajTT, is available at the UCI Repository [32] under the name *Character Trajectories Data Set*. This data set contains multivariate (3D) time series and is divided into two equal sized data sets (TRAIN and TEST) for the experiment.
- The last data set, *PWM2*, which stands for Pulse Width Modulation [33], was specifically defined to demonstrate a weakness in dynamic time warping (DTW) pseudo distance. This data set is composed of synthetic scalar time series.

For each dataset, a training subset (TRAIN) is defined as well as an independent testing subset (TEST). We use the training sets to extract single medoids or centroid estimates for each of the categories defined in the data sets.

Furthermore, for KDTW$_{Medoid}$, CTW2 and TEKA, the ν parameter is optimized using a *leave-one-out* (LOO) procedure carried out on the TRAIN data sets. The ν value is selected within the discrete set $\{.01, .05, .1, .25, .5, .75, 1, 2, 5, 10, 15, 20, 25, 50, 100\}$. The value that minimizes the LOO classification error rate on the TRAIN data is then used to provide the error rates that are estimated on the TEST data.

The classification results are given in Table 2. It can be seen from this experiment, that

i) Centroid-based methods outperform medoid-based methods: DBA and CTW (CTW2) yield lower error rates compared to DTW$_{Medoid}$, as do TEKA compared to KDTW$_{Medoid}$ and DTW$_{Medoid}$.
ii) CTW pairs much better with KDTW (CTW2 outperforms CTW1)
iii) TEKA outperforms DBA (under the same experimental conditions (maximum of 10 iterations)), and CTW.

The average ranking for all four tested methods, which supports our preliminary conclusion, is given at the bottom of Table 2.

In table 3 we report the P-values for each pair of tested algorithms using a Wilcoxon signed-rank test. The null hypothesis is that for a tested pair of classifiers, the difference between the pairs of classification error rates obtained on the 45 datasets follows a symmetric distribution around zero. With a .05 significance level, the P-values that lead to reject the null hypothesis are showed in bolded fonts in the table. This analysis confirms our previous analysis of the classification results. We observe that centroid-based approaches perform significantly better than medoid-based approaches. Furthermore, KDTW$_{Medoid}$ appears to be significantly better than DTW$_{Medoids}$.

Furthermore, TEKA is evaluated as significantly better than DBA and CTW2 in this experiment. Note also that DBA does not seem to perform significantly better than KDTW$_{Med}$ or CTW2, and that CTW1 performed similarly to DTW$_{Med}$ and poorly compared to the other centroid methods. Hence, it turns out that CTW method seems to pair well with KDTW measure but poorly with the DTW measure.

4.2 Instance set reduction

In this second experiment, we address an application that consists in summarizing subsets of training time series to speed-up an isolated gesture recognition process.

The dataset that we consider enables to explore the handshape and the upper body movement using 3D positions of skeletal joints captured using a Microsoft Kinect 2 sensor. 20 subjects have been selected (15 males and 5 females) to perform in front of the sensor (at a three meters distance) the six selected NATOPS gestures. Each subject repeated each gesture three times. Hence the isolated gesture dataset is composed of 360 gesture utterances that have been manually segmented to a fixed length of 51 frames (1.7 sec. duration). [1].

To evaluate this task, we have performed a subject cross validation experiment consisting of 100 tests: for each test, 10 subjects have been randomly drawn among 20 for training and the remaining 10 subjects have been retained for testing. 1-NN (our baseline) and SVM classifiers are evaluated, with or without summarizing the subsets composed with the three repetitions performed by each subjects using a single centroid (DBA, CTW, TEKA) or Medoid (KDTW-M). The ν parameter of the KDTW kernel as well as the SVM meta parameter (RBF bandwidth σ and C) are optimized using a leave one subject procedure on the training dataset. $exp(-DTW(.,.)/\sigma)$ and $exp(-KDTW(.,.)/\sigma)$ are used respectively in the SVM DTW and SVM KDTW classifiers.

Table 4 gives the assessment measures (ERR: average error rate, PRE: macro average precision, REC: macro average recall and $F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$) for the isolated gestures classification task. In addition, the number of reference

1. These datasets will be made available for the community at the earliest feasible opportunity

TABLE 2
Comparative study using the UCR and UCI data sets: classification error rates evaluated on the TEST data set (in %) obtained using the first nearest neighbour classification rule for DTW$_{Medoid}$, DBA (centroid), KDTW$_{Medoid}$, CTW1, CTW2 and TEKA (centroids). A single medoid/centroid extracted from the TRAIN data set represents each category.

