Times series averaging from a probabilistic interpretation of time-elastic kernel

Pierre-François Marteau

Times series averaging from a probabilistic interpretation of time-elastic kernel

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

Abstract—At the light of regularized dynamic time warping kernels, this paper re-consider the concept of time elastic centroid (TEC) for a set of time series. From this perspective, we show first how TEC can easily be addressed as a preimage problem. Unfortunately this preimage problem is ill-posed, may suffer from over-fitting especially for long time series and getting a sub-optimal solution involves heavy computational costs. We then derive two new algorithms based on a probabilistic interpretation of kernel alignment matrices that expresses in terms of probabilistic distributions over sets of alignment paths. The first algorithm is an iterative agglomerative heuristics inspired from the state of the art DTW barycenter averaging (DBA) algorithm proposed specifically for the Dynamic Time Warping measure. The second proposed algorithm achieves a classical averaging of the aligned samples but also implements an averaging of the time of occurrences of the aligned samples. It exploits a straightforward progressive agglomerative heuristics. An experimentation that compares for 45 time series datasets classification error rates obtained by first near neighbors classifiers exploiting a single medoid or centroid estimate to represent each categories show that: i) centroids based approaches significantly outperform medoids based approaches, ii) on the considered experience, the two proposed algorithms outperform the state of the art DBA algorithm, and iii) the second proposed algorithm that implements an averaging jointly in the sample space and along the time axes emerges as the most significantly robust time elastic averaging heuristic with an interesting noise reduction capability.

Index Terms—Time series averaging Time elastic kernel Dynamic Time Warping Time series clustering and classification.

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 of the scientific community in numerous domains such as information indexing and retrieval, pattern analysis, extraction and recognition, data mining, etc, impacting in a very large spectrum of applications relating to almost all the socio-economic areas such as environment, industry, health, energy, defense and so on.

Among other time elastic measures, Dynamic Time Warping (DTW) has been widely popularized during the seventies with the advent of speech recognition systems [2], [3] and a lot of variants have been proposed since to match time series with some time distortion tolerance.

The main issue we address in this paper is time series or shape averaging in the context of a time elastic distance. This is an old question that is becoming increasingly prevalent recently to summarize subsets of time series, define significant prototypes, identify outliers, perform data mining tasks (mainly exploratory data analysis such as clustering) or speed up classification, 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 the DTW measure itself as it has been already largely explored, but from the perspective of the so-called regularized DTW kernel (KDTW) that ensures positive definiteness. From this new perspective, the estimation of a time series average or centroid can be straightforwardly addressed as a preimage (inverse) problem, unfortunately with some theoretical and practical limitation that we will address in the following sections. A more promising direct approach, that we develop here, is based on a probabilistic interpretation of kernel alignment matrices, allowing for precisely defining the average of a pair of time series from the expected value of local alignments of samples. The experiment that we have carried out so far demonstrates the robustness and the efficiency of this approach comparatively to the state of the art approach.

The paper is organized as follows: the second section synthesizes most relevant related works on time series averaging as well as DTW kernelization. In the third section, we show how, in the scope of the DTW regularized kernel (KDTW), one can address the time elastic centroid question as a preimage problem. In the fourth section we derive a probabilistic interpretation from the kernel alignment matrices evaluated on a pair of time series. In the fifth section, we firstly define the average of a pair of time series, and secondly, from this pairwise averaging procedure, we propose two sub-optimal algorithms dedicated for the averaging of any subset of time series.

2 Related works

Time series averaging in the context of (multiple) time elastic distance alignments has been mainly addressed

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

in the scope of the Dynamic Time Warping (DTW) measure [2], [3]. Although any other time elastic distance such as the Edit Distance With Real Penalty (ERP) [4] or the Time Warp Edit Distance (TWED) [5] could be instead considered, without loss of generality, we will stay focused on DTW and its kernelization through out this paper.

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

A classical formulation of DTW is 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, a *alignment path* $\pi = (\pi_k)$ of length $p = |\pi|$ between u and u is 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 $\pi_k(2) = j_k$, 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 simultaneously along both time series from their beginnings to their ends; it cannot skip a point, but it can advance one time step along one series without advancing the other, thereby justifying the *time-warping* terminology.

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 induced by the L^2 norm:

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

Definition 2.2: For finite time series, 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 and finally $\text{DTW}(u, v)$ is defined as the minimal cost taken over all existing warping paths. Hence

$$\text{DTW}(u, v) = \min_{\pi} \text{cost}(\pi(u, v)) = \text{cost}(\pi^*(u, v)). \quad (1)$$

Definition 2.3: From the DTW measure, it is straightforwards to define the time elastic centroid $c(u, v)$ of a pair of time series u and v as the time series (c_k) whose elements are $c_k = \textit{Centroid}(u(\pi_k^*(1)), v(\pi_k^*(2))$, $\forall k \in 1, \cdots, |\pi^*|$, where *Centroid* stands for the usual definition in an Euclidean space.

2.2 Time elastic centroid of a set of time series

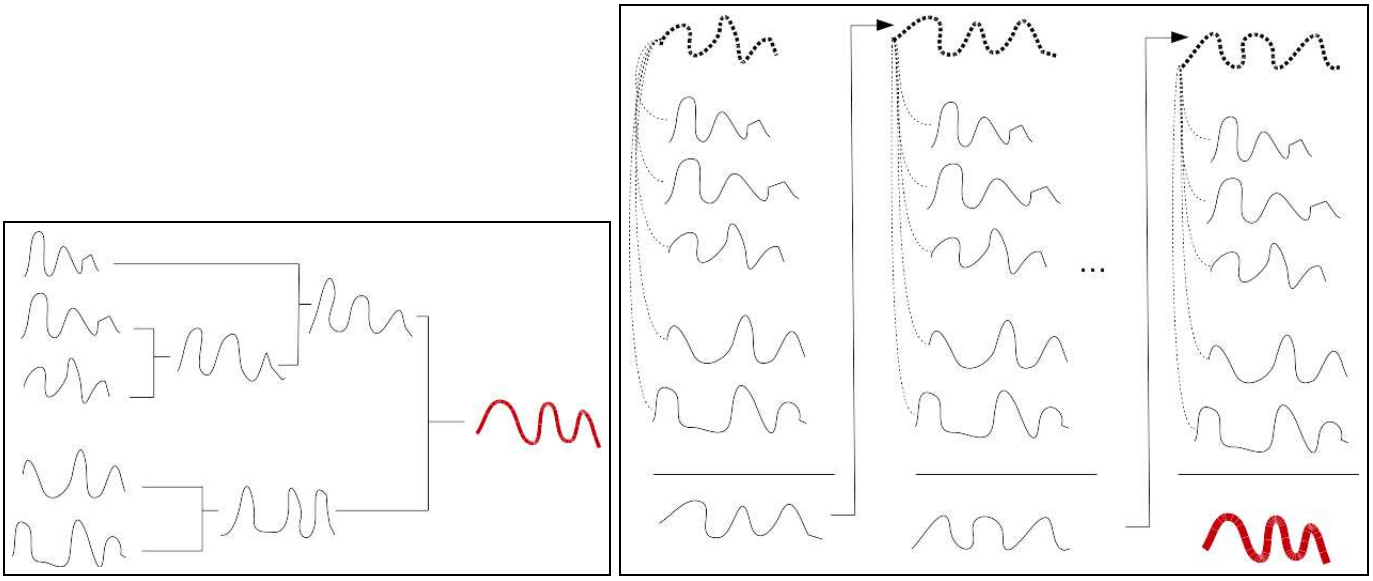
A single alignment path is required for calculating the time elastic centroid of a pair of time series (Def. 2.3). However multiple path alignments need to be considered for evaluating the centroid of a larger set of time series. Multiple alignments have been widely studied in bioinformatics [6] and it has been shown that the computational complexity of computing the optimal alignment of a set of sequences under the sum of all pairs (SP) score scheme is a NP-complete problem [7] [8] with a time and space complexity of $O(L^k)$ where k is the number of sequences in the set and L is the length of the sequences for the search of dynamic programming optimal solution [9]. This last complexity result applies for the search of the time elastic centroid of a set of k time series with respect to the DTW measure. Hence, the search for an optimal solution becomes rapidly intractable as k increases and sub-optimal heuristic solutions have been subsequently proposed, each of them mostly falling into one of the following three categories.

