On Recursive Edit Distance Kernels with Application to Time Series Classification

Pierre-François Marteau, Sylvie Gibet

On Recursive Edit Distance Kernels with Application to Time Series Classification

Pierre-François Marteau, *Member, IEEE* and Sylvie Gibet, *Member, IEEE*

Abstract—This paper proposes some extensions to the work on kernels dedicated to string or time series global alignment based on the aggregation of scores obtained by local alignments. The extensions we propose allow to construct, from classical recursive definition of time-warp distances, recursive edit distance (or time-warp) kernels that are positive definite if some sufficient conditions are satisfied. The sufficient conditions we end-up with are original and simply expressed as an inequality that need to be satisfied by the local kernel associated to the local editing operations. In practice such conditions are easily satisfied. The classification experiment we conducted on three classical time warp distances (two of which being metrics), using either first near neighbour classifier or Support Vector Machine classifier, leads to conclude that the positive definite recursive elastic kernels outperform the distance substituting kernels for the classical elastic distances we have tested.

Index Terms—Edit distance, Dynamic Time Warping, Recursive kernel, Time series classification, Support Vector Machine.

1 Introduction

Elastic similarity measures such as Dynamic Time Warping (DTW) or Edit Distances have proved to be quite efficient compared to non elastic similarity measures such as Euclidean measures or LP norms when addressing tasks that require the matching of time series data, in particular time series clustering and classification. A wide scope of applications as in physics, chemistry, finance, bio-informatics, network monitoring, etc, have demonstrated the benefits of using elastic measures. A natural follow-up to the elaboration of elastic measures is to question the existence of a Reproducing Time Warp Hilbert Spaces (RTWHS) for a given elastic measure, basically vector spaces characterized with inner products having time-warp capabilities. Unfortunately it seems that common elastic measures that are derived from DTW or more generally dynamic programming recursive algorithms are not directly induced by an inner product of any sort, even when such measures are metrics. One can conjecture that it is not possible to embed time series in an Hilbert space having a time-warp capability using these classical elastic measures, but at least one can try to propose (close) variant for which this construction is possible.

This paper aims at (re-)exploring this issue and, following earlier works ([12], [10], [33], [8], [9]) proposes Recursive Time Warp Kernels (RTWK) constructions that try to preserve the properties of elastic measures from which they are derived, while offering the possibility of embedding time series in Time Warped Hilbert Spaces. The main contributions of the paper are as follows

1) we verify the indefiniteness of the main time-warp measures used in the literature,
2) we propose some methods to construct positive definite kernels from three classical time-warp measures,
3) we experiment and compare the proposed kernels on some time series classification tasks using a large variety of time series datasets to estimate in practice the benefit we can get from such kernels.

The paper is organized as follows: the second section of the paper synthesizes the related works; the third section introduces the notation and mathematical backgrounds that are used throughout the paper; the fourth section addresses the non definiteness of classical elastic measures that prevents the direct construction of an inner product from these measures. The fifth section develops the construction of some RTWK from classical elastic measures and discusses their potential benefits. The sixth section gathers clustering and classification experimentations on a wide range of time series data and compares RTWK accuracies with classical elastic and non elastic measures. The seventh section proposes a conclusion and further research perspectives. Appendix A states the indefiniteness of classical elastic measures, and appendix B gives the proof of our main results.

2 Related works

During the last decades, the use of kernel-based methods in pattern analysis has provided numerous results and fruitful applications in various domains such as biology, statistics, networking, signal processing, etc. Some of these domains, such as bioinformatics, or more generally domains that rely on sequence or time series models, require the analysis and processing of variable length

• *P.F. Marteau and Sylvie Gibet are with the UMR IRISA, Université de Bretagne Sud, 56000 Vannes, France.*
E-mail: {Pierre-Francois.Marteau, Sylvie.Gibet}(AT)univ-ubs.fr

vectors, sequences or timestamped data. Various methods and algorithms have been developed to quantify the similarity of such objects. From the original dynamic programming [2] implementation of the symbolic edit distance [16] by Wagner and Fisher [34], the Smith and Waterman (SW) algorithm [29] has been designed to evaluate the similarity between two symbolic sequences by means of a local gap alignment. More efficient local heuristics have since been proposed to meet the massive symbolic data challenge, such as BLAST [1] or FASTA [20]. Similarly, dynamic time warping measures have been developed to evaluate similarity between numeric time series or timestamped data [32], [23], and more recently [6], [18] propose elastic metrics dedicated to such numeric data.

Our capability to construct kernels with elastic or time-warp properties from such an *elastic distance* allowing to embed time series in vector spaces (euclidean or not) has attracted attention (e.g. [10][13][11]) since significant benefits are expected from potential applications of kernel-based machine learning algorithms to variable length data, or more generally data for which some elastic matching has a meaning. Among the kernel machine algorithms applicable to discrimination or regression tasks, Support Vector Machines (SVM)[30], [4], [25] are still reported to yield state-of-the art performances although their accuracy greatly depends on the exploited kernel.

The definition of 'good' kernels from known elastic or time-warp distances applicable to data objects of variable lengths has been a major challenge since the 1990s. The notion of 'goodness' has rapidly been associated to the concept of definiteness. Basically SVM algorithms involve an optimization process whose solution is proved to be uniquely defined if and only if the kernel is positive definite: in that case the objective function to optimize is quadratic and the optimization problem convex. Nevertheless, if the definiteness of kernels is an issue, in practice, many situations exist where definite kernels are not applicable. This seems to be the case for the main elastic measures traditionally used to estimate the similarity of objects of variable lengths. A pragmatic approach consists of using indefinite kernels, although contradictory results have been reported about the impact of definiteness or indefiniteness of kernels on the empirical performances of SVMs. The sub-optimality of the non-convex optimization process is possibly one of the causes leading to these un-guaranteed performances [35], [10]. Regulation procedures have been proposed to locally approximate indefinite kernel functions by definite ones with some benefits. Among others, some approaches apply direct spectral transformations to indefinite kernels. This methods [36] consist in i) flipping the negative eigenvalues or shifting the eigenvalues using the minimal shift value required to make the spectrum of eigenvalues positive, and ii) reconstructing the kernel with the original eigenvectors in order to produce a positive semidefinite kernel. Yet, in general, 'convexification' procedures are difficult to interpret geometrically and the expected effect of the original indefinite kernel may be lost. Some theoretical highlights have been provided through approaches that consist in embedding the data into a pseudo-Euclidean (pE) space and in formulating the classification problem with an indefinite kernel, such as that of minimizing the distance between convex hulls formed from the two categories of data embedded in the pE space [11]. The geometric interpretation results in a constructive method allowing for the understanding, and in some cases the prediction of the classification behavior of an indefinite kernel SVM in the corresponding pE space.

Some other works like [28], [15] addresses the construction of elastic kernels for time series analysis through a decomposition of time series as a sum of local low degree polynomials and, using a resampling process, the piece-wise approximation of the time series are embedded into a proper so-called Reproducing Kernel Hilbert Space in which the SVM is learned.

Other approaches try to use directly the elastic distance into the kernel construction, without any approximation or resampling process. It is founded on the work of Haussler on convolution kernels [12] defined on a set of discrete structures such as strings, trees, or graphs. The iterative method that is developed is generative, as it allows for the building of complex kernels from the convolution of simple local kernels. Following the work of Haussler [12], Saigo et al [22] define, from the Smith and Waterman algorithm [29], a kernel to detect local alignment between strings by convolving simpler kernels. These authors show that the Smith and Waterman distance measure, dedicated to determining similar regions between two nucleotide or protein sequences, is not definite, but is nevertheless connected to the logarithm of a point-wise limit of a series of definite convolution kernels. Cuturi et al. [9] have adapted this approach to times series alignments covering the DTW elastic distance. In fact, these previous studies have very general implications, the first being that classical elastic measures can also be understood as the limit of a series of definite convolution kernels. In this paper we do not tackle the construction of recursive elastic kernel through the convolution paradigm developed by Haussler, but rather a direct constructing approach from the recursive definition of the elastic distances. This allows us to generalize to some extent the results presented by Cuturi et al. while covering a larger family of elastic distances.

3 Notations and mathematical backgrounds

We give in this section commonly used definitions, with few details, for metric, kernel and definiteness, sequence set, and classical elastic measures.

3.1 Kernel and definiteness

A very large literature exists on kernels, among which [3], [25] and [26] present a large synthesis of major results. We give hereinafter some basic definitions.

Definition 3.1: A kernel on a non empty set U refers to a complex (or real) valued symmetric function $\varphi(x,y) : U \times U \to \mathbb{C}$ (or $\mathbb{R}$).

Definition 3.2: Let U be a non empty set. A function $\varphi : U \times U \to \mathbb{C}$ is called a positive (resp. negative) definite kernel if and only if it is Hermitian (i.e. $\varphi(x,y) = \overline{\varphi(y,x)}$ where the *overline* stands for the conjugate number) for all x and y in U and $\sum_{i,j=1}^{n} c_i \bar{c_j} \varphi(x_i,x_j) \geq 0$ (resp. $\sum_{i,j=1}^{n} c_i \bar{c_j} \varphi(x_i,x_j) \leq 0$), for all n in $\mathbb{N}$, $(x_1,x_2,...,x_n) \in U^n$ and $(c_1,c_2,...,c_n) \in \mathbb{C}^n$.

Definition 3.3: Let U be a non empty set. A function $\varphi : U \times U \to \mathbb{C}$ is called a conditionally positive (resp. conditionally negative) definite kernel if and only if it is Hermitian (i.e. $\varphi(x,y) = \overline{\varphi(y,x)}$ for all x and y in U) and $\sum_{i,j=1}^{n} c_i \bar{c_j} \varphi(x_i,x_j) \geq 0$ (resp. $\sum_{i,j=1}^{n} c_i \bar{c_j} \varphi(x_i,x_j) \leq 0$), for all $n \geq 2$ in $\mathbb{N}$, $(x_1,x_2,...,x_n) \in U^n$ and $(c_1,c_2,...,c_n) \in \mathbb{C}^n$ with $\sum_{i=1}^{n} c_i = 0$.

In the last two above definitions, it is easy to show that it is sufficient to consider mutually different elements in U, i.e. collections of distinct elements $x_1,x_2,...,x_n$. This is what we will consider for the remaining of the paper.