DATASET	# Cat | L	**DTW$_{Med}$**	**DBA**	**CTW1**	**CTW2**	**KDTW$_{Med}$**	**TEKA**
Synthetic_Control	6|60	3.00	**2.00**	19.00	3.33	3.33	2.33
Gun_Point	2|150	44.00	32.00	54.67	**25.33**	52.00	27.33
CBF	3|128	7.89	5.33	34.22	3.55	8.11	**3.33**
Face_(all)	14|131	25.21	18.05	34.38	27.93	20.53	**13.61**
OSU_Leaf	6|427	64.05	56.20	64.05	57.02	53.31	**50.82**
Swedish_Leaf	15|128	38.56	30.08	32	25.76	31.36	**22.08**
50Words	50|270	48.13	41.32	48.57	36.48	23.40	**19.78**
Trace	4|275	**5.00**	7.00	6.00	18	23.00	16.00
Two_Patterns	4|128	1.83	1.18	26.75	37.75	1.17	**1.10**
Wafer	2|152	64.23	33.89	37.83	33.27	43.92	**8.38**
Face_(four)	4|350	12.50	13.64	19.32	15.91	17.05	**10.23**
Lightning-2	2|637	34.43	37.70	37.70	**29.51**	**29.51**	**29.51**
Lightning-7	7|319	27.40	27.40	41.10	38.35	19.18	**16.44**
ECG200	2|96	32.00	28.00	27.00	**25**	29.00	26.00
Adiac	37|176	57.54	52.69	54.73	34.78	40.67	**32.22**
Yoga	2|426	47.67	47.87	53.56	48.97	47.53	**44.90**
Fish	7|463	38.86	30.29	39.42	22.28	20.57	**14.28**
Beef	5|470	60.00	53.33	53.33	**50**	53.33	**50**
Coffee	2|286	57.14	32.14	32.14	**28.57**	32.14	32.14
OliveOil	4|570	26.67	**16.67**	13.33	23.33	30	**16.67**
CinC_ECG_torso	4|1639	74.71	53.55	73.33	42.90	66.67	**33.04**
ChlorineConcentration	3|166	65.96	68.15	67.40	67.97	65.65	**64.97**
DiatomSizeReduction	4|345	22.88	5.88	5.23	**2.61**	11.11	2.94
ECGFiveDays	2|136	47.50	30.20	34.49	13.47	**11.38**	16.37
FacesUCR	14|131	27.95	18.44	32.20	21.66	20.73	**12.19**
Haptics	5|1092	68.18	64.61	58.77	57.47	63.64	**53.57**
InlineSkate	7|1882	78.55	76.55	81.64	82.18	78.36	**75.09**
ItalyPowerDemand	2|24	31.68	20.99	15.84	9.33	**5.05**	6.61
MALLAT	8|1024	6.95	6.10	5.24	**3.33**	6.87	3.66
MedicalImages	10|99	67.76	58.42	58.29	59.34	**57.24**	59.60
MoteStrain	2|84	15.10	13.18	19.01	15.33	12.70	**9.35**
SonyAIBORobot_SurfaceII	2|65	26.34	21.09	20.57	**17.52**	26.230	19.30
SonyAIBORobot_Surface	2|70	38.10	19.47	14.48	**9.31**	39.77	17.95
Symbols	6|398	7.64	4.42	22.31	20.70	**3.92**	4.02
TwoLeadECG	2|82	24.14	**13.17**	20.37	19.23	27.04	18.96
WordsSynonyms	25|270	70.85	64.26	78.84	63.32	64.26	**56.11**
Cricket_X	12|300	67.69	**52.82**	78.46	73.85	61.79	**52.82**
Cricket_Y	12|300	68.97	52.82	69.74	65.64	**46.92**	50.25
Cricket_Z	12|300	73.59	**48.97**	78.21	64.36	56.67	51.79
uWaveGestureLibrary_X	8|315	38.97	33.08	37.33	34.61	34.34	**32.18**
uWaveGestureLibrary_Y	8|315	49.30	44.44	45.42	41.99	42.18	**39.64**
uWaveGestureLibrary_Z	8|315	47.40	**39.25**	47.65	39.36	41.96	39.97
PWM2	3|128	43.00	35.00	63.66	6.33	21.00	**4.33**
uWaveGestureLibrary_3D	8|315	10.11	**5.61**	9.35	7.68	13.74	7.73
CharTrajTT_3D	20|178	11.026	9.58	13.45	15.05	6.93	**4.99**
# Best Scores	-	1	7	0	9	6	**27**
# Uniquely Best Scores	-	1	5	0	7	5	**23**
Average rank	-	4.56	2.87	4.62	2.97	3.22	**1.6**

TABLE 3
Wilcoxon signed-rank test of the differences between pairs of 1NN classifiers carried out on the 45 datasets.