2.2.1 Progressive heuristics

Progressive heuristics estimate the time elastic centroid of a set of k time series by combining pairwise centroids (Def. 2.3). This kind of approach construct 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 completed, the root is associated to the estimate of the data set centroid. The proposed strategies differ on the way the tree is constructed. One popular approach consists in providing a random order for the leaves, and then constructing the binary tree up to the root from this ordering [10]. Another approach consists in constructing a dendrogram (a hierachical ascendent clustering) from the data set and then, according to this dendrogram, to calculate pairwise centroids starting with the closest pairs of time series and progressively aggregating the farthest ones [11] as depicted in the left drawing of Fig. 1. Note that these heuristics are fully based on the calculation of pairwise centroid, and thus they do not explicitly require the evaluation of a DTW centroid for more than two time series. Their complexities are linear with the number of time series in the data set.

2.2.2 Iterative heuristics

Iterative heuristics are based on an iterated three steps process. The first step consists, given a temporary centroid candidate, in calculating the inertia, i.e. the sum



(a) Progressive agglomeration (b) Iterative agglomeraation with refinement

Fig. 1. Progressive hierarchical with similar first agglomeration (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

of the DTW distances between the temporary centroid and each time series in the data set. The second step evaluates for each time series $u_j(i)$ in the data set ($j \in \{1 \cdots n\}$) its pairwise best alignment with the temporary centroid. A new time series $\tilde{u}_j(i)$ is thus constructed that contains all the samples of time series $u_j(i)$, but with time stretching or compressing accordingly to the best alignment path. The third step consists in producing the new temporary centroid candidate $c(i)$ from the set $\{\tilde{u}_j(i)\}$ by averaging (in the Euclidean centroid sens), successively all the samples at timestamps i of the $\tilde{u}_j(i)$ time series. Basically, $c(i) = \sum_{j=1..n_i} \tilde{u}_j(i).\mathbb{1}(i,j) / \sum_{j=1..n_i} \mathbb{1}(i,j))$, where $\mathbb{1}(i,j)$ is an indicator function that equals 1 if time series $\tilde{u}_j$ is defined for the timestamps i, 0 otherwise.

The new centroid candidate replace then the previous one and this process is iterated until the inertia is not decreased anymore or the maximum number of iterations is reached. Generally, the first temporary centroid candidate is set to the DTW medoid of the considered data set. This process is illustrated in the right drawing of Fig. 1. The three steps of the heuristics have been first proposed in [12]. The iterative aspect of the heuristics has been initially introduced by [13] and refined by [14]. Note that this kind of heuristics, contrarily to the progressive one, needs to evaluate, at each iteration, all the alignments with the current centroid candidate. The iterative heuristics complexity is higher than the progressive heuristics, the extra computing cost being linear with the number of iterations. More sophisticated approaches have been proposed to escape some local minima. In [15] a genetic algorithm, managing a population of centroid candidates, has been evaluated to improve with some success the straightforward iterative heuristics.

2.2.3 *Optimization approaches*

Given the set of all time series $\mathbb{S}$ and $S = \{u_j\}_{j=1\cdots n} \subseteq \mathbb{S}$ a subset of n time series, optimization approaches try to estimate the centroid of S from the definition of an optimization problem, that is generally expressed by Eq. 2 given below

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

To our knowledge, the first attempt to solve this kind of direct approach to time elastic centroid estimation has been recently described in [16].

The authors have derived a solution of their original non convex constrained optimization problem, that integrates a temporal weightings of local sample alignments to highlight temporal region of interest in the time series data set, and penalized the other temporal regions. Two time elastic measures have been specifically addressed: i) a dynamic time warping measure between a time series and a weighted time series (representing the centroid estimate) and ii) a (non definite) kernel DTW called DTAK [17]. Their results are quite promising. However, some questions remain open regarding the applicability and scalability of the method for long or multivariate time series, for which the number of parameters to optimize is becoming quite large, and over-fitting becoming potentially an issue. As numerous local *optima* may exist in practice, the method is not guaranteed to converge toward the best possible centroid, which is the case anyway for all other known approaches.

2.3 Discussion and motivation

The state of the art in time elastic centroid calculation tells us that an exact centroid, if it exists, is a solution of a NP-complete problem whose complexity is exponential with the number of time series to average. Heuristics with increasing time complexity have been proposed since the early 2000. The simple pairwise progressive aggregation technique is the less complex approach but is suffers from a dependence to initial conditions. The iterative aggregation is reputed more efficient, but with a higher computing cost. It could be combined with ensemble methods or soft optimization such as genetic algorithms. The non convex optimization approach has the merit to tackle directly the mathematical formulation of the centroid problem in a time elastic distance context. It nevertheless involves a higher complexity and must deal with a quite large set of parameters to optimize (the weights and the sample of the centroid). Its scalability could be questioned, specifically for high dimensional multivariate time series.

It should be also mentioned that some criticisms have been made about some of these heuristics in [18]. Among other, the fact that DTW is not a metric (the triangle inequality is not satisfied) is an issue that could explain undesired behavior such as the drift of the centroid out-side the cluster of time series it is supposed to average. One can also mentioned that keeping a single best alignment (although several may exists, without mentioning the *good* ones) can increase the dependance of the solution to the initial conditions or to the aggregating order of the time series that is proposed by the chosen heuristics, or potentially the convergence rates of involved heuristics.

In this paper we address the time elastic centroid estimation question not directly from the DTW perspective, but from its regularized positive definite kernel, KDTW point of view. This perspective allows for considering the centroid estimation as a preimage problem, which is by itself another optimization perspective. But more importantly, the KDTW alignment matrices can be used to derive a probabilistic interpretation of the pairwise alignment of time series. This leads to propose a solid interpolation scheme jointly along the time axis and within the sample space. We do not pretend that using KDTW and its probabilistic interpretation will solve all or even any of the fundamental question raised earlier: the problem we tackle is NP-complete, exact solution requires exponentially complex computations and any heuristics faces numerous local minima. Our aim is to bring some new light as well as new quantitative results, showing that, in this hard context, the alternative we propose are worth considering.

2.4 Time elastic kernels and their regularization

Dynamic Time Warping (DTW), [2], [3] as defined in Eq.1 can be recursively evaluated as

$$\begin{aligned} d_{dtw}(X_p, Y_q) &= d_E^2(x(p), y(q)) \\ &+ \text{Min} \begin{cases} d_{dtw}(X_{p-1}, Y_q) & sup \\ d_{dtw}(X_{p-1}, Y_{q-1}) & sub \\ d_{dtw}(X_p, Y_{q-1}) & ins \end{cases} \end{aligned} \tag{3}$$

where $d_E(x(p), y(q)$ is the Euclidean distance (possibly the square of the Euclidean distance) defined on $\mathbb{R}^k$ between the two postures in sequences X and Y taken at times p and q respectively.

Besides the fact that the DTW measure does not satisfy the triangle inequality, it is furthermore not possible to directly define a positive definite kernel from it. Hence, the optimization problem, inherent to the learning of a kernel machine, is no longer quadratic which could be, at least on some tasks, a source of limitation.

Regularized DTW: recent works [19], [20] allowed to propose new guidelines to ensure that kernels constructed from elastic measures such as DTW are positive definite. A simple instance of such regularized kernel, derived from [20] takes the following form, which relies on two recursive terms :

$$K_{rdtw}(X_p, Y_q) = K^{xy}_{rdtw}(X_p, Y_q) + K^{xx}_{rdtw}(X_p, Y_q)$$

$$K^{xy}_{dtw}(X_p, Y_q) = \tfrac{1}{3} e^{-\nu d_E^2(x(p), y(q))} \sum \begin{cases} h(p-1, q) K^{xy}_{rdtw}(X_{p-1}, Y_q) \\ h(p-1, q-1) K^{xy}_{rdtw}(X_{p-1}, Y_{q-1}) \\ h(p, q-1) K^{xy}_{rdtw}(X_p, Y_{q-1}) \end{cases}$$

$$K^{xx}_{rdtw}(X_p, Y_q) = \tfrac{1}{3} \sum \begin{cases} h(p-1, q) K^{xx}_{rdtw}(X_{p-1}, Y_q) e^{-\nu d_E^2(x(p), y(p))} \\ \Delta_{p,q} h(p, q) K^{xx}_{rdtw}(X_{p-1}, Y_{q-1}) e^{-\nu d_E^2(x(p), y(q))} \\ h(p, q-1) K^{xx}_{rdtw}(X_p, Y_{q-1}) e^{-\nu d_E^2(x(q), y(q))} \end{cases} \tag{4}$$

where $\Delta_{p,q}$ is the Kronecker's symbol, $\nu \in \mathbb{R}^+$ is a *stiffness* parameter which weights the local contributions, i.e. the distances between locally aligned positions, and $d_E(.,.)$ is a distance defined on $\mathbb{R}^k$.