Definition 3.4: A positive (resp. negative) definite kernel defined on a finite set U is also called a positive (resp. negative) semidefinite matrix. Similarly, a positive (resp. negative) conditionally definite kernel defined on a finite set is also called a positive (resp. negative) conditionally semidefinite matrix.

For convenience sake, we will use p.d. and c.p.d. for positive definite and conditionally positive definite to characterize either a kernel or a matrix having these properties.

Constructing p.d. kernels from c.p.d. kernels is quite straightforward. For instance, if $-\varphi$ is a c.p.d. kernel on a set U and $x_0 \in U$ then [3] $\psi(x,y) = \varphi(x,x_0) + \overline{\varphi(y,x_0)} - \varphi(x,y) - \varphi(x_0,x_0)$ is a p.d. kernel, so are $e^{(\psi(x,y))}$ and $e^{-\varphi(x,y)}$. The converse is also true.
Furthermore, $e^{-t\varphi(x,y)}$ is p.d. for $t > 0$ if $-\varphi$ is c.p.d. We will precisely use this last results to construct p.d. kernels from classical elastic distances.

3.2 Sequence set

Definition 3.5: Let $\mathbb{U}$ be the set of finite sequences (symbolic sequences or time series): $\mathbb{U} = \{A_1^p | p \in \mathbb{N}\}$. A_1^p is a sequence with discrete index varying between 1 and p. We note Ω the empty sequence (with null length) and by convention $A_1^0 = \Omega$ so that Ω is a member of set $\mathbb{U}$. $|A|$ denotes the length of the sequence A. Let $\mathbb{U}_p = \{A \in \mathbb{U} \mid |A| \leq p\}$ be the set of sequences whose length is shorter or equal to p.

Definition 3.6: Let A be a finite sequence. Let $A(i)$ be the i^{th} element (symbol or sample) of sequence A. We will consider that $A(i) \in S \times T$ where S embeds the multidimensional space variables (either symbolic or numeric) and $T \subset \mathbb{R}$ embeds the time stamp variable, so that we can write $A(i) = (a(i), t(i))$ where $a(i) \in S$ and $t(i) \in T$, with the condition that $t(i) > t(j)$ whenever $i > j$ (time stamps strictly increase in the sequence of samples). A_i^j with $i \leq j$ is the subsequence consisting of the i_{th} through the j_{th} element (inclusive) of A. So $A_i^j = A(i)A(i+1)...A(j)$. Λ denotes the null element. A_i^j with $i > j$ is the null time series, e.g. Ω.

3.3 General Edit/Elastic distance on a sequence set

Definition 3.7: An edit operation is a pair $(a,b) \neq (\Lambda,\Lambda)$ of sequence elements, written $a \to b$. Sequence B results from the application of the edit operation $a \to b$ into sequence A, written $A \Rightarrow B$ via $a \to b$, if $A = \sigma a \tau$ and $B = \sigma b \tau$ for some sub-sequences σ and τ. We call $a \to b$ a substitution operation if $a \neq \Lambda$ and $b \neq \Lambda$, a delete operation if $b = \Lambda$, an insert operation if $a = \Lambda$.

For any pair of sequences A_1^p, B_1^q, for which we consider the extensions A_0^p, B_0^q whose first element is the null symbol Λ, and for each elementary edit operation related to position $0 \leq i \leq p$ in sequence A and to position $0 \leq j \leq q$ in sequence B is associated a cost value $\Gamma_{A(i) \to B(j)}(A_1^p, B_1^q)$, or $\Gamma_{A(i) \to \Lambda,j}(A_1^p, B_1^q)$ or $\Gamma_{\Lambda,i \to B(j)}(A_1^p, B_1^q) \in \mathbb{R}$. To simplify this writing we will simply write $\Gamma(A(i) \to B(j))$, $\Gamma(A(i) \to \Lambda)$ or $\Gamma(\Lambda \to B(j))$ although this will not be fully appropriate in general.

Definition 3.8: A function $\delta : \mathbb{U} \times \mathbb{U} \to \mathbb{R}$ is called an edit distance defined on $\mathbb{U}$ if, for any pair of sequences A_1^p, B_1^q, the following recursive equation is satisfied

$$\delta(A_1^p, B_1^q) = \\ \text{Min} \begin{cases} \delta(A_1^{p-1}, B_1^q) + \Gamma(A(p) \to \Lambda) & del \\ \delta(A_1^{p-1}, B_1^{q-1}) + \Gamma(A(p) \to B(q)) & sub \\ \delta(A_1^p, B_1^{q-1}) + \Gamma(\Lambda \to B(q)) & ins \end{cases}$$

Note that not all edit/elastic distances are metric. In particular, the dynamic time warping distance does not satisfy the triangle inequality.

3.3.1 Levenshtein distance

The Levenshtein distance $\delta_{lev}(x,y)$ has been defined for string matching. For this edit distance, the *delete* and *insert* operations induce unitary costs, i.e. $\Gamma(A(p) \to \Lambda) = \Gamma(\Lambda \to B(q)) = 1$ while the *substitution* cost is null if $A(p) = B(q)$ or 1 otherwise.

3.3.2 Dynamic time warping

The DTW similarity measure δ_{dtw} [32][23] is defined according to the previous notations such as:

$$\begin{aligned}\delta_{dtw}(A_1^p, B_1^q) &= d_{LP}(a(p), b(q)) \\ &+ \text{Min} \begin{cases} \delta_{dtw}(A_1^{p-1}, B_1^q) \\ \delta_{dtw}(A_1^{p-1}, B_1^{q-1}) \\ \delta_{dtw}(A_1^p, B_1^{q-1}) \end{cases}\end{aligned}$$

where $d_{LP}(a(p), b(q)$ is the LP norm in $\mathbb{R}^k$, and so for DTW, $\Gamma(A(p) \rightarrow \Lambda) = \Gamma(A(p) \rightarrow B(q)) = \Gamma(\Lambda \rightarrow B(q)) = d_{LP}(a(p), b(q))$. Let us note that the time stamp values are not used, therefore the costs of each edit operation involve vectors a and b in S instead of vectors (a, t_a) and (b, t_b) in $S \times T$. One of the main restrictions of δ_{dtw} is that it does not comply with the triangle inequality as shown in [6].

3.3.3 Edit Distance with real penalty

$$\delta_{erp}(A_1^p, B_1^q) = \text{Min} \begin{cases} \delta_{erp}(A_1^{p-1}, B_1^q) + \Gamma(A(p) \rightarrow \Lambda) & ins \\ \delta_{erp}(A_1^{p-1}, B_1^{q-1}) + \Gamma(A(p) \rightarrow B(q)) & sub \\ \delta_{erp}(A_1^p, B_1^{q-1}) + \Gamma(\Lambda \rightarrow B(q)) & del \end{cases}$$

with

$$\begin{aligned}&\Gamma(A(p) \rightarrow \Lambda) = d_{LP}(a(p), g) \\ &\Gamma(A(p) \rightarrow B(q) = d_{LP}(a(p), b(q) \\ &\Gamma(\Lambda \rightarrow B(q)) = d_{LP}(g, b(q))\end{aligned}$$

where g is a constant in S and $d_{Lp}(x, y)$ is the Lp norm of vector $(x - y)$ in S.

Note that the time stamp coordinate is not taken into account, therefore δ_{erp} is a distance on S but not on $S \times T$. Thus, the cost of each edit operation involves vectors a and b in $\mathbb{R}^k$ instead of vectors (a, t_a) and (b, t_n) in $\mathbb{R}^{k+1}$.

According to the authors of ERP [6], the constant g should be set to 0 for some intuitive geometric interpretation and in order to preserve the mean value of the transformed time series when adding gap samples.

3.3.4 Time warp edit distance

Time Warp Edit Distance (TWED) [17], [18] is defined similarly to the edit distance defined for string [16][34]. The similarity between any two time series A and B of finite length, respectively p and q is defined as:

$$\delta_{twed}(A_1^p, B_1^q) = \text{Min} \begin{cases} \delta_{twed}(A_1^{p-1}, B_1^q) + \Gamma(A(p) \rightarrow \Lambda) & del_A \\ \delta_{twed}(A_1^{p-1}, B_1^{q-1}) + \Gamma(A(p) \rightarrow B(q)) & subs \\ \delta_{twed}(A_1^p, B_1^{q-1}) + \Gamma(\Lambda \rightarrow B(q)) & del_B \end{cases}$$

with

$$\begin{aligned}&\Gamma(A(p) \rightarrow \Lambda) = d(A(p), A(p-1)) + \lambda \\ &\Gamma(A(p) \rightarrow B(q)) = d(A(p), B(q)) + d(A(p-1), B(q-1)) \\ &\Gamma(\Lambda \rightarrow B(q)) = d(B(q), B(q-1)) + \lambda\end{aligned}$$

The time stamps are exploited to evaluate $d(A(p), B(q))$. In practice, $d(A(p), B(q)) = d_{LP}(a(p), b(q)) + \nu d_{LP}(t(p), t(q))$ where λ is a positive constant that represents a gap penalty and ν is a non negative constant which characterizes the *stiffness* of the δ_{twed} elastic measure.

3.4 Indefiniteness of elastic distance kernels

In appendix A, we give counter examples, one for each elastic distance we have previously defined, showing that these distances do not lead to definite kernels.

This demonstrates that the metric properties of a distance defined on $\mathbb{U}$, in particular the triangle inequality, are not sufficient conditions to establish definiteness (conditionally or not) of the associated distance kernel. One could conjecture that elastic distances cannot be definite (conditionally or not), possibly because of the presence of the $\max$ or $\min$ operators in the recursive equation. In the following sections, we will see that replacing these $\min$ or $\max$ operators by a sum operator allows, under some conditions, for the construction of series of positive definite kernels whose limit is quite directly connected to the previously addressed elastic distance kernels.

4 Constructing positive definite kernels from elastic distance

The main idea leading to the construction of positive definite kernels from a given elastic distance defined on $\mathbb{U}$ is to replace the $\min$ or $\max$ operator into the recursive equation defining the elastic distance by a $\sum$ operator. Instead of keeping one of the best alignment paths, the new kernel will sum up all the subsequence alignments with some weighting factor that could be optimized. This has been done successfully for the Smith and Waterman symbolic distance that is also known to be indefinite [22] and more recently for dynamic time warping kernels for time series alignment [9]. In the following sub sections, we propose some generalizations and extensions of these results that we confront in some time series classification experiments.