Method	KDTW	DBA	CTW1	CTW2	TEKA
DTW	**6.99e-06**	**9.64e-08**	0.638	**2.35e-04**	**2.62e-08**
KDTW	-	0.395	**4.86e-04**	0.5261	**5.08e-07**
DBA	-	-	**2.29e-06**	0.8214	**1.02e-05**
CTW1	-	-	-	**3.51e-07**	**7.47e-08**
CTW2	-	-	-	-	**2.25e-05**

TABLE 4
Assessment measures (ERR:Error rate, PRE: Precision, REC:Recall and $\mathbf{F_1}$ score) for the isolated gestures recognition. $\overline{\#Ref}$ is the number of training gestures for the 1NN classifiers and the mean number of support vectors for the SVM classifiers.

Method	**ERR** mean ‖ std	**PRE**	**REC**	$\mathbf{F_1}$	$\overline{\#Ref}$
1NN DTW	.134 ‖ .012	.869	.866	0.867	180
1NN KDTW	**.128** ‖ .016	.876	.972	.874	180
1NN DTW-DBA	.136 ‖ .014	.868	.864	.866	**60**
1NN KDTW-CTW	.135 ‖ .016	.871	.865	.868	**60**
1NN KDTW-TEKA	**.133** ‖ .014	.871	.867	.869	**60**
SVM DTW	.146 ‖ .015	.871	.854	.862	164.97
SVM KDTW	**.051** ‖ .015	.952	.949	.951	**103.10**
SVM KDTW-M	**.087** ‖ .02	.92.9	.92.6	.92.7	47.62
SVM KDTW-DBA	.080 ‖ .017	.935	.931	.931	**46.74**
SVM KDTW-CTW	.085 ‖ .021	.933	.927	.930	50.12
SVM KDTW-TEKA	**.079** ‖ .019	.937	.933	.935	47.45

TABLE 5
Wilcoxon signed-rank test of the differences between pairs of 1NN classifiers. DTW and KDTW methods exploit the entire training sets while the other methods only use one centroid for each subject and each gesture label, which corresponds to a reduction in the size of the learning set by a factor of 3.

Method	1NN DTW	1NN KDTW	1NN DBA	1NN CTW	1NN TEKA
1NN DTW	-	**5.438e-06**	0.140	0.886	0.371
1NN KDTW	-	-	**9.673e-05**	**0.026**	0.087
1NN DBA	-	-	-	0.281	**0.006**
1NN CTW	-	-	-	-	0.199

TABLE 6
Wilcoxon signed-rank test of the differences between pairs of SVM classifiers. DTW and KDTW methods exploit the entire training sets while the other methods only use one centroid for each subject and each gesture label, which corresponds to a reduction in the size of the learning set by a factor of 3.

Method	SVM DTW	SVM KDTW	SVM KDTW-M	SVM DBA	SVM CTW	SVM TEKA
SVM DTW	-	**2.2e-16**	**2.2e-16**	**2.2e-16**	**2.2e-16**	**2.2e-16**
SVM KDTW	-	-	**9.68e-16**	**8.81e-15**	**7.96e-16**	**2.0e-15**
SVM KDTW-M	-	-	-	**0.002**	0.57	**0.0002**
SVM DBA	-	-	-	-	0.107	0.339
SVM CTW	-	-	-	-	-	**0.013**

instances used by the 1NN classifiers or the number of support vectors exploited by the SVM ($\overline{\#Ref}$ column in the table) are reported to demonstrate the data reduction that is induced by the methods in the training sets.