The initialization is simply $K^{xy}_{rdtw}(X_0, Y_0) = K^{xx}_{rdtw}(X_0, Y_0) = 1$.

The main idea behind this line of regularization is to replace the operators $\min$ and $\max$ (which prevent the symmetrization of the kernel) by a summation operator ($\sum$). This leads to consider, not only the best possible alignment, but also all the best (or nearly the best) paths by summing up their overall cost. The parameter ν is used to control what we call nearly-the-best alignment, thus penalizing more or less alignments too far from the optimal ones. This parameter can be easily optimized through a cross-validation.

3 KDTW CENTROID AS A PREIMAGE PROBLEM

In this section, we tackle the centroid estimation question from a *kernelized centroid* point of view, the kernel of interest being KDTW.

The Moore-Aronszajn theorem [21] establishes that a reproducing kernel Hilbert space (RKHS) uniquely exists for every positive definite kernel and *vice-versa*. Let $\mathcal{H}$ be the RKHS associated to kernel κ defined on a set $\mathcal{X}$, and let $\langle .,. \rangle_{\mathcal{H}}$ be the inner product defined on $\mathcal{H}$. In addition, the representer of functional evaluation property in $\mathcal{H}$ expresses as: for any $\psi \in \mathcal{H}$ and any $x_j \in \mathcal{X}$, $\psi(x_j) = \langle \psi(.), \kappa(., x_j) \rangle_{\mathcal{H}}$.

Denoting $\phi(.)$ the map that assigns to each input $x \in \mathcal{X}$ the kernel function $\kappa(., x)$, the reproducing property of the kernel implies that for any $(x_i, x_j) \in \mathcal{X}^2$, $\kappa(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$.

Furthermore, $D_{\mathcal{H}}(x_i, x_j)^2 = ||\phi(x_i) - \phi(x_j)||^2_{\mathcal{H}} = \langle \phi(x_i), \phi(x_i) \rangle_{\mathcal{H}} + \langle \phi(x_j), \phi(x_j) \rangle_{\mathcal{H}} - 2.\langle \phi(x_i), \phi(x_) \rangle_{\mathcal{H}}$ is the generalization of the squared Euclidean distance defined in the feature space $\mathcal{H}$: it expresses in kernel terms as $D_{\mathcal{H}}(x_i, x_j)^2 = \kappa(x_i, x_i)^2 + \kappa(x_j, x_j)^2 - 2.\kappa(x_i, x_j)$ (the so-called kernel trick).

Finally, the representer theorem [22] states that any function $\varphi(.)^*$ of a RKHS $\mathcal{H}$ minimizing a regularized cost functional of the form

$$\sum_{i=1}^{n} \mathbf{J}(\varphi(x_i), y_i) + g(||\varphi||^2_{\mathcal{H}})$$

-with predicted output $\varphi(x_i)$ for input x_i and desired output y_j, and $g(.)$ a strictly monotonically increasing function on $\mathbb{R}^+$- is equal to a kernel expansion that is expressed in term of available data ($\{(x_i, y_i)\}$) as

$$\varphi^*(.) = \sum_{i=1}^{n} \gamma_i \kappa(x_i, .), \text{ where } \forall i, \gamma_i \in \mathbb{R}. \tag{5}$$

Hence, a direct definition of the kernelized centroid of the set $\{x_i, i = 1..n\}$ expressed in the RKHS $\mathcal{H}$ feature space associated to kernel κ can be written as

$$\begin{aligned} \varphi^*(.) &= \arg\min_{\varphi(.) \in \mathcal{H}} \sum_{i=1}^{n} ||\varphi(.) - \kappa(., x_i)||^2_{\mathcal{H}} \\ &= \arg\min_{\varphi(.) \in \mathcal{H}} n \cdot ||\varphi(.)||^2_{\mathcal{H}} - 2 \cdot \sum_{j=1}^{n} \langle \varphi(.), \kappa(., x_j) \rangle_{\mathcal{H}} \end{aligned} \tag{6}$$

The representer theorem applies and thus $\varphi^*(.)$ takes the form given in Eq. 5 which allows to rewrite Eq. 6 as

$$\begin{aligned} \varphi^*(.) = \arg \min_{\{\lambda_i\}_{i=1 \cdots n}} & \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_i \gamma_j \kappa(x_i, x_j) \\ & - 2 \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_j \kappa(x_i, x_j) \end{aligned} \tag{7}$$

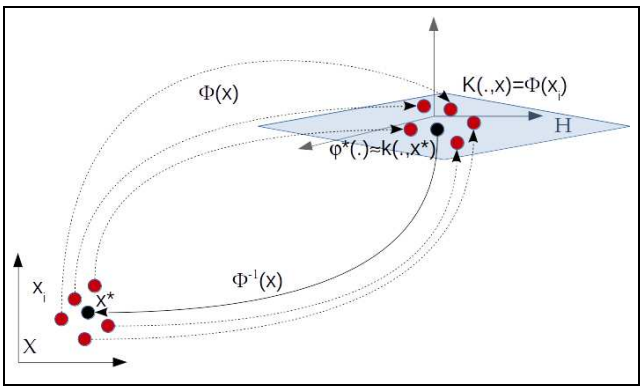


Fig. 2. Centroid estimation viewed as a preimage problem.

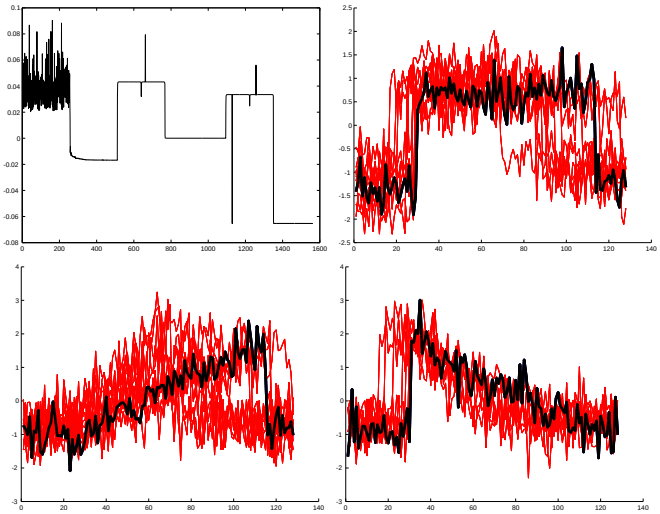


Fig. 3. Centroid estimation for the 3 categories contained in the CBF dataset as a solution of the preimage problem. In bold black, the centroid time series, in light red the time series of the dataset that are averaged. In the top left subfigure, the value of the functional that is minimized in a log-scale, as a function of the iteration index.

Unfortunately, if the kernelized centroid is related to a well defined quadratic optimization problem in the RKHS space $\mathcal{H}$ (Eq. 7), it is an ill-posed problem in set $\mathcal{X}$, known as the preimage problem, since the existence of the pre-image of $\phi(.)^*$ might not exist. Instead, we are seeking the best approximation, namely $x^* \in \mathcal{X}$ whose map $\phi(x^*) = \kappa(., x^*)$ is as close as possible to $\varphi(.)^*$, as illustrated in Fig.2.