4.1 Recursive Time Warped Kernels

Definition 4.1: A function $\langle .;. \rangle : \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}$ is called a Recursive Time Warp Kernel (RTWK) if, for any pair of sequences A_1^p, B_1^q, there exists a function $f : \mathbb{R} \rightarrow \mathbb{R}$ such that the following recursive equation is satisfied:

$$\langle A_1^p, B_1^q \rangle = \sum \begin{cases} \langle A_1^{p-1}, B_1^q \rangle f(\Gamma(A(p) \rightarrow \Lambda)) & del \\ \langle A_1^{p-1}, B_1^{q-1} \rangle f(\Gamma(A(p) \rightarrow B(q))) & sub \\ \langle A_1^p, B_1^{q-1} \rangle f(\Gamma(\Lambda \rightarrow B(q))) & ins \end{cases} \quad (1)$$

This recursive definition requires to define an initialization. To that end we set $\langle \Omega, \Omega \rangle = \xi$, where ξ is a real constant, and Ω the null sequence.

Furthermore, this type of kernel sums up the multiplication of the local quantities $f(\Gamma(A(i) \rightarrow B(j)))$ for all the possible alignment paths between the two time series, the concept of alignment path being precisely defined in appendix B(sea definition B.2).

4.1.1 Definiteness of RTWK - Main result

The following theorem, related to the ones presented in [9] establishes sufficient conditions on $f(\Gamma(a \rightarrow b))$ for an RTWK to be definite and thus is a basis for the construction of definite RTWK. The proof of this theorem that is given in Appendix B is not based on the Hausslers convolution kernel [12] or it extension proposed by Shin et al. [27], and the sufficient conditions that are proposed here are less restrictive than those proposed by [9] and [27].

Theorem 4.2: Definiteness of RTWK:

If the local kernel $k(x,y) = f(\Gamma(x \rightarrow y))$ is positive definite on $((S \times T) \cup \{\Lambda\})^2$ and if $\xi > 0$, then the resulting RTWK is positive definite on $\mathbb{U} \times \mathbb{U}$.

A sketch of proof for theorem 4.2 is given in the appendix B.

As the cost function Γ is, in general, conditionally negative definite, choosing $f(h)$ for the exponential ensures that $f(\Gamma(x \rightarrow y))$ is a positive definite kernel [24].
Note here that Cuturi et al. [9] state that to ensure the definiteness of the DTW-RTWK kernel not only $f(\Gamma(x \rightarrow y))$ but also $f(\Gamma(x \rightarrow y))/(1 + f(\Gamma(x \rightarrow y)))$ need to be p.d. kernels which forms a stronger condition.
Other functions can be used, such as the Inverse Multi Quadric kernel $k(x,y) = \frac{1}{\sqrt{(\Gamma(x \rightarrow y))^2 + \theta^2}}$. As with the exponential (Gaussian or Laplace) kernel, the Inverse Multi Quadric kernel results in a positive definite matrix with full rank [19] and thus forms a infinite dimension feature space.

4.1.2 Computational cost of RTWK

Although the number of paths that are summed up exponentially increases with the lengths $|A|$ and $|B|$ of the time series that are evaluated, the recursive computation of $RTWK(A,B)$ leads to a quadratic computational cost $O(|A||B|)$, e.g. $O(N^2)$ if N is the average length of the time series that are considered. This quadratic complexity can be reduced to a linear complexity by limiting the number of alignment paths to consider in the recursion. This can be achieved when using a search corridor [23] as far as the kernel remains symmetric, which is the case when processing time series of equal lengths and restraining the search space using, for instance, a fixed size corridor symmetrically displayed around the diagonal as shown in Fig. 1.

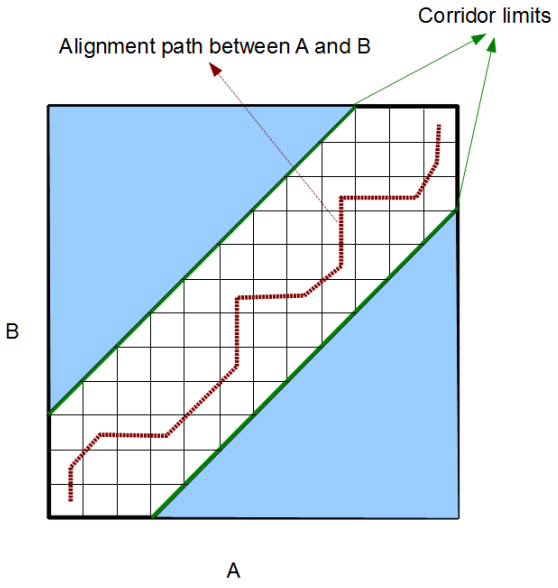

Fig. 1. Example of a symmetric corridor used to reduced the number of admissible alignment paths

4.2 *Exponentiated* RTWK

Definition 4.3:

$$\langle A_1^p, B_1^q \rangle_e = \sum \begin{cases} \langle A_1^{p-1}, B_1^q \rangle_e \, e^{-\nu' \Gamma(A(p) \rightarrow \Lambda)} \\ \langle A_1^{p-1}, B_1^{q-1} \rangle_e \, e^{-\nu' \Gamma(A(p) \rightarrow B(q))} \\ \langle A_1^p, B_1^{q-1} \rangle_e \, e^{-\nu' \Gamma(\Lambda \rightarrow B(q))} \end{cases} \quad (2)$$

where ν' is a *stiffness* parameter that weighs the contribution of the local elementary costs. The larger ν' is, the more the kernel is selective around the optimal paths. At the limit, when $\nu' \rightarrow \infty$, only the optimal path costs are summed up by the kernel. Note that, as is generally seen, several optimal paths leading to the same global cost exist, $\lim_{\nu' \rightarrow +\infty} -1/\nu' log(\langle A, B \rangle_e)$ does not coincide with the elastic distance δ that involves the same corresponding elementary costs.
The alignment cost of two null time series being 0, we suggest setting $\xi = 1$ in this exponentiation context.

Proposition 4.4: Definiteness of the exponentiated RTWK $\langle .,. \rangle_e$:
$\langle .,. \rangle_e$ is positive definite for the cost functions $\Gamma(A(p) \rightarrow \Lambda)$, $\Gamma(A(p) \rightarrow B(q))$ and $\Gamma(\Lambda \rightarrow B(q))$ involved in the computation of the δ_{lev}, δ_{dtw}, δ_{erp} and δ_{twed} distances.

The RTWKs constructed from these distances are referred respectively to $RTWK_{lev}$, $RTWK_{erp}$, $RTWK_{dtw}$, $RTWK_{twed}$ in the rest of the paper.

The proof of proposition 4.4 is straightforward and is omitted.

4.2.1 Interpretation of the exponentiated RTWK

For RTWK each alignment path is assigned with a cost that is the multiplication of the local cost functions attached to each edge of the path. For exponentiated RTWK, the local cost function, e.g. $e^{-\nu' \Gamma(A(p) \rightarrow B(q))}$ can be interpreted, up to a normalizing constant, as a probability to align symbol $A(p)$ with symbol $B(q)$, and the value affected to each path can be interpreted

as the probability of a specific alignment between two sequences. In that case the RTWK, that sums up the probability of all possible alignment paths between two sequences, can be interpreted as a matching probability between two sequences. This probabilistic interpretation suggests an analogy between RTWK and the *alpha-beta* algorithm designed to learn HMM models: while the Viterbi's algorithm that uses a *max* operator in a dynamic programming implementation (just like the *DTW* algorithm) evaluates only the probability of the best alignment path, the *alpha-beta* algorithm is based on the summation of the probabilities of all possible alignment paths. As reported in [22], the main drawbacks of these kind of kernels is the vanishing of the product of local cost functions (that are lower than one) when comparing long sequences. When considering gram-matrix (pair-wise distances on finite sets) this leads to a matrix that suffers from the diagonal dominance problem, i.e. the fact that the kernel value decreases extremely fast when the similarity slightly decreases.

5 Classification experiments

We empirically evaluate the effectiveness of some RTWK comparatively to Gaussian Radial Basis Function (RBF) Kernels or elastic distance substituting kernels [10] using some classification tasks on a set of time series coming from quite different application fields. The classification task we have considered consists of assigning one of the possible categories to an unknown time series for the 20 data sets available at the UCR repository [14]. As time is not explicitly given for these datasets, we used the index value of the samples as the time stamps for the whole experiment.

For each dataset, a training subset (TRAIN) is defined as well as an independent testing subset (TEST). We use the training sets to train two kinds of classifiers:

- the first one is a first near neighbor (1-NN) classifier: first we select a training data set containing time series for which the correct category is known. To assign a category to an unknown time series selected from a testing data set (different from the train set), we select its nearest neighbor (in the sense of a distance or similarity measure) within the training data set, then, assign the associated category to its nearest neighbor. For that experiment, a leave one out procedure is performed on the training dataset to optimized the meta parameters of the considered comparability measure.
- the second one is a SVM classifier [4], [31] configured with a Gaussian RBF kernel whose parameters are $C > 0$, a trade-off between regularization and constraint violation and σ that determines the width of the Gaussian function. To determine the C and σ hyper parameter values, we adopt a 5-folded cross-validation method on each training subset. According to this procedure, given a predefined training set TRAIN and a test set TEST, we adapt the meta parameters based on the training set TRAIN: we first divide TRAIN into 5 stratified subsets $TRAIN_1, TRAIN_2, , TRAIN_5$; then for each subset $TRAIN_i$ we use it as a new test set, and regard $(TRAIN - TRAIN_i)$ as a new training set; Based on the average error rate obtained on the five classification tasks, the optimal values of meta parameters are selected as the ones leading to the minimal average error rate.

We have used the LIBSVM library [5] to implement the SVM classifiers.