The results show that the DTW measure does not fit well with SVM comparatively to KDTW: the error rate or the F1 score are about 9% higher or lower for the isolated gesture task. Hence, to compare the DBA, CTW and TEKA centroids using a SVM classification, the KDTW kernel as been used. When using the centroids (SVM KDTW-DBA, SVM KDTW-CTW, SVM KDTW-TEKA), or Medoids (SVM KDTW-M) the error rate or F_1 score increases or decreases only by around 2.5% and 2% comparatively to the SVM-KDTW that achieves the best scores. Meanwhile the number of support vectors exploited by the SVM drops by a two factor, leading to an expected speed-up of 2. Compared to 1NN classification without centroids, the SVM KDTW with centroids achieves a much better performance, with an expected speed-up of 4 (~ 50 support vectors comparatively to 180 gesture instances). This demonstrates the capacity of centroid methods to reduce significantly the size of the training sets while maintaining a very similar level of accuracy.

In more details, the TEKA is the centroid-based method that achieves the lowest error rates for the two classification tasks, while DBA is the centroid-based method that exploits the fewest support vectors (46.5).

Table 5 and 6 give the P-values for the Wilcoxon signed-rank tests. With the same null hypothesis as above (difference between the pairs of error rates follows a symmetric distribution around zero), and with a $.05$ significance level, the P-values that lead to reject the null hypothesis are presented in bolded fonts in the tables. From Table 5 we note that 1NN-KDTW (which exploits the full training set) performs significantly better than 1NN DTW, 1NN DTW-DBA and 1NN KDTW-CTW but not significantly than 1NN KDTW-TEKA. Conversely, 1NN KDTW-TEKA performs significantly better that 1NN DTW-DBA but not significantly better that 1NN KDTW-CTW. Similarly, from Table 6 we observe that SVM KDTW, which exploits the full training set, performs significantly better than all centroid or medoid based methods. Also, SVM KDTW-TEKA performs significantly better than SVM KDTW-CTW but not significantly better than SVM KDTW-DBA. Finally SVM KDTW-TEKA and SVM KDTW-DBA outperform the medoid based method (SVM KDTW-M) but not SVM KDTW-CTW.

If the three centroid methods are rather close on this experiment, TEKA shows to be slightly more robust.

4.3 Denoising experiment

To demonstrate the utility of TEKA for denoising data, we construct a demonstrative synthetic experiment that provides some insights. The test is based on the following 2D periodic signal:

$$X_k(t) = \left(A_k + B_k \sum_{i=1}^{\infty} \delta(t - \frac{2\pi i}{6\omega_k}) \right) cos(\omega_k t + \phi_k) \quad (17)$$
$$Y_k(t) = \left(A_k + B_k \sum_{i=1}^{\infty} \delta(t - \frac{2\pi i}{6\omega_k}) \right) sin(\omega_k t + \phi_k)$$

where $A_k = A_0 + a_k$, $B_k = (A_0 + 5) + b_k$ and $\omega_k = \omega_0 + w_k$, A_0 and ω_0 are constant and a_k, b_k, ω_k, ϕ_k are small perturbation in amplitude, frequency and phase respectively and randomly drawn from $a_k \in [0, A_0/10]$, $b_k \in [0, A_0/10]$, $\omega_k \in [-\omega_0/6.67, \omega_0/6.67]$, $\phi_k \in [-\omega_0/10, \omega_0/10]$.

In practice we have adopted the following setting: $f_0 = \omega_o/(2.\pi) = 20Hz$, and $A_0 = 1$. We then center and normalize this 2D signal to get $(\tilde{X}_k(t), \tilde{Y}_k(t))$ corresponding to the plots given in Figure 4. The log power spectrum of

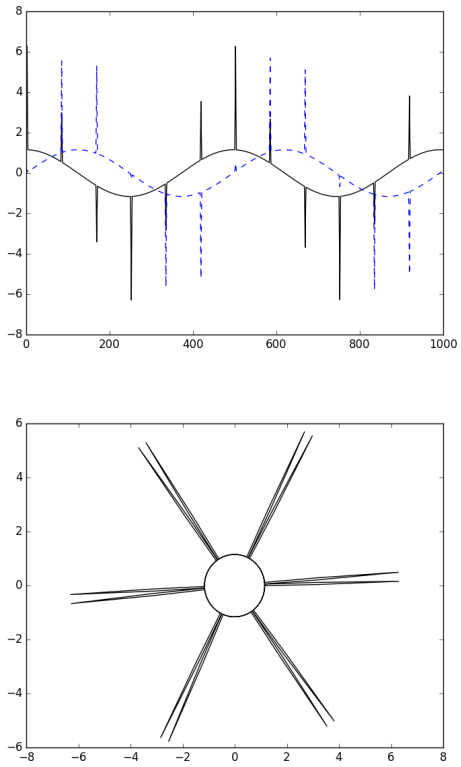


Fig. 4. $(\tilde{X_k}(t), \tilde{Y_k}(t))$ waveforms (top) and corresponding 2D shape (bottom, plain black curve) of the synthetic signal. Three instances of the shapes are also given in dotted lines, showing the frequency, phase and amplitude uncertainties.