Hence, if we drop the term that does depend upon x, the optimization problem becomes

$$\begin{aligned} x^* &= \operatorname*{argmin}_{x \in \mathcal{X}} n \cdot ||\kappa(., x)||^2_{\mathcal{H}} - 2 \cdot \sum_{j=1}^{n} \langle \kappa(., x), \kappa(., x_j) \rangle_{\mathcal{H}} \\ &= \operatorname*{argmin}_{x \in \mathcal{X}} n \cdot \kappa(x, x) - 2 \cdot \sum_{j=1}^{n} \kappa(x, x_j) \end{aligned} \tag{8}$$

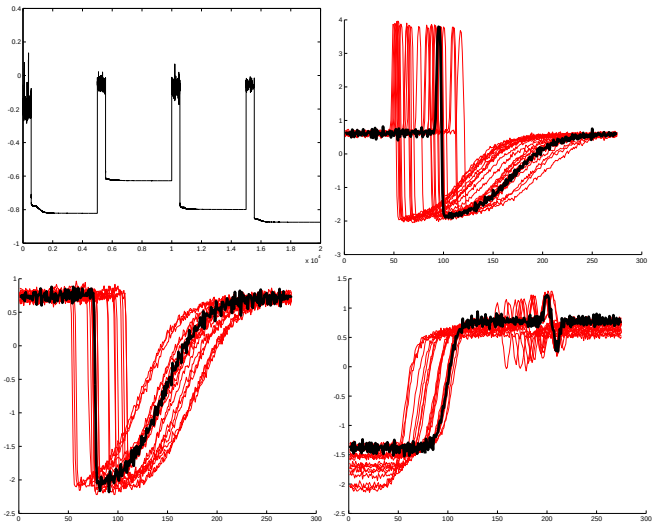


Fig. 4. Centroid estimations of the first 3 categories (among 4) contained in the Trace dataset as a solution of the preimage problem. In bold blue, the centroid time series, in light red the time series of dataset that are averaged. In the top left sub-figure, the value of the functional that is minimized in a log-scale, as a function of the iteration index.

For KDTW, such optimization problem cannot be straightforwardly addressed using gradient based approaches mainly because the derivative cannot be derived analytically but also because the number of variables (linear with the length of the time series and with the dimensionality of each sample) is generally high so that this approach easily meets over-fitting problems. Derivative free method, that tries to locally model the functional to optimize could be nevertheless attempted. To experiment such preimage formulation for the time elastic centroid estimation of a set of time series, we have used the state of the art BOBYQA algorithm developed for bound constrained optimization without derivatives [23]. Fig.3 and Fig.4 give the centroid estimations for each categories of the CBF and Trace datasets respectively [24]. The values of the functional to minimize express as the number of iteration increase are given in the top left sub-figures. The optimization process has been initialized using the medoid for each category. We show that the required number of iterations is quite high and depends upon the number of variables. For the CBF dataset, the time series are 128 samples long while for the Trace dataset they are 275 samples long. The convergence rate is roughly ten time slower for the Trace data set comparatively to the CBF dataset, mainly because KDTW complexity is quadratic with the length of the time series. The iteration cost becomes somehow prohibitive for long time series or large time series dataset. Although this approach could be possibly optimized, parameters need to be setup carefully (basically the trust region definition) and, in any case, as stated earlier, the optimum that is provided remains an estimation for the seek centroid. Finally, notice that the functional starts to be lowered after a number of iterations (here twice the length of the time series) that are used to initially estimate locally the functional.

4 Probabilistic interpretation of time elastic kernel alignment matrices

We consider in this section the recursive term $K_{dtw}^{xy}(.,.)$ that is exploited in Eq. 4. When evaluating the similarity between two time series X_p and Y_q of respective length p and q, this recursion allows for the construction of an alignment matrix $AM(i,j)$ with $i \in \{1 \cdots p\}$ and $i \in \{1 \cdots q\}$. The cell at location (i,j) contains the summation of the global costs of all alignment paths, as defined in *definition* 2.1, that connect cell $(1,1)$ with cell (i,j). For any alignment path π, the global cost expresses as

$$cost(\pi) = \prod_{k=1}^{|\pi|} e^{-\nu d_E^2(X(\pi_k(1)),Y(\pi_k(2)))} \quad (9)$$

i.e. as the product along the path of the local alignment costs. We can give a probabilistic interpretation of these local costs $exp(-\nu d_E^2(X(\pi_k(1)),Y(\pi_k(2))))$: basically we can consider that these local costs correspond (to within a multiplicative scalar constant) to the local *a priori* probability to align sample $X(\pi_k(1))$ with sample $Y(\pi_k(2))$. By doing so, we end up attaching a probability distribution to the set of all alignment paths, $cost(\pi)$ being (to within a multiplicative scalar constant) the probability attached to alignment path π.

Hence, the cell (i,j) of matrix AM, contains the sum of the probabilities (to within a multiplicative scalar constant) of the paths that connect cell $(1,1)$ to cell (i,j).

Similarly, if, instead of X and Y we evaluate the similarity between X_r and Y_r, obtained from X and Y by reversing the temporal index, we get an alignment matrix AM_r whose cell (i,j) contains the sum of the probabilities (to within a multiplicative scalar constant) of the paths that connect cell (p,q) tp cell (i,j).

Finally, multiplying properly cells of AM with cells of AM_r gives the Alignment Matrix Average (AMA) defined as

$$AMA(i,j) = AM(i,j) \cdot AM_r(p-i+1, q-j+1) \quad (10)$$

and whose cell (i,j) contains the sum of the probabilities (upto the normalization constant) of the paths that connect cell $(1,1)$ to cell (p,q) while going through the cell (i,j).

From this path probability distribution we can now derive a alignment probability distribution between the samples of X and the samples of Y as follows

- for all i, the probability to align sample $X(i)$ is $P(i) = 1$; all samples need to be aligned.
- Similarly, for all j, the probability to align sample $Y(j)$ is $P(j) = 1$.
- The probability to align sample $X(i)$ with sample $Y(j)$ is $P(i,j) = P(i|j).P(j) = P(i|j)$. $P(i|j)$ is the

probability that sample $X(i)$ is aligned with sample $Y(j)$ given that the alignment process is in state j. The estimation of $P(i|j)$ is obtained thanks to matrix AMA as

$$P(i|j) = \frac{AMA(i,j)}{\sum_{i=1}^{p} AMA(i,j)}$$

- Furthermore, the probability to align sample $X(i)$ with sample $Y(j)$ is also $P(i,j) = P(j|i).P(i) = P(j|i)$. Similarly, the estimation of $P(j|i)$ is obtained thanks to matrix AMA as

$$P(j|i) = \frac{AMA(i,j)}{\sum_{j=1}^{q} AMA(i,j)} \quad (11)$$

Notice that the normalization constant previously mentioned earlier is eliminated.

Since $P(i,j) = P(i|j) = P(j|i)$, we can finally estimate the probability to align sample $X(i)$ with sample $Y(j)$ as

$$P(i,j) = \frac{1}{2} \cdot \left(\frac{AMA(i,j)}{\sum_{i=1}^{p} AMA(i,j)} + \frac{AMA(i,j)}{\sum_{j=1}^{q} AMA(i,j)} \right) \quad (12)$$

Eq. 12 is at the basis of our pairwise time elastic time series averaging algorithm given below.

5 Time elastic centroid based on the AMA alignment matrix

From the KDTW kernel structure, the AMA matrix structure and following the so-called DtwBarycenter Averaging (DBA) method developed by [12], [14], [13] we develop first the KernelDtwNarycenter Averaging (KDBA) algorithm for estimating a time elastic centroid for a set of time series according to an iterative agglomerative heuristic as depicted in Fig. 1b. Secondly, we detail the concept of a time elastic average for a pair of time series (KDTW-PWA), and then propose a progressive heuristic as depicted in Fig. 1a that exploits KDTW-PWA to estimate another kind of time elastic centroid (KDTW-C1) for a set of time series of any cardinal.

5.1 KDTW-Centroid of a set of time series based on KDBA algorithm

Following the DBA algorithmic philosophy [12], [13], we address here the elaboration of our kernelized version called KDBA. KDBA directly exploits the definition of the alignment matrix average (AMA) as defined in Eq.10 and its probabilistic interpretation Eq.12.