5.1 Experimenting with RTWK

We tested the exponentiated RTWK based on the $\delta_{erp}, \delta_{dtw}, \delta_{twed}$ distance costs. We consider respectively the positive definite $RTWK_{erp}, RTWK_{dtw}, RTWK_{twed}$ kernels.
Our experiment compares classification errors on the test data for

- the first near neighbor classifiers based on the $\delta_{erp}, \delta_{dtw}, \delta_{twed}$ distance measures (1-NN δ_{erp}, 1-NN δ_{dtw} and 1-NN δ_{twed}),
- the SVM classifiers using Gaussian distance substituting kernels based on the same distances and their corresponding RTWK, e.g. SVM δ_{erp}, SVM $RTWK_{erp}$, SVM δ_{dtw}, SVM $RTWK_{dtw}$, SVM δ_{twed}, SVM $RTWK_{twed}$.

For δ_{erp}, δ_{twed}, $RTWK_{erp}$ and $RTWKtwed$ we used the L1-norm, while the L2-norm has been implemented for δ_{dtw} and $RTWK_{dtw}$, a classical choice for DTW [21].

5.1.1 Meta parameters

For δ_{erp} kernel, meta parameter g is optimized for each dataset on the train data by minimizing the classification error rate of a first near neighbor classifier using a Leave One Out (LOO) procedure. For this kernel, g is selected in $\{-3, -2.99, -2.98, \cdots, 2.98, 2.99, 3\}$. This optimized value is also used for comparison with the $RTWK_e(ERP)$ kernel.

For δ_{twed} kernel, meta parameters λ and ν are optimized for each dataset on the train data by minimizing the classification error rate of a first near neighbor classifier. For our experiment, the *stiffness* value (ν) is selected from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ and λ is selected from $\{0, .25, .5, .75, 1.0\}$. If different (ν, λ) values lead to the minimal error rate estimated for the training data then the pairs containing the highest ν value are selected first, then the pair with the highest λ value is finally selected. These optimized (λ, ν)values are also used for comparability purposes with the $RTWK_{twed}$ kernel.

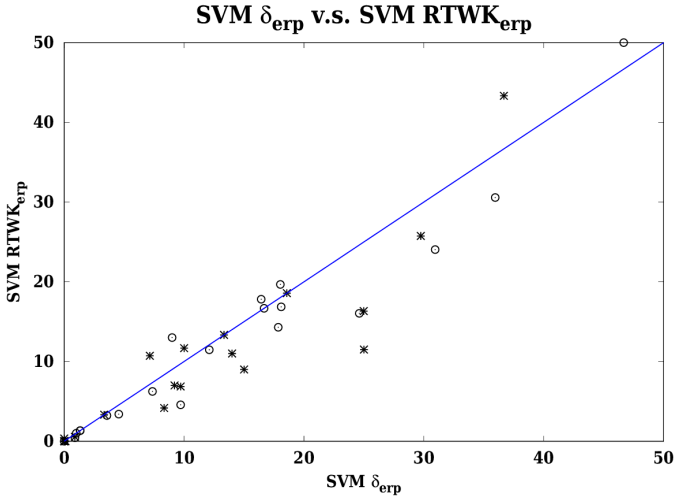

Fig. 2. Comparison of error rates (in %) between two SVM classifiers, the first one based on the δ_{erp} substituting kernel (SVM δ_{erp}), and the second one based on an additive time-warp kernel induced by the ERP distance (SVM $RTWK_{erp}$). The straight line has a slope of 1.0 and dots correspond, for the pair of classifiers, to the error rates on the train (star) or test (circle) data sets. A dot below (resp. above) the straight line indicates that SVM $RTWK_{erp}$ has a lower (resp. higher) error rate than distance SVM δ_{erp}

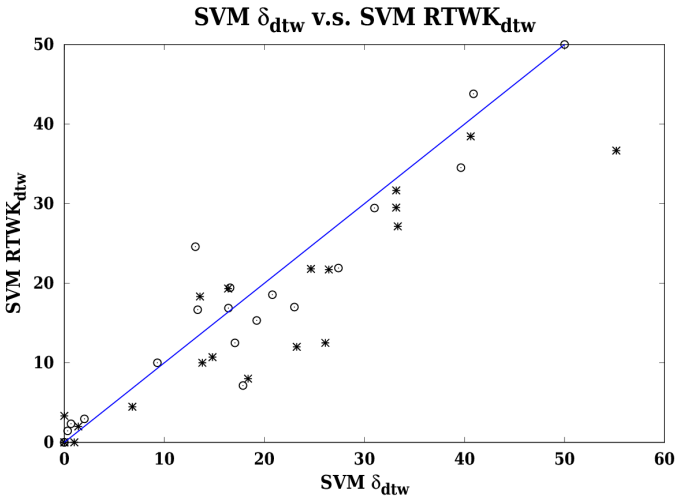

Fig. 3. Comparison of error rates (in %) between two SVM classifiers, the first one based on the δ_{dtw} substituting kernel (SVM δ_{dtw}), and the second one based on an additive time-warp kernel induced by the δ_{dtw} distance (SVM $RTWK_{dtw}$)). The straight line has a slope of 1.0 and dots correspond, for the pair of classifiers, to the error rates on the train (star) or test (circle) data sets. A dot below (resp. above) the straight line indicates that SVM $RTWK_{dtw}$ has a lower (resp. higher) error rate than distance δ_{dtw}

The kernels exploited by the SVM classifiers are the Gaussian Radial Basis Function (RBF) kernels $K(A,B) = e^{\delta(A,B)^2/(2\sigma^2)}$ where δ stands for $\delta_{erp}, \delta_{dtw}, \delta_{twed}$, $RTWK_{erp}(ERP$, $RTWK_{dtw}$, $RTWK_{twed}$. Meta parameter C is selected from $\{2^{-5}, 2^{-4}, ..., 1, 2, ..., 2^{10}\}$, and σ^2 from $\{2^{-5}, 2^{-4}, ..., 1, 2, ..., 2^{10}\}$. The best values are obtained using a cross validation procedure.

For the $RTWK_{erp}$, $RTWK_{dtw}$ and $RTWK_{twed}$ kernels, meta parameter $1/\nu'$ is selected from the discrete set $S = \{10^{-5}, 10^{-4}, ..., 1, 10, 100\}$.

The optimization procedure is as follows:

- for each value in S, we train a SVM $RTWK_*$ classifier on the training dataset using the previously described 5-folded cross validation procedure to select the SVM meta parameters *cost* and σ and the average of the classification error is recorded.
- the best σ, C and ν' values are the ones that lead to the minimal average error.

Table 1 gives for each data set and each tested kernel (δ_{erp}, δ_{dtw}, δ_{twed}, $RTWK_{erp}$, $RTWK_{dtw}$ and $RTWK_{twed}$) the corresponding optimized values of the meta parameters.

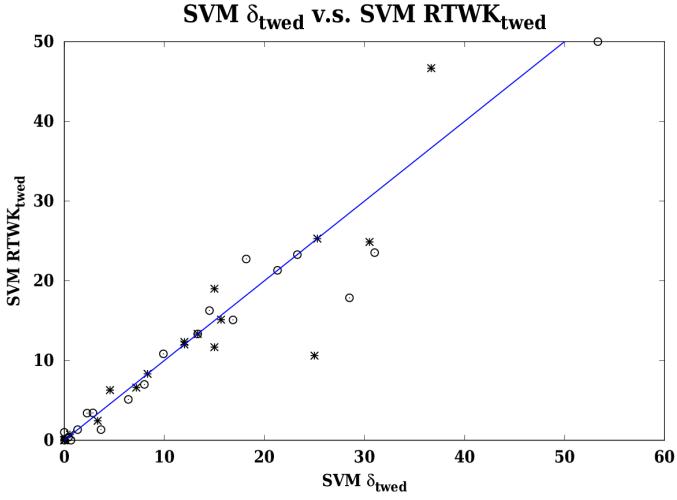

Fig. 4. Comparison of error rates (in %) between two SVM classifiers, the first one based on the δ_{twed} substituting kernel (SVM δ_{twed}), and the second one based on an additive time-warp kernel induced by the ERP distance (SVM $RTWK_{twed}$). The straight line has a slope of 1.0 and dots correspond, for the pair of classifiers, to the error rates on the train (star) or test (circle) data sets. A dot below (resp. above) the straight line indicates that SVM $RTWK_{twed}$ has a lower (resp. higher) error rate than distance SVM δ_{twed}

TABLE 1
Meta parameters used in conjunction with δ_{erp}, $RTWK_{erp}$, δ_{dtw}, $RTWK_{dtw}$, δ_{twed} and $RTWK_{twed}$ kernels