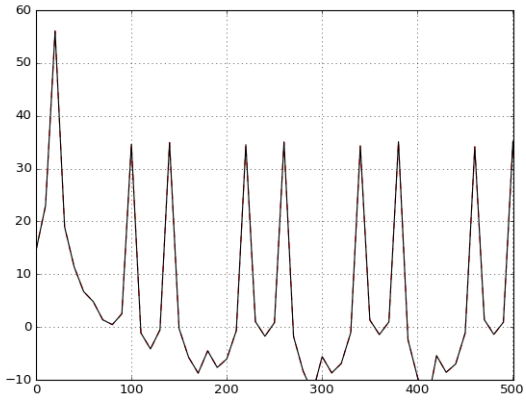


Fig. 5. Log power spectra of a $\tilde{X_k}$ component.

the $\tilde{X_k}$ component, that is presented in Figure 5, shows the Dirac spike located at $f_0 = 20Hz$ (corresponding to the sine component), and the convolution of this spike with a Dirac comb in the frequency domain that results in pairs of Dirac spikes symmetrically located ($\pm 20Hz$) around multiples of $6f_0$, namely $120Hz$, $240Hz$, etc. This shows that this signal is characterized by an infinite spectrum.

We consider then a noise $\epsilon(t)$ with zero mean and

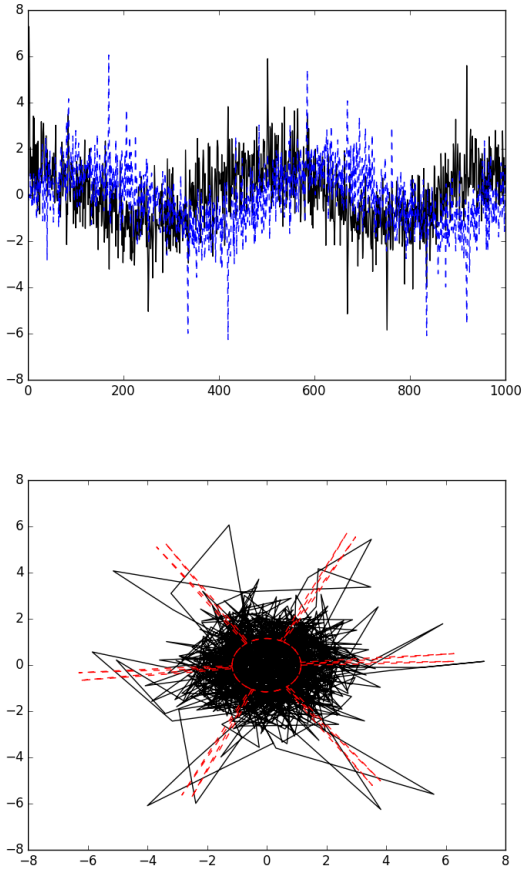


Fig. 6. Noisy $(x_k(t), y_k(t))$ waveforms (top) and corresponding 2D shape (bottom) of the synthetic signal.

variance one added to each instances of the 2D signal:

$$x_k(t) = \tilde{X_k}(t) + \epsilon(t)$$
$$y_k(t) = \tilde{Y_k}(t) + \epsilon(t)$$

leading to a signal to noise ratio of $0dB$. An example of such noisy instance is given in Figure 6. Because of the scattering of the random components of the signal in a wide spectral band, traditional noise reduction techniques, such as those presented in [5] for instance, will not allow to recover the signal properly.