Let consider a set S of N time series, $S = \{S_1, S_2, \cdots, S_N\}$, and R a reference time series. Let $|R|$ and $|S_n|$ be the lengths of R and S_n respectively. $P_n(i,j)$, with $i = 1\{1, ...|S_n|\}$ and $j = 1\{1, ...|R|\}$, is obtained from the AMA matrix resulting from the alignment of S_n with R, according to Eq.12. Algorithm 1 compute an average time series A according to the following equation

$$\forall i \in \{1, \cdots, |r|\},\ \mathtt{A}(i) = \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{|S_n|} P_n(i,j) S_n(j) \quad (13)$$

Algorithm 1 KDBA

```
1:  procedure KDBA(R,S, ν)
2:      // R: a reference time series
3:      // S: a set of time series {S_1, ··· , S_N}
4:      // ν: the stiffness parameter of KDTW kernel
5:      Double AMA(.,.);
6:      Vector-Of-SetOfSamples SampleAssociations(L);
7:      Ts A(|R|); //Create a D dimensional
8:              //time series of length L;
9:      for Int i = 1 to |R| do SampleAssociations(i)={};
10:     for Int n = 1 to |S| do
11:         Evaluate AMA matrix for R, S_n with ν;
12:         Ts ts//containing L "zeroed" samples;
13:         Double normFactor(|R|);
14:         for Int i = 1 to |R| do
15:             normFactor(i)=0;
16:             for Int j = 1 to |S_n| do
17:                 ts(i) = ts(i) + S_n(j) * AMA(i,j);
18:                 normFactor(i) = normFactor(i)+
19:                         AMA(i,j);
20:             ts(i) = ts_1(i)/normFactor(i);
21:             SampleAssociations(i)=(ts(i));
22:     for Int i = 1 to |R| do
23:         A(i)=barycenter(SampleAssociations(i));
24:     return A
```

Algorithm 2 iKDBA

```
1:  procedure IKDBA(C,S, ν)
2:      //C: a reference time series
3:      //S: a set of time series
4:      //maxIter: maximum number of iterations
5:      //ν: the stiffness parameter of KDTW kernel
6:      Ts A; //a D dimensional Timeseries
7:      Double inertia = computeInertia(C, S);
8:      Boolean Continue=True;
9:      Int i = 0;
10:     while Continue do
11:         A=C;
12:         C=KDBA(C,S, ν);
13:         Double new_inertia = computeInertia(C, S);
14:         if new_inertia > inertia OR i > maxIter then
15:             Continue = False;
16:         i=i+1;
17:     return A
```

Note that the iterative average produced by algorithm 1 as the same size of the reference time series R.

The algorithm 1 can be refined by iterating until no improvement can be obtained [14]. An improvement is observed when the sum of the distances (resp. similarities) between the current average R and the new one provided by KDBA, A, is lowered (resp. increased). Algorithm 2 implements this iterative strategy, that will find necessarily a local minimum or will stop when a maximum number of iterations has been reached.

5.2 KDTW average of a pair of time series (KDTW-PWA)

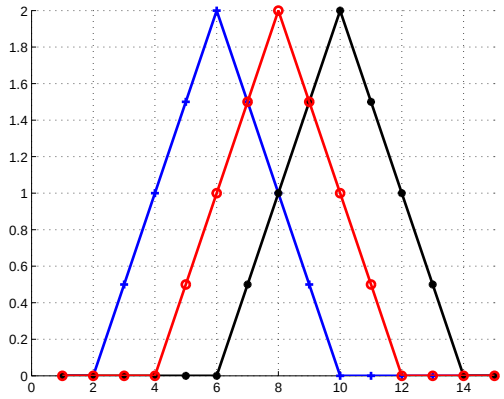


Fig. 5. Expected time location for the centroid (in red circles) of two triangular shaped time series shifted in time (in blue '+' and black '*')

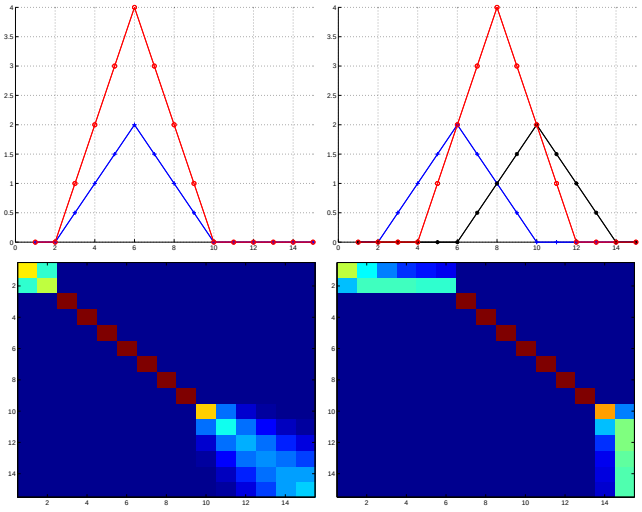


Fig. 6. Averaging triangular shaped time series. On the left, the two time series (in blue) are identical (superimposed) and the centroid (red) has been amplified by a two factor. On the right the two time series (in blue) have the same shape but have been shifted temporally. The KDTW-PWA is given in red, still amplified by a two factor. The corresponding (normalized) AMA alignment matrices are given at the bottom.

KDBA, similarly to DBA averages a set of time series in the sample space but not along the time axis. Basically, suppose we consider averaging two triangular shaped time series such as the blue '+' curve and the black dot curve presented in Fig.5.2. When using DBA or KDBA algorithms with one of the two curves playing the role of the reference time series, then the calculated average would be the reference curve itself. However, one would expect to average also the time shift between the two curves, that is to get the red round circle curves presented in Fig.fig:time-shift. This is precisely our main motivation behind the derivation of the following Pair Wise Averaging (KDTW-PWA) algorithm dedicated to average a pair of time series in the sample space but also along the time axis.

Algorithm 3 provides the KDTW-PWA average (A) of the two time series X and Y according to Eq.14.

$$\begin{aligned} \forall k = 1 \cdots L, \quad A(k) &= \sum_{i,j|\frac{i+j}{2}=k} \left(P(i,j) \cdot \frac{X(i)+Y(j)}{2} \right) \\ &= \sum_{i,j|\frac{i+j}{2}=k} \left(\frac{P(i|j)+P(j|i)}{2} \cdot \frac{X(i)+Y(j)}{2} \right) \end{aligned} \tag{14}$$

Algorithm 3 KDTW-PWA

procedure KDTW-PWA(X,Y, AMA)
  //X,Y: two time series of D dimensional samples
  //AMA: the average alignment matrix for X,Y
  Int $p = |X|$, $q = |Y|$, $L = \max\{p, q\}$;
  Ts $A(L)$, $B(L)$; //Create 2 D dimensional
    //time series of length L;
  Double α;
  Double $N_A(L), N_B(L)$; //two double arrays
  for Int $i = 1$ to L **do**
    for d=1 to D **do**
      $A(i,d) = 0$, $B(i,d) = 0$;
    $N_A(i) = 0$, $N_B(i) = 0$;
  for Int $i = 1$ to L **do**
    if $i < p$ **then**
      for Int $j = 1$ to q **do**
        $\alpha = (i+j)/2 - \lfloor (i+j)/2 \rfloor$;
        for d=1 to D **do**
          $A(\lfloor (i+j)/2 \rfloor, d) += \alpha \cdot (X(i,d) + Y(j,d)) \cdot AMA(i,j)$;
          $A(\lceil (i+j)/2 \rceil, d) += (1-\alpha) \cdot (X(i,d) + Y(j,d)) \cdot AMA(i,j)$;
        $N_A(\lfloor (i+j)/2 \rfloor) += \alpha * AMA(i,j)$;
        $N_A(\lceil (i+j)/2 \rceil) += (1-\alpha) * AMA(i,j)$;
    if $i < q$ **then**
      for Int $j = 1$ to p **do**
        $\alpha = (i+j)/2 - \lfloor (i+j)/2 \rfloor$;
        for d=1 to D **do**
          $B(\lfloor (i+j)/2 \rfloor, d) += \alpha \cdot (X(j,d) + Y(i,d)) \cdot AMA(j,i)$;
          $B(\lceil (i+j)/2 \rceil, d) += (1-\alpha) \cdot (X(j,d) + Y(i,d)) \cdot AMA(j,i)$;
        $N_B(\lfloor (i+j)/2 \rfloor) += \alpha * AMA(j,i)$;
        $N_B(\lceil (i+j)/2 \rceil) += (1-\alpha) * AMA(j,i)$;
  for Int $i = 1$ to L **do**
    for d=1 to D **do**
      $A(i,d) = (A(i,d)/N_A(i) + B(i,d)/N_B(i))/4$;
  return A

As the time index are considered discrete (integer

value), the time averaging $(i+j)/2$ is smoothed between the floor and cell integer values, using the smoothing coefficient α (line 17 of the algorithm).

Thus, the KDTW-PWA averages jointly the sample values of the two time series, but also their time locations. Eq. 14 allows for interpreting the centroid of a pair of time series as the mathematical expectation of aligning the two sequences of samples.

In Fig 6 we present a very simple experiment consisting in averaging two identical time series having triangular shapes (left of the figure) and two time series having identical triangular shapes but shifted in time. At the bottom of the figure, the corresponding AMA matrices are presented. The KDTW-PWA curves, presented in red, have been multiplied by a two factor to ease the reading of the figure. We can verify that, for the two situations, the centroid is precisely located at the correct averaged time of occurrence of the two time series, wether they are shifted in time or not. The AMA matrices show in red color the most likely alignment areas and in blue color the less likely alignment areas. The time shift is clearly visible on the right figure.