DATASET	$\delta_{erp} : g; C; \sigma$	$RTWK_{erp} : g; \nu'; C; \sigma$	$\delta_{dtw} : C; \sigma$	$RTWK_{dtw} : \nu'; C; \sigma$	$\delta_{twed} : \lambda; \nu; C; \sigma$	$RTWK_{twed} : \lambda; \nu; \nu'; C; \sigma$
Synth. cont.	0.0;2.0;0.25	0.0;0.457;256.0;0.062	8.0;4.0	0.047;1024.0;0.062	0.75;0.01;1.0;0.25	0.75;0.01;0.685;8.0;4.0
Gun-Point	-0.35;4.0;0.031	-0.35;0.457;128.0;1.0	16.0;0.0312	0.457;64.0;2.0	0.0;0.001;8.0;1.0	0.0;0.001;0.685;32;32
CBF	-0.11;1.0;1.0	-0.11;0.203;4.0;32.0	1.0;1.0	0.457;2.0;1.0	1.0;0.001;1.0;1.0	1.0;0.00;,0.20;4.0;32.0
Face (all)	-1.96;4.0;0.5	-1.96;1.028;8.0;0.62	2.0;0.25	1.028;4.0;0.25	1.0;0.01;8.0;4.0	1.0;0.01;2.312;8.0;4.0
OSU Leaf	-2.25;2.0;0.062	-2.25;1.541;256;0.031	4.0;0.062	1.541;32.0;0.062	1.0;1e-4;8.0;0.25	1.0;1e-4;1.028;64.0;1.0
Swed. Leaf	0.3;8.0;0.125	0.3;0.203;1.0;4.0	4.0;0.031	5.202;0.062;0.5	1.0;1e-4;16.0;0.062	1.0;1e-4;0.304;32.0;1.0
50 Words	-1.39;16.0;0.25	-1.39;0.685;16;0.25	4.0;0.062	1.028;64.0;0.062	1.0;1e-3;8.0;0.5	1.0;1e-3;1.028;32.0;2.0
Trace	0.57;32;0.62	0.57;0.457;256;4.0	4;0.25	0.685;16;0.25	0.25;1e-3;8.0;0.25	0.25;1e-3;300;0.0625;0.25
Two Patt.	-0.89;0.25;0.125	-0.89;0.304;0.004;1.0	0.25,0.125	0.457;2.0;0.125	1.0;1e-3;0.25;0.125	1.0;1e-3;0.685;0.25;0.125
Wafer	1.23;2.0;0.062	1.23;0.685;4.0;0.5	1.0;0.016	1.541;1024;0.031	1.0;0.125;4.0;0.62	1.0;0.125;1.541;1.0;4.0
face (four)	1.97;64;16	1.97;0.685;32;2	16;0.5	0.457;16;2	1.0;0.01;4;2	1.0;0.01;1.027;4;2
Ligthing2	-0.33;2;0.062	-0.33;2.312;128;0.062	2.0;0.031	1.541;32;0.062	0.0;1e-6;8;0.25	0.0;1e-6;1.541;8;8
Ligthing7	-0.40;128;2	-0.40;0.685;32;0.25	4;0.25	0.685;32;0.062	0.25;01;4;0.5	0.25;0.1;0.685;4;8
ECG	1.75;8;0.125	1.75;0.457;16;0.5	2;0.62	1.028;32;0.062	0.5;1.0;4;0.125	0.5;1.0;5.202;8;16
Adiac	1.83;16;0.0156	1.83;2.312;4096;0.031	16;0.0039	1.028;2048;0.031	0.75;1e-4;16;0.016	0.75;1e-4;2.312;128;1
Yoga	0.77;4;0.031	0.77;11.7054096;0.031	4;0.008	26.337;1024;0.031	0.5;1e-5;2;0.125	0.5;1e-5;3.468;256;2
Fish	-0.82;64;0.25	-0.82;0.685;32;0.5	8;0.016	3.468;64;16	0.5;1e-4;4;.5	0.5;1e-4;0.457;16;16
Coffee	-3.00;16;0.062	-3.00;26.337;4096;16	8;0.062	5.202;512;4	0;0.1;16;4	0;0.1;300;1024;128
OliveOil	-3.00;8;0.5	-0.82;0.457;256;0.062	2;0.125	0.457;32;0.125	0;0.001;256;32	0;0.001;32;32
Beef	-3.00;128;0.125	-3.00;0.685;0.004;16384	16;0.016	0.457;0.004;16	0;1e-4;2;1	0;1e-4;0.135;0.004;16

TABLE 2
Comparative study using the UCR datasets: classification error rates (in %) obtained using the first near neighbor classification rule and a SVM classifier for the erp, $RTWK_{erp}$, dtw and $RTWK_{dtw}$ kernels. Two scores are given S1|S2: the first one, S1, is evaluated on the training data, while the second one, S2, is evaluated on the test data.

DATASET	1-NN δ_{erp}	SVM δ_{erp}	SVM $RTWK_{erp}$	1-NN δ_{dtw}	SVM δ_{dtw}	SVM $RTWK_{dtw}$
Synthetic control	0.67|3.7	**0|1.33**	.33|**1.33**	1.0|0.67	**0**|2.33	**0|1**
Gun-Point	6.12|4	**0|1.33**	**0|1.33**	18.36|9.3	8|10	**0|1.33**
CBF	0|0.33	**3.33**|3.56	**3.33|3.22**	0|0.33	**3.33|1.44**	**3.33**|5.44
Face (all)	10.73|20.18	.89|18.1	**.54|16.86**	6.8|19.23	4.47|**15.32**	**.54**|16.98
OSU Leaf	30.15|40.08	25|35.95	**11.5|30.57**	33.17|40.9	29.5|43.8	**20|23.55**
Swedish Leaf	11.02|12	9.2|7.36	**7|6.24**	24.65|20.8	21.8|18.56	**7|5.6**
50 Words	19.38|28.13	24.98|24.61	**16.32|16.04**	33.18|31	31.66|29.45	**15.21|17.58**
Trace	10.01|17	**0|1**	**0|1**	0|0	**0|0**	**0**|2
Two Patterns	0|0	**0|0**	**0|0**	0|0	**0|0**	**0|0**
Wafer	.1|0.9	.1|0.89	**0|0.44**	1.4|2.01	2|2.95	**0|0.39**
face (four)	4.35|10.2	8.33|4.55	**4.17|3.4**	26.09|17.05	12.5|12.5	**8.33|5.68**
Ligthing2	11.86|14.75	**10|18.03**	11.67|19.67	13.56|13.1	18.33|24.59	**8.33|19.67**
Ligthing7	23.19|30.1	**18.57|16.43**	**18.57**|17.81	33.33|27.4	27.15|21.91	**17.14|16.43**
ECG	10.01|13	15|**9**	**9**|13	23.23|23	12|17	**7|13**
Adiac	35.99|37.85	29.74|30.94	**25.74|24.04**	40.62|39.64	38.46|34.52	**24.61|25.32**
Yoga	14.05|14.7	14|12.1	**11|11.47**	16.37|16.4	19.33|16.87	**11|11.2**
Fish	16.09|12	9.71|9.71	**6.86|4.57**	26.44|16.57	21.72|19.43	**6.86|4.57**
Coffee	25.93|25	**7.14**|17.85	10.71|**14.29**	14.81|17.86	**10.71|7.14**	**10.71**|17.86
OliveOil	17.24|16.67	**13.33|16.67**	**13.33|16.67**	13.79|13.33	**10|16.67**	13.33|**16.67**
Beef	68.97|50	**36.67|46.67**	43.33|50	55.17|50	36.67|50	**32.14|42.85**
# Best Scores	-	10/9	**16/16**	-	6/6	**19/16**
# Uniquely Best Scores	-	4/4	**10/11**	-	1/4	**14/14**

5.2 Discussion

5.2.1 RTWK experiment analysis

Tables 2 and 3 show the classification error rates obtained for the tested methods, e.g. the first near neighbor classifier based on the δ_{erp}, δ_{dtw} and δ_{twed} distances (1-NN δ_{erp}, 1-NN δ_{dtw} and 1-NN δ_{twed}), the Gaussian RBF kernel SVM based on the same distances (SVM δ_{erp}, SVM δ_{dtw} and SVM δ_{twed}) and euclidean distance and the Gaussian RBF kernel SVM based on the RTWK kernels (SVM $RTWK_{erp}$, SVM $RTWK_{dtw}$ and SVM $RTWK_{twed}$).

In this experiment, we show that the SVM classifiers clearly outperform the 1-NN classifiers. But the interesting results reported in tables 2 and 3 and figures 2, 3 and 4 is that SVM $RTWK_{erp}$ and SVM $RTWK_{twed}$ perform slightly better than SVM δ_{erp} and SVM δ_{twed} respectively, and the SVM $RTWK_{dtw}$ is clearly much efficient than the SVM δ_{dtw}. This could come from the fact that δ_{erp} and δ_{twed} are metrics but not δ_{dtw}. SVM δ_{dtw} behaves poorly compared to the other tested classifiers probably because the SVM optimization process does not perform well. Nevertheless, the $RTWK_{dtw}$ kernel based on δ_{dtw} seems to correct greatly its drawbacks. To explore further the potential impact of indefiniteness on classification

rates, we give in 4 two quantified hints of deviation to conditionally definiteness for the gram-matrices corresponding to the δ_{dtw}, δ_{erp} and δ_{twed} distances. Since to be conditionally definite (negative) a gram-matrix should have a single positive eigenvalue, the first hint is the number of positive eigenvalues $\#Pev$ (we give also as a reference the total number of eigenvalues, $\#Ev$). The second hint, $\Delta_p = 100 * \frac{\sum_{ev_i>0}(ev_i) - \text{ArgMax}_{ev_i>0}\{ev_i\}}{\sum_{ev_i>0} ev_i}$, where ev_i is an eigenvalue of the gram matrix, quantifies the weight of the extra positive eigenvalues relatively to the weight of the total number of positive eigenvalues. Therefore, a conditionally definite (negative) gram-matrix should be such that simultaneously $\#Pev = 0$ and $\Delta_p = 0$. By examining the gram-matrices corresponding to each training datasets and for each distances $\delta_{dtw}(A, B)$, $\delta_{erp}(A, B)$ and $\delta_{twed}(A, B)$, we can show that the δ_dtw kernel is much more far away from a conditionally definite matrix than the δ_{erp} and δ_{twed} kernels. The distance that is closer to conditional definiteness is the δ_{twed} distance. This is clearly measurable by the number of positive eigenvalues and their amplitudes. Furthermore, for datasets of small sizes (such as *CBF*, *Beef*, *Coffee*, *OliveOil*, etc.), δ_{erp} and δ_{twed} kernels produce conditionally definite Gram-matrices where δ_dtw does not. The regularization brought by RTWK is therefore more effective on δ_dtw. This is the case for instance on the *Beef* dataset for which, on the train data, the δ_{twed} performs slightly better than the $RTWK_{twed}$. In this case, both kernels lead to a definite gram-matrix, and the extra parameter ν' in use in the $RTWK_{twed}$ kernel explains probably a poorer classification rate due to a lack of learning data. Nevertheless, similarly to the additive RTWK, some datasets are better classified by SVM that use directly the distance kernel instead of the derived RTWK kernel. The same reasons mentioned above in the case of additive RTWK can be invoked here also for such cases. The extra parameter ν' makes the search for an optimal setting on the train data more difficult and requires more learning data to converge. The trade-off between learning and generalization is therefore even more complex.

6 Conclusion

Following the work on convolution kernels [12] and local alignment kernels defined for string processing around the Smith and Waterman algorithm [29] [22] or defined for time series on the basis of the DTW similarity measure [9], we have addressed the definiteness of elastic distances through a direct constructive approach from the recursive definitions of elastic (editing) distances themselves and achieve some extension of these previous results. In particular our sufficient conditions for definiteness are less restrictive. They apply to a wider range of elastic distances. The recursive time-warp kernels (RTWK) we have studied are applicable for string and time series processing. We give some simple sufficient conditions to build positive definite RTWK that apply for the exponentiated version that we have tested: these conditions are basically satisfied by classical elastic distances defined by a recursive equation. In particular this is true for the edit distance, the well known Dynamic Time Warping measure and for some variants such as the Edit Distance With Real penalty and the Time Warp Edit Distance, the latter two being metrics as well as the symbolic edit distance. The experiments conducted on a variety of time series datasets show that the positive definite RTWKs outperforms the indefinite elastic distances they are derived from when considering 1-NN and SVM classification tasks.