The task consists in reducing the noise as far as possible to recover the 2D shape of the noise free signal from a small set of noisy instances $\{(x_k, y_k)\}_{k=1\cdots 8}$ containing a single period of the clean signal. Figure 7 presents the centroid shapes obtained using, from left to right, Euclidean, DBA, CTW and TEKA methods respectively. We can see that the Euclidean centroid retrieves partially the low frequency sine component without properly sorting out the spikes components, while DBA more accurately retrieves the spikes, however without achieving to suppress the low frequency noise around the sine component. CTW centroid appears to be in between and achieves partially to reduce the low frequency noise and to extract the spikes. TEKA achieves the best retrieval of the sine and spikes components that are better timely and spatially separated. The spectral analysis presented in Figure 7 (top) gives further insight: for DBA

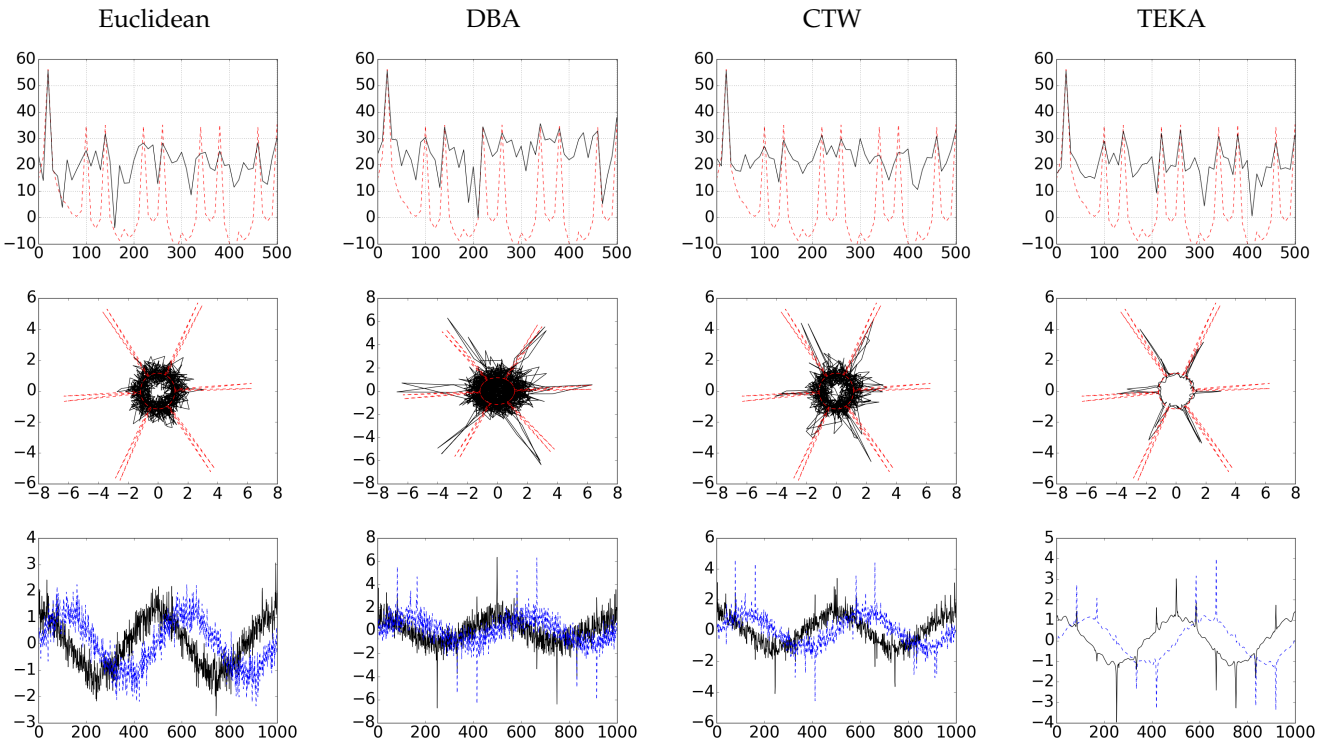


Fig. 7. Centroids obtained from a set of height noisy instances $\{(x_k, y_k)\}_{k=1\cdots8}$ for Euclidean, DBA, CTW and TEKA averaging methods. The log power spectra in dB (top), the 2D shape (center) and x,y waveforms (bottom) are shown.

and CTW centroids, top center sub-figures, the series of pairs of Dirac spikes (in dotted red) are still hidden into the noise level (black curve), while it is much more separated from the noise for the TEKA centroid, as shown in the top right side sub-figure.

Moreover, if we take the clean shapes as ground truth, the signal to noise ratio (SNR) gains estimated from the log power spectra (to get rid of the phase) is $0dB$ for the noisy shapes , while it is $1.58dB$ for the Euclidean centroid, $1.17dB$ for the DBA centroid, $1.57dB$ for the CTW centroid, and $3.88dB$ for the TEKA centroid. Note that in the calculation of the SNR, preserving the spikes as a lower impact compared to preserving the low frequency sine wave, which explains why the SNR values obtained by the DBA and CTW centroid are lower than for the Euclidean centroid.