5.3 KDTW-Centroid of a set of time series based on KDTW-PWA

Algorithm 4 pKDTW-PWA

1: **procedure** pKDTW-PWA(S, ν)
2: //S: a set of time series of D dimensional samples
3: //ν: the stiffness parameter of KDTW kernel
4: Ts A; //a D dimensional time series
5: SetOfTimeSeries S_0;
6: **while** $|S| > 1$ **do**
7: $S_0 = \emptyset$
8: **while** $|S| > 1$ **do**
9: Let ts_1, ts_2 the first two time series in S;
10: Evaluate the AMA matrix for ts_1 and ts_2
11: with ν as the stiffness parameter
12: A = KDTW-PWA(ts_1, ts_2, AMA);
13: $S_0 = S_0 \cup \{A\}$;
14: $S = S \setminus \{ts_1, ts_2\}$;
15: $S = S_0 \cup S$;
16: Let A be the single element of S;
17: **return** A

To average a larger set of time series using the pairwise average KDTW-PWA, we simply adopt the progressive agglomerative approach presented in Fig.1a. This heuristic, detailed in Algorithm 4 is in $O(n)$ complexity, n being the size of the considered set of time series.

Figures presented in Table 1 compare the centroid estimates provided by the iterated DBA, iKDBA and pKDTW-PWA algorithms. For the experiment, The DBA and iKDBA have been iterated at most 20 times. If DBA and iKDBA estimates look quite similar, the centroid estimates provided by the pKDTW-PWA algorithm is much smooth. This is a general property of this algorithm that implements a time averaging principle, based on the time expectation of sample occurrences, that manages, somehow, to filter *noisy* data. Notice also that the DBA and iKDBA estimates are also close for the CBF data set to the ones provided by the preimage approach (Fig.3).

6 Experimentation

The purpose of this experiment is to evaluate the effectiveness of the proposed time elastic averaging methods against a double baseline, namely kmedoids-based approaches and the DBA algorithm. The first baseline will allow to compare centroid-based to medoid-based approaches. The second baseline will bring some highlights about the benefit one can expect from using p.d elastic kernels instead of indefinite ones such as DTW in the context of time series averaging. DBA is also considered currently as a state of the art method to average a set of sequences consistently with DTW.

To that end, we empirically evaluate the effectiveness of the methods using a first near neighbor (1-NN) classification task on a set of time series coming from quite diverse application fields. The task consists in representing each categories contained in a training data set by a its medoid or centroid estimate and then evaluating the error rate of a 1-NN classifier on an independent testing data set. Hence, the classification rule consists in affecting to the tested time series the category which corresponds to the closest (or most similar) medoid or centroid according to DTW or KDTW measures.

DBA and iKDBA iterative centroid methods are iterated at most 20 times and provides a local estimates of the centroid. pKDTW-PWA progressive agglomerative centroid method is only processed once and hence is roughly 20 times faster than iKDBA method and about 10 times faster than DBA.

A collection of 45 data sets in total has been used to assess the proposed algorithms. This collection includes synthetic and real data sets, univariate and multivariate time series data sets. These sets are distributed as follow:

- 42 of these data sets are available at the UCR repository [24]. Basically we have used all the data sets but the three *StarLightCurves*, *Non-Invasive Fetal ECG Thorax1* and *Non-Invasive Fetal ECG Thorax2* data sets. Although these data sets are still tractable, their computational cost is heavy because of their size and the length of the time series they content. All these data sets are composed with scalar time series.
- One data set, uWaveGestureLibrary_3D as been constructed from the uWaveGestureLibrary_X—Y—Z scalar data sets to compose a new set of multivariate (3D) time series.

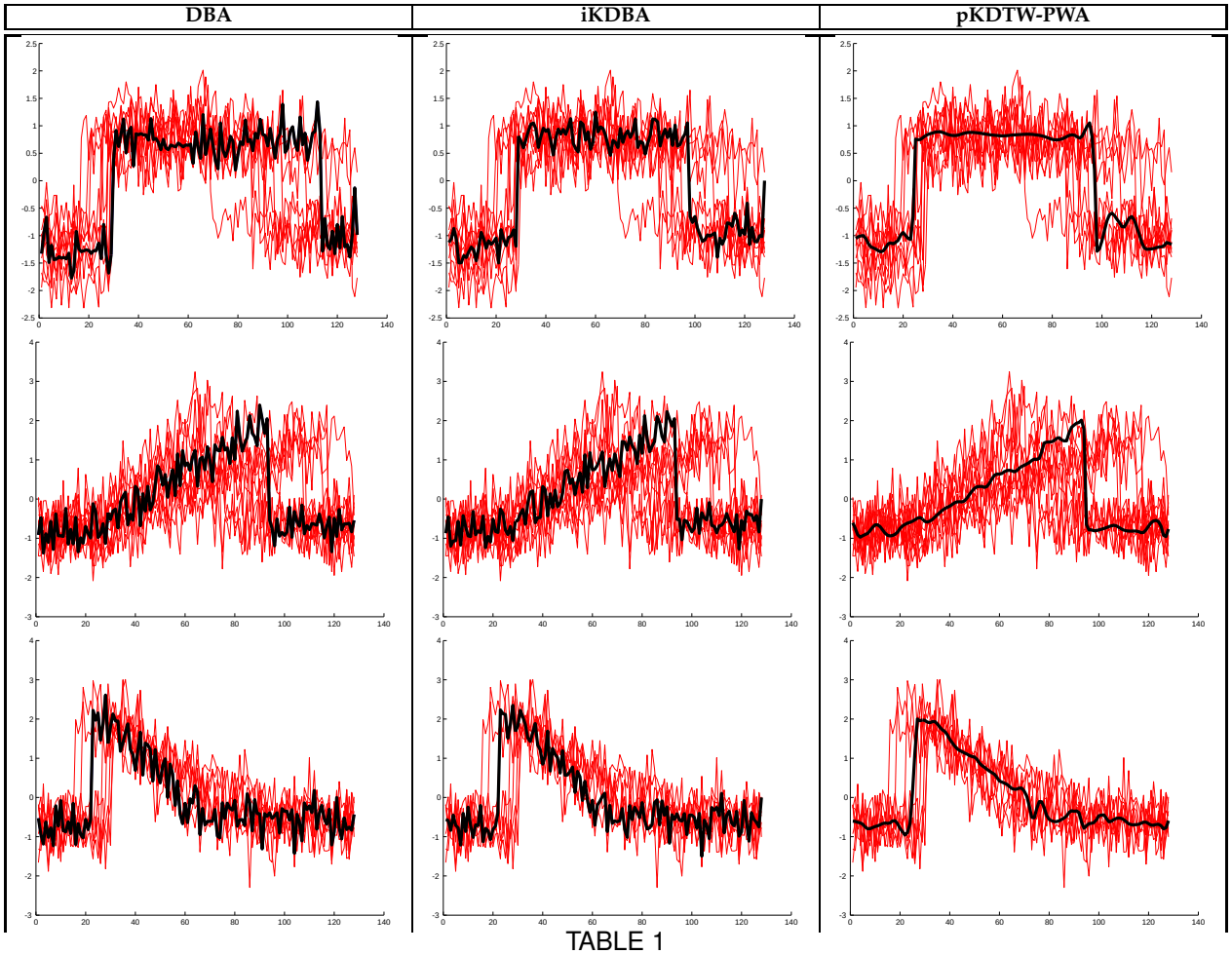


TABLE 1

Centroid estimation for the 3 categories of the CBF dataset. The centroid estimation is presented as a bold black curve above the superposition of the time series (in light red) that are averaged. The centroid estimates provided by the DBA algorithm are given in the left column, the center column shows the estimates provided by the iKDBA algorithm and the right column shows the estimates provided by the pKDTW-PWA algorithm.

- One data set, CharTrajTT, is available at the UCI Repository [25] under the name *Character Trajectories Data Set*. This data set that contains multivariate (3D) time series has been divided in two equal size TRAIN and TEST data sets for the experiment.
- The last data set *PWM2* that stands for Pulse Width Modulation [26] has been specifically defined to demonstrate a weakness in dynamic time warping (DTW) pseudo distance. This data set is composed with artificial scalar time series.