Appendix A
Indefiniteness of classical elastic measures

A.1 The Levenshtein distance

The Levenshtein distance kernel $\varphi(x, y) = \delta_{lev}(x, y)$ is known to be indefinite. Below, we discuss the first known counter-example produced by [7]. Let us consider the subset of sequences $V = \{abc, bad, dab, adc, bcd\}$ that leads to the following distance matrix

$$M_{lev}^V = \begin{pmatrix} 0 & 3 & 2 & 1 & 2 \\ 3 & 0 & 2 & 2 & 1 \\ 2 & 2 & 0 & 3 & 3 \\ 1 & 2 & 3 & 0 & 3 \\ 2 & 1 & 3 & 3 & 0 \end{pmatrix} \tag{3}$$

and consider coefficient vectors C and D in $\mathbb{R}^5$ such that
$C = [1, 1, -2/3, -2/3, -2/3]$ with $\sum_{i=1}^5 c_i = 0$ and
$D = [1/3, 2/3, 1/3, -2/3, -2/3]$ with $\sum_{i=1}^5 d_i = 0$.

Clearly $CM_{lev}^V C^T = 2/3 > 0$ and $DM_{lev}^V D^T = -4/3 < 0$, showing that M_{lev}^V has no definiteness.

A.2 The Dynamic Time Warping distance

The DTW kernel $\varphi(x, y) = \delta_{dtw}(x, y)$ is also known not to be conditionally definite. The following example demonstrates this known result. Let us consider the subset of sequences $V = \{01, 012, 0122, 01222\}$.

Then the DTW distance matrix evaluated on V is

$$M_{dtw}^V = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \end{pmatrix} \tag{4}$$

and consider coefficient vectors C and D in $\mathbb{R}^4$ such that
$C = [1/4, -3/8, -1/8, 1/4]$ with $\sum_{i=1}^4 c_i = 0$ and
$D = [-1/4, -1/4, 1/4, 1/4]$ with $\sum_{i=1}^4 d_i = 0$. Clearly $CM_{dtw}^V C^T = 2/32 > 0$ and $DM_{dtw}^V D^T = -1/2 < 0$, showing that M_{dtw}^V has no definiteness.

A.3 The Time Warp Edit Distance

Similarly, it is easy to find simple counter examples that show that TWED kernels are not definite.

Let us consider the subset of sequences $V = \{010, 012, 103, 301, 032, 123, 023, 003, 302, 321\}$.

For the TWED metric, with $\nu = 1.0$ and $\lambda = 0.0$ we get the following matrix:

$$M^V_{twed} = \begin{pmatrix} 0 & 2 & 7 & 9 & 6 & 7 & 5 & 5 & 10 & 9 \\ 2 & 0 & 5 & 9 & 4 & 5 & 3 & 3 & 8 & 9 \\ 7 & 5 & 0 & 6 & 7 & 4 & 6 & 2 & 5 & 10 \\ 9 & 9 & 6 & 0 & 13 & 10 & 12 & 8 & 1 & 4 \\ 6 & 4 & 7 & 13 & 0 & 5 & 3 & 5 & 12 & 9 \\ 7 & 5 & 4 & 10 & 5 & 0 & 2 & 6 & 9 & 6 \\ 5 & 3 & 6 & 12 & 3 & 2 & 0 & 4 & 11 & 8 \\ 5 & 3 & 2 & 8 & 5 & 6 & 4 & 0 & 7 & 10 \\ 10 & 8 & 5 & 1 & 12 & 9 & 11 & 7 & 0 & 5 \\ 9 & 9 & 10 & 4 & 9 & 6 & 8 & 10 & 5 & 0 \end{pmatrix} \quad (5)$$

The eigenvalue spectrum for this matrix is the following:
$\{$ 4.62, 0.04, -2.14, -0.98, -0.72, -0.37, -0.19, -0.17, -0.06, -0.03 $\}$. This spectrum contains 2 strictly positive eigenvalues, showing that M^V_{twed} has no definiteness.

A.4 The Edit Distance with Real Penalty

For the ERP metric, with $g = 0.0$ we get the following matrix:

$$M^V_{erp} = \begin{pmatrix} 0 & 2 & 3 & 3 & 4 & 5 & 4 & 2 & 4 & 5 \\ 2 & 0 & 3 & 5 & 2 & 3 & 2 & 2 & 4 & 5 \\ 3 & 3 & 0 & 4 & 3 & 2 & 3 & 1 & 3 & 4 \\ 3 & 5 & 4 & 0 & 7 & 6 & 7 & 5 & 1 & 2 \\ 4 & 2 & 3 & 7 & 0 & 3 & 2 & 2 & 6 & 5 \\ 5 & 3 & 2 & 6 & 3 & 0 & 1 & 3 & 5 & 4 \\ 4 & 2 & 3 & 7 & 2 & 1 & 0 & 2 & 6 & 5 \\ 2 & 2 & 1 & 5 & 2 & 3 & 2 & 0 & 4 & 5 \\ 4 & 4 & 3 & 1 & 6 & 5 & 6 & 4 & 0 & 1 \\ 5 & 5 & 4 & 2 & 5 & 4 & 5 & 5 & 1 & 0 \end{pmatrix} \quad (6)$$

The eigenvalue spectrum for this matrix is the following:
$\{$ 4.63, 0.02, $1.39e-17$, -2.21, -0.97, -0.56, -0.41, -0.26, -0.17, -0.08 $\}$. This spectrum contains 3 strictly positive eigenvalues (although the third positive eigenvalue which is very small could be the result of the imprecision of the used diagonalization algorithm), showing that M^V_{erp} has no definiteness.

APPENDIX B
PROOF OF OUR MAIN RESULT

B.1 Proof of theorem 4.2

Let $\kappa_{i\to j}(A,B)$, $\kappa_{\Lambda\to j}(A,B)$ and $\kappa_{i\to\Lambda}(A,B)$ be the extensions of local functions associated to the editing operation costs from $\mathbb{U}^2$ to $\mathbb{R}^+$ defined as follows:

- $\forall(A,B) \in \mathbb{U}^2$, $\kappa_{i\to j}(A,B) = f(\Gamma(A(i) \to B(j)))$ if $0 \le i \le |A|$ and $0 \le j \le |B|$, $\kappa_{i\to j}(A,B) = \xi$ otherwise.
- $\forall(A,B) \in \mathbb{U}^2$, $\kappa_{\Lambda\to j}(A,B) = f(\Gamma(\Lambda \to B(j)))$ if $0 \le j \le |B|$, $\kappa_{\Lambda\to j}(A,B) = \xi$ otherwise.
- $\forall(A,B) \in \mathbb{U}^2$, $\kappa_{i\to\Lambda}(A,B) = f(\Gamma(A(i) \to \Lambda))$ if $0 \le i \le |A|$, $\kappa_{i\to\Lambda}(A,B) = \xi$ otherwise.

We define also for convenience sake the value of the $null$ local operation as: $\forall(A,B) \in \mathbb{U}^2$, $\kappa_{null}(A,B) = \xi$.

We define the symmetrization for these extensions as follows:

- $\widetilde{\kappa}_{i\to j}(A,B) = 1/2.(\kappa_{i\to j}(A,B) + \kappa_{i\to j}(B,A))$
- $\widetilde{\kappa}_{i\to\Lambda}(A,B) = 1/2.(\kappa_{i\to\Lambda}(A,B) + \kappa_{i\to\Lambda}(B,A))$
- $\widetilde{\kappa}_{\Lambda\to j}(A,B) = 1/2.(\kappa_{\Lambda\to j}(A,B) + \kappa_{\Lambda\to j}(B,A))$
- and $\widetilde{\kappa}_{null}(A,B) = \kappa_{null}(A,B)$

Lemma B.1: $\widetilde{\kappa}_{i\to j}$, $\widetilde{\kappa}_{i\to\Lambda}$, $\widetilde{\kappa}_{\Lambda\to j}$ and $\widetilde{\kappa}_{null}$ are p.d. kernels on $\mathbb{U}^2$ iff $f(x,y) = f(\Gamma(x \to y)) : ((S \times T) \cup \{\Lambda\})^2 \to \mathbb{R}$ is positive definite.

The proof of Lemma **??** is trivial and is omitted.

Definition B.2: Let π be an ordered alignment map between two finite non empty sets of integers. Basically π a finite sequence of pairs of integers $\pi(l) = (i_l, j_l)$ for $l \in \{1, ..., |\pi|\}$, satisfying the conditions:

- $i_{l-1} \le i_l, \forall l \in 2, .., |\pi|$
- $j_{l-1} \le j_l, \forall l \in 2, .., |\pi|$
- $i_{l-1} < i_l$ or $j_{l-1} < j_l, \forall l \in \{2, .., |\pi|\}$

$\pi(l)_1 = i_l$ and $\pi(l)_2 = j_l$ are the two coordinate access functions for the l^{th} pair of mapped integers.

Let $\mathcal{M}_n$ be the set of alignment maps π such that the integers mapped by π are all lower or equal to n.

Algorithm 1 provides a constructive way to define uniquely, for all n and all map π in $\mathcal{M}_n$, a finite sequence of local editing operations (insertion, substitution or deletion), also called alignment path, γ_π that defines a transformation on the set of sequences $\mathbb{U}$.

Given any finite mapping π, the global alignment function of two sequences A and B is defined as

$$K_\pi(A,B) = \xi \prod_{l=1\cdots|\gamma_\pi|} \kappa_{\gamma_\pi(l)}(A,B) \quad (7)$$

where $\xi = 1$ in general.

We can construct a symmetric kernel closely related to $K_\pi(A,B)$ (see Proposition B.3) as follows

$$\widetilde{K}_{\pi,n}(A,B) = K_\pi(A,B) = \xi \prod_{l=1\cdots|\gamma_\pi|} \widetilde{\kappa}_{\gamma_\pi(l)}(A,B) \quad (8)$$

Proposition B.3: If $n \geq min(|A|,|B|)$, then

$$\forall n, \sum_{\pi\in\mathcal{M}_n} \widetilde{K}_{\pi,n}(A,B) = \sum_{\pi\in\mathcal{M}_n} K_{\pi,n}(A,B) \quad (9)$$

The proof of proposition B.3 can easily be obtained by recursion and is omitted.