In terms of noise reduction, this experiment demonstrates the ability of the TEKA centroid to better recover, from few noisy utterances, a signal whose components are scattered in a wide band spectrum. Indeed, if the noise level increases, the quality of the denoising will be reduced.

4.4 Discussion

We believe that the noise filtering ability of TEKA is mainly due to the averaging technique described in the equation (16), which aggregates many plausible alignments between samples (instead of a best one) while averaging also the time of occurrence of the samples, in particular those corresponding to expected pattern location and duration such as the CBF shapes or the spike locations in the third experiment. This ability is also likely to explain the best accuracy results obtained by TEKA comparatively to the state of the art methods, CTW and DBA.

Furthermore, it seems that the KDTW measure is more adapted to match centroids than DTW. Here again, handling several good to best alignments rather than a single optimal one allows for matching the centroids in many ways that are averaged by the measure. This has been verified for CTW in 1NN classification tasks and is true for TEKA and DBA also.

The main limitation in exploiting TEKA (and KDTW) is the tuning of the ν parameter that control the selectivity of the local kernel. ν is dependent on the length of the time series and need to be adapted to the task itself. Basically, if ν is too small TEKA will filter out high frequency events just as a moving average filter. Conversely, if ν is too high, the computation of the products of local probabilities along the alignment paths will bear some loss of significance in terms of the numerical calculation. Despite this tuning requirement, the three experiments, that we have carried out in this study, demonstrate its applicability and usefulness.

5 Conclusion

In this paper, we have addressed the problem of averaging a set of time series in the context of a time elastic distance measure such as Dynamic Time Warping. The new perspective provided by the kernelization of the elastic distance allows a re-interpretation of pairwise kernel alignment matrices as the result of a forward-backward procedure applied on the states of an equivalent stochastic alignment automata. From this re-interpretation, we have proposed a new algorithm, TEKA, based on an iterative agglomerative heuristic method that allows for efficiently computing good solutions to the multi-alignment of time series. This algorithm exhibits quite interesting denoising capabilities which enlarges the area of its potential applications.

We have presented extensive experiments carried out on synthetic and real data sets, containing univariate but also multivariate time series. Our results show that centroid-based methods significantly outperform medoid-based methods in the context of a first nearest neighbor and SVM classification tasks. More strikingly, the TEKA algorithm, which integrates joint averaging in the sample space and along the time axis, is significantly better than the state-of-the art DBA and CTW algorithms, with a similar algorithmic complexity. It enables robust training set reduction which has been experimented on an isolated gesture recognition task. Finally we have developed a dedicated synthetic test to demonstrate the denoising capability of our algorithm, a property that is not supported at a same level by the other time-elastic centroid methods on this test.

ACKNOWLEDGMENTS

The authors thank the French Ministry of Research, the Brittany Region and the European Regional Development Fund that partially funded this research. The authors also thank the promoters of the UCR and UCI data repositories for providing the datasets used in this study.