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

Furthermore, for $KDTW_{Medoid}$, $iKDBA$ and $pKDTW-PWA$ methods, 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 $\{.05, .1, .25, .5, 1, 2, 5, 10, 25, 50, 100\}$. The value that minimizes, on the TRAIN data, the LOO classification error rate is then used for providing 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 provides lower error rates comparatively to DTW_{Medoid}, so do $iKDBA$ and $pKDTW-PWA$ comparatively to $KDTW_{Medoid}$.
ii) $iKDBA$ outperforms DBA: in the same experimental conditions (at most 20 iterations), the kernalized version of the DTW measure leads to better classification accuracies. This somehow confirms previous results obtained for SVM classification [20] on such kind of datasets.
iii) $pKDTW-PWA$ outperforms $iKDBA$: this results seems to show that a joint averaging in the sample space and along the time axis improves the

classification accuracies. As $pKDTW - PWA$ provides a centroid estimation in a single agglomerative step, one can conjecture that this method converges faster toward a satisfactory centroid candidate.

The average ranking for all five tested methods that sustains our preliminary conclusion is given at the bottom of Table 2.

Following work by [27] on statistical tests available to evaluate significance of classifiers error rate differences over multiple data dets, we have conducted a Friedman's significance test, a sort of non-parametric counterpart of the well known ANOVA. This test ranks the algorithms for each data set separately, the best performing algorithm getting the rank of 1, the second best rank 2, etc.

According to this test, the null hypothesis is rejected (with a $P - value < 2.2e - 16$). Post-hoc tests can thus be carried out to compare pairwise algorithms using the Wilcoxon-Nemenyi-McDonald-Thompson test [28]. To that end, we have used the R code provided by [29] to produced the parallel coordinate plots and the boxplots presented in Fig.7 and the results reported in Table 3.

TABLE 3
Significance test: $Algorithm_1$ is considered to be significantly better than $Algorithm_2$ according to the Friedman's test if the P-value (in bold characters) associated to the pairwise test is below 0.05.

$Algorithm_1$	$Algorithm_2$	P-value
DBA	DTW_{Medoid}	**7.79e-06**
$KDTW_{Medoid}$	DTW_{Medoid}	**4.41e-03**
$iKDBA$	DTW_{Medoid}	**6.85e-12**
$pKDTW - PWA$	DTW_{Medoid}	**2.44e-15**
$KDTW_{Medoid}$	DBA	6.21e-01
$iKDBA$	DBA	1.73e-01
$pKDTW - PWA$	DBA	**9.91e-03**
$iKDBA$	$KDTW_{Medoid}$	**2.62e-03**
$pKDTW - PWA$	$KDTW_{Medoid}$	**2.91e-05**
$pKDTW - PWA$	$iKDBA$	8.36e-01

Table 3 reports the P-values for each pair of tested algorithms. This post-hoc analysis confirms partially our previous analysis of the classification results. If we consider that the null hypothesis is rejected when the P-value is below 0.05 the post-hoc analysis shows that centroid-based approaches perform significantly better than medoid-based approaches. Furthermore, $KDTW_{Medoid}$ appears to be significantly better than DTW_{Medoid}.

$pKDTW - PWA$ is furthermore evaluated significantly better than DBA but not significantly better than $iKDBA$ on this experiment. Notice also that DBA is not shown to perform significantly better than $KDTW_{Medoid}$.

This post-hoc analysis is synthesized in Fig.8 which shows, in the context of our experiment, the ranking graph for the five tested algorithms.

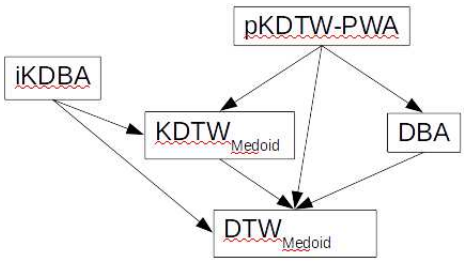


Fig. 8. Dominance graph for the five tested algorithms, according to the significance relation corresponding to Table 3 with a P-value threshold set to .05.

7 Conclusion

In this paper, we have addressed the reputedly difficult problem of averaging a set of time series in the context of a time elastic distance such as Dynamic Time Warping. The new perspective brought by the kernelization of such elastic distance firstly allows for addressing this problem as a preimage problem, which is unfortunately an ill-posed problem that could suffers, in the scope of long time series, from an excess of variables. Furthermore, this kind of preimage problem can only be resolved using gradient free optimization procedures that are computationally very expensive (since a lot of costly functional evaluation are required).

However, this kernelization perspective allows for re-interpreting pairwise kernel alignment matrices as some distribution of probabilities over alignment paths. Based on this re-interpretation, we have been able to propose two distinct algorithms, $iKDBA$ and $pKDTW - PWA$, that exploit respectively iterative and progressive agglomerative heuristics that have been developed to propose approximate solution to the multi-alignment of time series problem.

The quite extensive experiment we have carried out on synthetic or real data sets, containing mostly univariate but also some multivariate time series, shows that centroid-based methods significantly outperform medoid-based methods in the context of a first near neighbor classification task. Most strikingly the $pKDTW - PWA$ algorithm ,that integrates a joint averaging principle in the sample space and along the time axis, is significantly better than the state of the art DBA algorithm, with potentially a lower computational cost involved. Indeed, the simple one pass progressive agglomerative heuristics that is used in that algorithm can be furthermore optimized.

Acknowledgment

The authors thank the French Ministry of Research, the Brittany Region, the General Council of Morbihan

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 near neighbor classification rule for DTW_{Medoid}, DBA (centroid), $KDTW_{Medoid}$, $iKDBA$ (centroid) and $pKDTW - PWA$ (centroid). A single medoid/centroid extracted on the TRAIN data set represent each category.

DATASET	# Cat | L	DTW_{Medoid}	DBA	$KDTW_{Medoid}$	$iKDBA$	$pKDTW - PWA$
Synthetic_Control	6|60	3.00	2.00	3.33	**2.00**	5.00
Gun_Point	2|150	44.00	32.00	52.00	30.00	**26.00**
CBF	3|128	7.89	5.33	8.11	**3.79**	4.33
Face_(all)	14|131	25.21	18.05	20.53	**14.91**	17.22
OSU_Leaf	6|427	64.05	56.20	**53.31**	55.37	55.37
Swedish_Leaf	15|128	38.56	30.08	31.36	42.08	**24.00**
50Words	50|270	48.13	41.32	23.30	23.30	**19.34**
Trace	4|275	5.00	7.00	23.00	18.00	**2.00**
Two_Patterns	4| 128	1.83	1.18	1.18	**0.70**	1.68
Wafer	2|152	64.23	33.89	43.92	**31.96**	33.73
Face_(four)	4|350	12.50	13.64	17.05	**5.70**	10.23
Lightning-2	2|637	34.43	37.70	29.51	32.79	**22.95**
Lightning-7	7|319	27.40	27.40	23.29	21.92	**19.18**
ECG200	2|96	32.00	28.00	29.00	25.00	**24.00**
Adiac	37|176	57.54	52.69	**40.67**	54.73	41.43
Yoga	2|426	47.67	47.87	47.53	46.40	**42.97**
Fish	7|463	38.86	30.29	20.57	20.00	**17.14**
Beef	5|470	60.00	53.33	56.67	53.33	**50.00**
Coffee	2|286	57.14	32.14	32.14	32.14	**21.43**
OliveOil	4|570	26.67	16.67	23.33	20.00	**13.33**
CinC_ECG_torso	4|1639	74.71	53.55	66.67	54.78	**49.64**
ChlorineConcentration	3|166	65.96	68.15	65.65	68.33	**54.40**
DiatomSizeReduction	4|345	22.88	5.88	11.11	4.58	**1.96**
ECGFiveDays	2|136	47.50	30.20	**11.38**	12.66	19.16
FacesUCR	14|131	27.95	18.44	20.73	**13.37**	13.80
Haptics	5|1092	68.18	64.61	63.64	57.79	**57.47**
InlineSkate	7|1882	78.55	76.55	78.36	76.55	**75.82**
ItalyPowerDemand	2|24	31.68	20.99	5.05	**3.89**	6.51
MALLAT	8|1024	6.95	6.10	6.87	4.22	**3.58**
MedicalImages	10|99	67.76	**58.42**	68.03	63.03	61.84
MoteStrain	2|84	15.10	13.18	12.70	12.54	**10.46**
SonyAIBORobot_SurfaceII	2|65	26.34	**21.09**	25.50	22.46	26.02
SonyAIBORobot_Surface	2|70	38.10	19.47	39.77	14.31	**7.65**
Symbols	6|398	7.64	4.42	3.92	**3.72**	3.82
TwoLeadECG	2|82	24.14	**13.17**	27.04	17.38	21.60
WordsSynonyms	25|270	70.85	64.26	64.26	61.29	**59.09**
Cricket_X	12|300	67.69	**52.82**	61.79	55.90	58.46
Cricket_Y	12|300	68.97	52.82	46.92	**46.67**	52.82
Cricket_Z	12|300	73.59	**48.97**	56.67	51.03	60.26
uWaveGestureLibrary_X	8|315	38.97	33.08	34.40	**32.55**	33.31
uWaveGestureLibrary_Y	8|315	49.30	44.44	42.18	**39.87**	40.06
uWaveGestureLibrary_Z	8|315	47.40	**39.25**	41.96	39.67	40.62
uWaveGestureLibrary_3D	8|315	10.11	**6.00**	13.74	9.32	8.46
CharTrajTT_3D	20|178	6.58	5.18	4.34	11.83	**4.20**
PWM2	3|128	43.00	35.00	21.33	20.33	**15.67**
# Best Scores	-	0	7	3	12	**23**
# Uniquely Best Scores	-	0	7	3	12	**23**
Average rank	-	4,33	2,8	3,29	2.13	**1,8**