Algorithm 1 Uniquely provides a sequence of editing operations given n and an alignment map $\pi \in \mathcal{M}$. This sequence of editing operations transforms any sequence A in $\mathbb{U}$ in any sequence B in $\mathbb{U}$. We refer it as γ_π

Require: π, an alignment map
  $l \leftarrow 1$, $p \leftarrow 1$
  $i_l \leftarrow 0$, $j_l \leftarrow 0$
  while $l \leq |\pi|$ **do**
    while $i_l < \pi(l)_1$ **do**
      if $(i_l < n)$ **then**
        $\gamma_\pi(p) = i_l \rightarrow \Lambda$ // deletion of $A(i_l)$
      else
        $\gamma_\pi(p) = null$;
      end if
      $p \leftarrow p+1$, $i_l \leftarrow i_l + 1$
    end while
    while $j_l < \pi(l)_2$ **do**
      if $(j_l < n)$ **then**
        $\gamma_\pi(p) = \Lambda \rightarrow j_l$ // insertion of $B(j_l)$
      else
        $\gamma_\pi(p) = null$;
      end if
      $p \leftarrow p+1$, $j_l \leftarrow j_l + 1$
    end while
    if ($i_l < |A|$ and $j_l < |B|$) **then**
      $\gamma_\pi(p) = i_l \rightarrow j_l$ // substitution of of $A(i_l)$ by $B(j_l)$
    else if $(i_l < |A|)$ **then**
      $\gamma_\pi(p) = i_l \rightarrow \Lambda$ // deletion of $A(i_l)$
    else if $(j_l < |B|)$ **then**
      $\gamma_\pi(p) = \Lambda \rightarrow j_l$ // insertion of $B(j_l)$
    else
      $\gamma_\pi(p) = null$;
    end if
    $p \leftarrow p+1$, $l \leftarrow l+1$
    $i_l \leftarrow i_l + 1$, $j_l \leftarrow j_l + 1$
  end while
  return γ_π

Lemma B.4: $\mathcal{P}$: For all finite mapping π sufficient conditions for $\widetilde{K}_\pi$ to be a p.d. kernel on $\mathbb{U}^2$ are

- $f(x,y) = f(\Gamma(x \rightarrow y)) : ((S\times T)\cup\{\Lambda\})^2 \rightarrow \mathbb{R}$ is positive definite.
- $\xi > 0$

We prove lemma B.4 by induction on $r = |\gamma_\pi|$.

i) For a mapping π for which $|\gamma_\pi| = 0$ (π is the void mapping), $\widetilde{K}_\pi(A_u, A_v) = \xi$, for all A_u, A_v in $\mathbb{U}^2$. Hence, since $\xi > 0$, $\widetilde{K}_\pi$ is p.d. at rank 0, and $\mathcal{P}$ is verified at rank 0.

ii) Let us assume that $\mathcal{P}$ is verified at rank r, and consider a mapping π such that $|\gamma_\pi| = r+1$. Then for all A_u, A_v in $\mathbb{U}^2$,

$$\begin{aligned}\widetilde{K}_\pi(A_u,A_v) &= \xi \textstyle\prod_{l=1\cdots r} \widetilde{\kappa}_{\gamma_\pi(l)}(A_u,A_v) \\ &= \left(\xi \textstyle\prod_{l=1\cdots r} \widetilde{\kappa}_{\gamma_\pi(l)}(A_u,A_v)\right) \widetilde{\kappa}_{\gamma_\pi(r)}(A_u,A_v) \\ &= \widetilde{K_r}(A_u,A_v)\widetilde{\kappa}_{\gamma_\pi(r)}(A_u,A_v)\end{aligned}$$

with $\widetilde{K_r}(A_u,A_v) = \xi \prod_{l=1\cdots r} \widetilde{\kappa}_{\gamma_\pi(l)}(A_u,A_v)$.

By induction hypothesis, $\widetilde{K_r}(A_u,A_v)$ p.d., so is $\widetilde{\kappa}_{\gamma_\pi(r)}(A_u,A_v)$ according to lemma B.1. The closure properties under the multiplication of p.d. kernels establishes that the proposition $\mathcal{P}$ is true at rank $r+1$.

iii) By induction the proposition $\mathcal{P}$ is proved for all alignment path γ_π (or equivalently for all finite mapping π), which establishes lemma B.4 □.

Proposition B.5: For any pair $(A,B) \in \mathbb{U}^2$, any $(i,j) \in \mathbb{N}^2$ such that $0 \leq i \leq |A|$ and $0 \leq j \leq |B|$ we have

$$\langle A_1^i, B_1^j\rangle = \sum_{\gamma\in\mathcal{E}(A_1^i,B_1^j)} \xi \prod_{l=1\cdots|\gamma|} \kappa_{\gamma(l)}(A_1^i, B_1^j) \quad (10)$$

where $\mathcal{E}(A,B)$ is the set of finite sequences of local editing operations allowing to transform sequence A into sequence B.

We prove proposition B.5 by induction on $r = |A|+|B|$. The proposition is true for $r = 0$, since then $i = j = 0$ and then $\langle A_1^0, B_1^0\rangle = \langle \Lambda, \Lambda\rangle = \xi$.

The proposition B.5 is also true for $r = 1$, since

- if $|B| = 1$, $\langle A_1^0, B_1^1\rangle = \langle \Lambda,\Lambda\rangle f(\Gamma(\Lambda \rightarrow B(1)))$
  $= \xi\kappa_{\Lambda\rightarrow 1}(A_1^0, B_1^1)$,
- similarly, if $|A| = 1$, $\langle A_1^1, B_1^0\rangle = \langle \Lambda,\Lambda\rangle . f(\Gamma(A(1) \rightarrow \Lambda))$
  $= \xi\widetilde{\kappa}_{1\rightarrow\Lambda}(A_1^1, B_1^0)$.

Let suppose proposition B.5 is true for all $m \leq r$ for $r \geq 1$ and let show that it is true for $r+1$.

First case: If $i > 0$ and $j > 0$, then by definition of $\langle .,. \rangle$ we have

$$\begin{aligned}\langle A_1^i, B_1^j\rangle = &\langle A_1^{i-1}, B_1^j\rangle f(\Gamma(A(i) \rightarrow \Lambda)) \\ &+\langle A_1^{i-1}, B_1^{j-1}\rangle f(\Gamma(A(i) \rightarrow B(j))) \\ &+\langle A_1^i, B_1^{j-1}\rangle f(\Gamma(\Lambda \rightarrow B(j))).\end{aligned} \quad (11)$$

which rewrites

$$\begin{aligned}\langle A_1^i, B_1^j\rangle &= \langle A_1^{i-1}, B_1^j\rangle \kappa_{\Gamma(A(i)\to\Lambda)}(A_1^i, B_1^j) \\ &+\langle A_1^{i-1}, B_1^{j-1}\rangle \kappa_{\Gamma(A(i)\to B(j))}(A_1^i, B_1^j) \\ &+\langle A_1^i, B_1^{j-1}\rangle \kappa_{\Gamma(\Lambda\to B(j))}(A_1^i, B_1^j).\end{aligned} \quad (12)$$

The three terms $\langle A_1^{i-1}, B_1^j\rangle$, $\langle A_1^{i-1}, B_1^{j-1}\rangle$ and $\langle A_1^i, B_1^{j-1}\rangle$ in the right hand side of the previous equality enter into the inductive hypothesis and thus decomposes as follows:

$$\begin{aligned}\langle A_1^{i-1}, B_1^j\rangle &= \sum_{\gamma\in\mathcal{E}(A_1^{i-1}, B_1^j)} \xi . \prod_{l=1\cdots|\gamma|} \kappa_{\gamma(l)}(A_1^i, B_1^j) \\ \langle A_1^{i-1}, B_1^{j-1}\rangle &= \sum_{\gamma\in\mathcal{E}(A_1^{i-1}, B_1^{j-1})} \xi . \prod_{l=1\cdots|\gamma|} \kappa_{\gamma(l)}(A_1^i, B_1^j) \quad (13) \\ \langle A_1^{i}, B_1^{j-1}\rangle &= \sum_{\gamma\in\mathcal{E}(A_1^{i}, B_1^{j-1})} \xi . \prod_{l=1\cdots|\gamma|} \kappa_{\gamma(l)}(A_1^i, B_1^j)\end{aligned}$$

Recombining equations 12 and 13 and completing the editing sequences with the three local editing operations we get the expected decomposition for $\langle A_1^i, B_1^j\rangle$.

Second case: If $i = r$ and $j = 0$, then by definition of $\langle ., .\rangle$ we have necessarily

$$\begin{aligned}\langle A_1^i, B_1^j\rangle &= \xi . f(\Gamma(A(1)\to\Lambda))\cdots f(\Gamma(A(r)\to\Lambda)) \\ &= \xi \prod_{l=1\cdots|\gamma|} \kappa_{A(l)\to\Lambda}(\langle A_1^i, B_1^j\rangle)\end{aligned}$$

which leads to the expected decomposition (a single editing sequence exists in that case).

Third case: If $i = 0$ and $j = r$, the result is obtained similarly to the second case.

Therefore proposition B.5 is true for $r + 1$. By induction, proposition B.5 is true for all $r \in \mathbb{N}^+ \cup \{0\}$ □

Proof of Proposition 4.2 (Definiteness of RTWK)

Proposition B.5 states that $\forall (A, B) \in \mathbb{U}^2$,

$$\langle A, B\rangle = \sum_{\gamma\in\mathcal{E}(A,B)} \xi \prod_{l=1\cdots|\gamma|} \kappa_{\gamma(l)}(A, B)$$

It is easy to establish that $\forall (A, B) \in \mathbb{U}^2$,

$$\langle A, B\rangle = \sum_{\pi\in\mathcal{M}_n} \xi \prod_{l=1\cdots|\gamma_\pi|} \kappa_{\gamma_\pi(l)}(A, B)$$

whenever $n \geq min\{|A|, |B|\}$, leading to (proposition B.3):

$$\langle A, B\rangle = \sum_{\pi\in\mathcal{M}_n} K_{\pi,n}(A, B) = \sum_{\pi\in\mathcal{M}_n} \widetilde{K}_{\pi,n}(A, B)$$

whenever $n \geq min\{|A|, |B|\}$.