REFERENCES

[1] M. Fréchet, *Sur quelques points du calcul fonctionnel*, ., Ed. Thèse, Faculté des sciences de Paris., 1906.
[2] V. M. Velichko and N. G. Zagoruyko, "Automatic recognition of 200 words," *Int. Jour. of Man-Machine Studies*, vol. 2, pp. 223–234, 1970.
[3] H. Sakoe and S. Chiba, "A dynamic programming approach to continuous speech recognition," in *Proceedings of the 7th International Congress of Acoustic*, 1971, pp. 65–68.
[4] R. Kaiser and W. Knight, "Digital signal averaging," *Journal of Magnetic Resonance (1969)*, vol. 36, no. 2, pp. 215 – 220, 1979.
[5] U. Hassan and M. S. Anwar, "Reducing noise by repetition: introduction to signal averaging," *European Journal of Physics*, vol. 31, no. 3, p. 453, 2010.
[6] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recogn.*, vol. 44, no. 3, pp. 678–693, Mar. 2011.
[7] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, Sept 2014.
[8] F. Petitjean, G. Forestier, G. Webb, A. Nicholson, Y. Chen, and E. Keogh, "Dynamic time warping averaging of time series allows faster and more accurate classification," in *Proceedings of the 14th IEEE International Conference on Data Mining*, 2014, pp. 470–479.
[9] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, ser. VLDB '04. VLDB Endowment, September 2004, pp. 792–803.
[10] P.-F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 306–318, Feb 2009.
[11] L. Gupta, D. Molfese, R. Tammana, and P. Simos, "Nonlinear alignment and averaging for estimating the evoked potential," *Biomedical Eng., IEEE Trans. on*, vol. 43, no. 4, pp. 348–356, 1996.
[12] K. H. Fasman and S. S. L., "An introduction to biological sequence analysis," in *Computational Methods in Molecular Biology,*. In Salzberg, S.L., Searls, D.B., and Kasif, S., eds., Elsevier, 1998, pp. 21–42.
[13] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment." *Jour. of Comp. Biology*, vol. 1, no. 4, pp. 337–348, 1994.
[14] W. Just and W. Just, "Computational complexity of multiple sequence alignment with sp-score," *Journal of Computational Biology*, vol. 8, pp. 615–623, 1999.
[15] H. Carrillo and D. Lipman, "The multiple sequence alignment problem in biology," *SIAM J. Appl. Math.*, vol. 48, no. 5, pp. 1073–1082, Oct. 1988.
[16] V. Niennattrakul and C. Ratanamahatana, "Shape averaging under time warping," in *Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th Int. Conf. on*, vol. 02, May 2009, pp. 626–629.
[17] W. Abdulla, D. Chow, and G. Sin, "Cross-words reference template for dtw-based speech recognition systems," in *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*, vol. 4, Oct 2003, pp. 1576–1579 Vol.4.
[18] V. Hautamaki, P. Nykanen, and P. Franti, "Time-series clustering by approximate prototypes," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, Dec 2008, pp. 1–4.
[19] F. Petitjean and P. Gançarski, "Summarizing a set of time series by averaging: From Steiner sequence to compact multiple alignment," *Journal of theoretical computer science*, vol. 414, no. 1, pp. 76–91, Jan. 2012.
[20] F. Zhou and F. Torre, "Canonical time warping for alignment of human behavior," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 2286–2294.
[21] F. Zhou and F. D. la Torre, "Generalized canonical time warping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 279–294, Feb 2016.
[22] S. Soheily-Khah, A. Douzal-Chouakria, and E. Gaussier, "Generalized k-means-based clustering for temporal data under weighted and kernel time warp," *Patt. Recog. Lett.*, vol. 75, pp. 63 – 69, 2016.
[23] V. Niennattrakul and C. Ratanamahatana, "Inaccuracies of shape averaging method using dynamic time warping for time series data," in *Computational Science – ICCS 2007*, ser. Lecture Notes in Computer Science, Y. Shi, G. van Albada, J. Dongarra, and P. Sloot, Eds. Springer Berlin Heidelberg, 2007, vol. 4487, pp. 513–520.
[24] P.-F. Marteau and S. Gibet, "On Recursive Edit Distance Kernels with Application to Time Series Classification," *IEEE Trans. on Neural Networks and Learning Systems*, pp. 1–14, Jun. 2014.
[25] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui, "A kernel for time series based on global alignments," in *IEEE ICASSP 2007*, vol. 2, April 2007, pp. II–413–II–416.
[26] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.
[27] B. Juang, "On the hidden Markov model and dynamic time warping for speech recognition – A unified view," *AT&T Bell Labs Technical Jour.*, vol. 63, no. 7, pp. 1213–1242, 1985.
[28] S. Nakagawa and H. Nakanishi, "Speaker-Independent English Consonant and Japanese Word Recognition by a Stochastic Dynamic Time Warping Method," *Journal of Institution of Electronics and Telecommunication Engineers*, vol. 34, no. 1, pp. 87–95, Jan. 1989.
[29] D. Chudova, S. Gaffney, and P. Smyth, "Probabilistic models for joint clustering and time-warping of multidimensional curves," in *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 134–141. [Online]. Available: http://dl.acm.org/citation.cfm?id=2100584.2100600
[30] N. Saito, "Local feature extraction and its applications using a library of bases," Ph.D. dissertation, Dept. of Mathematics, Yale University, 1994.
[31] E. J. Keogh, X. Xi, L. Wei, and C. Ratanamahatana, "The UCR time series classification-clustering datasets," 2006, http://wwwcs.ucr.edu/ eamonn/time_series_data/.
[32] M. Lichman, "Uci machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml
[33] P.-F. Marteau, "Pulse width modulation data sets," 2007. [Online]. Available: http://people.irisa.fr/Pierre-Francois.Marteau/PWM/