and the European Regional Development Fund that had partially fund this research. The authors also thank the promoters for the UCR and UCI data repositories for making available the time series data sets that have been 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," *International Journal of Man-Machine Studies*, vol. 2, pp. 223–234, 1970.

[3] H. Sakoe and S. Chiba, "A dynamic programming approach to

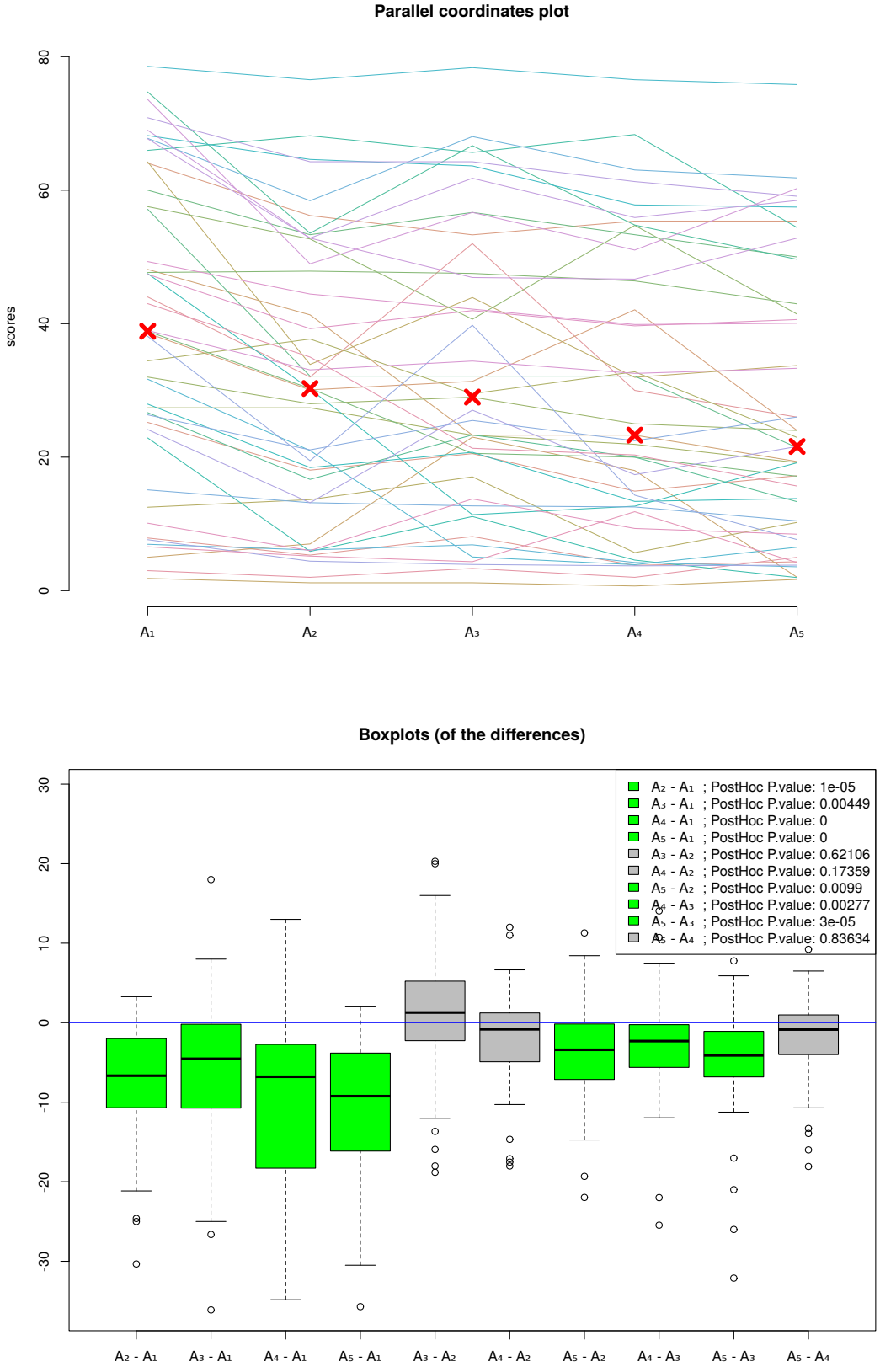


Fig. 7. *Post hoc* analysis of the Friedman's test: (A_1) DTW Medoid, (A_2) DBA, (A_3) KDTW Medoid, (A_4) iKDBA and (A_5) pKDTW-PWA.

continuous speech recognition," in *Proceedings of the 7th International Congress of Acoustic*, 1971, pp. 65–68.

[4] 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.

[5] 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.

[6] 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.

[7] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment." *Journal of Computational Biology*, vol. 1, no. 4, pp. 337–348, 1994.

[8] W. Just and W. Just, "Computational complexity of multiple sequence alignment with sp-score," *Journal of Computational Biology*, vol. 8, pp. 615–623, 1999.

[9] 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. [Online]. Available: http://dx.doi.org/10.1137/0148063

[10] L. Gupta, D. Molfese, R. Tammana, and P. Simos, "Nonlinear alignment and averaging for estimating the evoked potential," *Biomedical Engineering, IEEE Transactions on*, vol. 43, no. 4, pp. 348–356, April 1996.

[11] V. Niennattrakul and C. Ratanamahatana, "Shape averaging under time warping," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on*, vol. 02, May 2009, pp. 626–629.

[12] 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.

[13] 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.

[14] 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. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2010.09.013

[15] 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.

[16] S. Soheily-Khal, A. Douzal-Chouakria, and E. Gassier, "Time series centroid estimation under weighted and kernel dynamic time warping," *Unpublished work, under submission*, 2015.

[17] H. Shimodaira, K. I. Noma, M. Nakai, and S. Sagayama, "Dynamic Time-Alignment Kernel in Support Vector Machine," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002.

[18] 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. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72584-8_68

[19] 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.

[20] 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. [Online]. Available: http://hal.inria.fr/hal-00486916

[21] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, 1950.

[22] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*, ser. COLT '01/EuroCOLT '01. London, UK, UK: Springer-Verlag, 2001, pp. 416–426. [Online]. Available: http://dl.acm.org/citation.cfm?id=648300.755324

[23] M. J. D. Powell, "The bobyqa algorithm for bound constrained optimization without derivatives," Aug. 2009.

[24] 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/.

[25] M. Lichman, "Uci machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[26] P.-F. Marteau, "Pulse width modulation data sets," 2007. [Online]. Available: http://people.irisa.fr/Pierre-Francois.Marteau/PWM/

[27] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1248547.1248548

[28] M. Hollander and D. Wolfe, *Nonparametric Statistical Methods*, ser. Wiley Series in Probability and Statistics. Wiley, 1999. [Online]. Available: https://books.google.fr/books?id=RJAQAQAAIAAJ

[29] T. Galili, "R code for the friedman test post hoc analysis." february 2010. [Online]. Available: http://www.r-statistics.com/2010/02/post-hoc-analysis-for-friedmans-test-r-code/