Now, let $\mathbb{U}_n$ be the set of sequences in $\mathbb{U}$ whose lengths are $\leq n$. Since, according to lemma B.4, $\widetilde{K}_{\pi,n}$ is a p.d. kernel for all $\pi \in \mathcal{M}_n$, according to the closure properties under the addition of p.d. kernels, we show that $\langle A, B\rangle$ is a p.d. kernel defined on $\mathbb{U}_n^2$.

As this result is true for all n, one can sees $\langle A, B\rangle$ as a point-wise limit as n tends toward infinity of a p.d. kernel defined on $\mathbb{U}_n^2$. This establishes that , $B\rangle$ is a p.d. kernel on $\mathbb{U}^2$ □.

ACKNOWLEDGMENTS

We would like to thank the French Ministry of Research, the Brittany Region, the General Council of Morbihan and the European Regional Development Fund that had partially fund this research.

REFERENCES

[1] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. A basic local alignment serach tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[2] R. Bellman. *Dynamic Programming*. Princeton Univ Press, 1957. New Jersey.

[3] Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*, volume 100 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, April 1984.

[4] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, pages 144–152, 1992.

[5] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001.

[6] L. Chen and R. Ng. On the marriage of lp-norm and edit distance. In *Proceedings of the 30th International Conference on Very Large Data Bases*, pages 792–801, 2004.

[7] Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Positive Definite Rational Kernels. In *Proceedings of COLT'03*, volume 2777 of *Lecture Notes in Computer Science*, pages 41–56, Washington D.C., August 2003. Springer, Heidelberg, Germany.

[8] Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Rational kernels: Theory and algorithms. *Journal of Machine Learning Research*, 5:1035–1062, 2004.

[9] M. Cuturi, J-P. Vert, O. Birkenes, and T. Matsui. A Kernel for Time Series Based on Global Alignments. In *Proceedings of ICASSP'07*, Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, pages II–413 – II–416, Honolulu, HI, April 2007. IEEE.

[10] B. Haasdonk and C. Bahlmann. Learning with distance substitution kernels. In *DAGM-Symposium*, pages 220–227, 2004.

[11] Bernard Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:482–492, 2005.

[12] D. Haussler. Convolution kernels on discrete structures. Technical report, University of California, Santa Cruz, 2008. Technical Report.

[13] Akira Hayashi, Yuko Mizuhara, and Nobuo Suematsu. Embedding time series data for classification. In *MLDM*, pages 356–365, 2005.

[14] E. J. Keogh, X. Xi, L. Wei, and C.A. Ratanamahatana. The ucr time series classification-clustering datasets, 2006. http://wwwcs.ucr.edu/ eamonn/time_series_data/.

[15] Karthik Kumara, Rahul Agrawal, and Chiranjib Bhattacharyya. A large margin approach for writer independent online handwriting classification. *Pattern Recogn. Lett.*, 29:933–937, May 2008.

[16] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR, 163(4):845-848, 1965 (Russian)*, pages 707–710, 1966. English translation in Soviet Physics Doklady, 10(8).

[17] P. F. Marteau. Time warp edit distance. Technical report, VALORIA, Universite de Bretagne Sud, 2008. Technical Report valoriaUBS-2008-3v, http://hal.archives-ouvertes.fr/hal-00258669/fr/.

[18] P. F. Marteau. Time warp edit distance with stiffness adjustment for time series matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):306–318, 2009.
[19] C. A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive functions. *Constructive Approximation*, 2:11–22, 1986.
[20] W. Pearson. Rapid and sensitive sequence comparisons with fasp and fasta. *Methods Enzymol*, 183:63–98, 1990.
[21] C. A. Ratanamahatana and E. J. Keogh. Making time-series classification more accurate using learned constraints. In *Proceedings of the Fourth SIAM International Conference on Data Mining (SDM'04)*, pages 11–22, 2004.
[22] H. Saigo, J.P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20:1682–1689, 2004.
[23] H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. In *Proceedings of the 7th International Congress of Acoustic*, pages 65–68, 1971.
[24] I. J. Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536, nov 1938.
[25] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
[26] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.
[27] Kilho Shin and Tetsuji Kuboyama. A generalization of haussler's convolution kernel: mapping kernel. In *ICML*, pages 944–951, 2008.
[28] K.R. Sivaramakrishnan and C. Bhattacharyya. Time series classification for online tamil handwritten character recognition a kernel based approach. In Nikhil R. Pal, Nikola Kasabov, Rajani K. Mudi, Srimanta Pal, and Swapan K. Parui, editors, *Neural Information Processing*, volume 3316 of *Lecture Notes in Computer Science*, pages 800–805. Springer Berlin / Heidelberg, 2004.
[29] T. Smith and Waterman M. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
[30] Vladimir Vapnik. *Statistical Learning Theory*. Wiley-Interscience, ISBN 0-471-03003-1, 1989.
[31] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
[32] V. M. Velichko and N. G. Zagoruyko. Automatic recognition of 200 words. *International Journal of Man-Machine Studies*, 2:223–234, 1970.
[33] J. P. Vert, H. Saigo, and T. Akutsu. Local alignment kernels for biological sequences. In B. Scholkopf, K. Tsuda, and J.P. Vert, editors, *Kernel Methods in Computational Biology*, pages 131–154. MIT Press, 2004.
[34] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21:168–173, 1973.
[35] Adam Woznica, Alexandros Kalousis, and Melanie Hilario. Distances and (indefinite) kernels for sets of objects. *Data Mining, IEEE International Conference on*, 0:1151–1156, 2006.
[36] Gang Wu, Edward Y. Chang, and Zhihua Zhang. Learning with non-metric proximity matrices. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 411–414, New York, NY, USA, 2005. ACM.

TABLE 3
Comparative study using the UCR datasets: classification error rates (in %) obtained using the first near neighbor classification rule and a SVM classifier for the δ_{twed} and $RTWK_{twed}$ kernels. Two scores are given S1|S2: the first one, S1, is evaluated on the training data, while the second one, S2, is evaluated on the test data.

DATASET	**1-NN** δ_{twed}	**SVM** δ_{twed}	**SVM** $RTWK_{twed}$
Synthetic control	1|2.33	**0**|**1.33**	**0**|**1.33**
Gun-Point	**0**|1.33	**0**|0.67	**0**|**0**
CBF	0|0.67	3.33|3.67	**2.44**|**1.33**
Face (all)	1.43|18.93	**0.56**|16.86	0.72|**15.09**
OSU Leaf	17.59|24.79	**15**|**18.18**	19|22.73
Swedish Leaf	8.82|10.24	7.2|6.4	**6.6**|**5.12**
50 Words	18.26|18.9	15.66|**14.51**	**15.12**|16.26
Trace	1|5	**0**|**0**	**0**|1
Two Patterns	0|0.12	**0**|0.025	**0**|**0**
Wafer	.1|.86	**0.1**|0.41	**0.1**|**0.37**
face (four)	8.7|3.41	**8.33**|**2.27**	**8.33**|3.4
Ligthing2	13.56|21.31	15|**21.31**	**11.67**|**21.31**
Ligthing7	24.64|24.66	**25.29**|**23.29**	**25.29**|**23.29**
ECG	13.13|10	**12**|8	**12**|**7**
Adiac	36.25|37.6	30.51|31.02	**24.87**|**23.53**
Yoga	19.06|12.97	**12**|**9.9**	12.33|10.83
Fish	12.07|5.14	**4.57**|**2.86**	6.29|3.43
Coffee	18.52|21.43	25|28.5	**10.61**|**17.86**
OliveOil	11.11|16.67	**13.33**|**13.33**	**13.33**|**13.33**
Beef	58.62|53.3	**36.67**|53.33	46.67|**50**
# Best Scores	-	13|10	**15**|**14**
# Uniquely Best Scores	-	5|6	**6**|**10**

TABLE 4
Analysis of the deviation to conditionally definiteness for the gram-matrices associated to the δ_{dtw}, δ_{erp} and δ_{twed} distances. We report for each dataset the number of positive eigenvalues ($\#Pev$) relatively to the total number of eigenvalues ($\#Ev$)and the deviation to definiteness estimated as Δ_p that expresses in %. The expectation is a single positive eigenvalue, $\#Pev = 1$, corresponding to $\Delta_p = 0\%$.

DATASET	δ_{dtw}		δ_{erp}		δ_{twed}	
-	**#Pev/#Ev**	Δ_p	**#Pev/#Ev**	Δ_p	**#Pev/#Ev**	Δ_p
Synthetic control	110/300	15.66%	8/300	.16%	6/300	.22 %
Gun-Point	23/50	2.54%	1/50	0%	1/50	0%
CBF	5/30	3.36%	1/30	0%	1/30	0%
Face (all)	242/560	26.6%	83/560	2.42%	41/560	1.89%
OSU Leaf	96/200	31.79%	29/200	2.97%	16/200	.89%
Swedish Leaf	206/500	17.04%	24/500	.68%	23/500	.41%
50 Words	218/450	34.03%	119/450	9.54%	93/450	4.85%
Trace	43/100	5.42%	1/100	0%	1/100	0%
Two Patterns	453/1000	36.7%	259/1000	13.8%	226/1000	9.85%
Wafer	497/1000	14.84%	137/1000	1.29%	39/1000	.04%
face (four)	2/24	.74%	1/24	0%	1/24	0%
Ligthing2	20/60	13.44%	1/60	0%	1/60	0%
Ligthing7	24/70	14.25%	1/70	0%	1/70	0%
ECG	38/100	14.7%	1/100	0%	1/100	0%
Adiac	159/390	5.54%	26/390	.82%	39/390	.69%
Yoga	142/300	23.4%	29/300	3.17%	10/300	.41%
Fish	71/175	17.57%	1/175	0%	1/175	0%
Coffee	12/28	8.83%	1/28	0%	1/28	0%
OliveOil	4/30	.24%	1/30	0%	1/30	0%
Beef	15/30	6.17%	1/30	0%	1/30	0%