Constructing Positive Definite Elastic Kernels with Application to Time Series Classification

Pierre-François Marteau, Sylvie Gibet

Constructing Positive Definite Elastic 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 alignment (biological sequence alignment) based on the summing up of scores obtained by local alignments with gaps. The extensions we propose allow to construct, from classical time-warp distances, what we call summative time-warp kernels that are positive definite if some simple sufficient conditions are satisfied. Furthermore, from the same formalism, we derive a time-warp inner product that extends the usual euclidean inner product, providing the capability to handle discrete sequences or time series of variable lengths in an Hilbert space. The classification experiment we conducted, using either first near neighbor classifier or Support Vector Machine classifier leads to conclude that the positive definite elastic kernels we propose outperform the distance substituting kernels for some classical elastic distances we tested. In a similar way, for the considered task, the kernel based on the distance induced by the time-warp inner product significantly outperforms the kernel based on the Euclidean distance.

Index Terms—Elastic distance, Time warp kernel, Time warp inner product, Definiteness, Time series classification, SVM.

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 examine whether or not it is possible to construct Reproducing Time Warp Hilbert Spaces (RTWHS) from a given elastic measure, basically vector spaces characterized with inner products having time-warp capabilities. Another intriguing question is to determine whether it is possible or not to define an inner product structure from which a given elastic measure is induced? This question, apart from its theoretical implication, has a great impact when considering the potential application fields, since, if the answer is positive, it provides direct access to the Linear Algebra results and tools.

Unfortunately it seems that common elastic measures that are derived from DTW or Edit Distance 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.

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

This paper aims at exploring this issue and proposes Time Warp Kernels (TWK) constructions that try to preserve the properties of elastic measures from which they are derived, while offering the possibility possibility of embedding time series in Time Warped Hilbert Spaces. The main contributions of the paper are as follows

1) we establish the indefiniteness of the main time-warp measures used in the literature,
2) we propose some methods to construct positive definite kernels from classical time-warp measures,
3) we define simple Time Warp Inner Product (TWIP) as an extension to the Euclidean Inner Product, and
4) we experiment and compare the proposed kernels on time series classification tasks using a large variety of time series datasets.

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 TWK and TWIP 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 TWK and TWIP accuracies with classical elastic and non elastic measures. The seventh section proposes a conclusion and further research perspectives. Appendix A 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 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 [15] by Wagner and Fisher [31], the Smith and Waterman (SW) algorithm [27] 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 [19]. Similarly, dynamic time warping measures have been developed to evaluate similarity between numeric time series or timestamped data [30], [22], and more recently [6], [17] 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. [9][12][10]) 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) are reported to yield state-of-the art performances.

SVM or vast margin classifiers [28], [4], [24] are a set of supervised algorithms that learn how to solve discrimination or regression problems from positive and negative examples. They generalize linear classification algorithms by integrating two concepts: the maximal margin principle and a kernel function that defines the similarity or dissimilarity of any pair of examples, typically such as an inner product between the vector representation of two examples.

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 [32], [9]. 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 [33] 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 [10]. 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 work like [26], [14] 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 piecewise approximation of the time series are embedded into a proper so-called Reproducing Kernel Hilbert Space in which the SVM is learned.

Our approach is more direct, as it tries 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 [11] 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 [11], Saigo et al [21] define, from the Smith and Waterman algorithm [27], 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 nevertheless is nevertheless connected to the logarithm of a point-wise limit of a series of definite convolution kernels. 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. We generalize to some extent the results presented by Saigo et al. on the Smith and Waterman algorithm and propose extensions to construct time-warp inner products.

3 NOTATIONS AND MATHEMATICAL BACKGROUNDS

To ensure that this paper is relatively self-contained, we give in this section commonly used definitions, with few details, for metric, kernel and definiteness, sequence set, and classical elastic measures.

3.1 Metric

Definition 3.1: A metric, also called a distance, on a set U is a function $\delta : U \times U \rightarrow \mathbb{R}$ which satisfies the following axioms:
For all $(x, y) \in U \times U$,

1) $\delta(x, y) \geq 0$ (non negativity)
2) $\delta(x, y) = 0$ if and only if x=y. (null iff identical)
3) $\delta(x, y) = \delta(y, x)$ (symmetry)
4) $\delta(x, z) \leq \delta(x, y) + \delta(y, z)$. (subadditivity/triangle inequality)

3.2 Kernel and definiteness

A very large literature exists on kernels, among which [3], [24] and [25] present a large synthesis of major results. We give hereinafter some basic definitions.

Definition 3.2: A kernel on a non empty set U refers to a complex (or real) valued symmetric function $\varphi(x, y) : U \times U \rightarrow \mathbb{C}$ (or $\mathbb{R}$).

Definition 3.3: Let U be a non empty set. A function $\varphi : U \times U \rightarrow \mathbb{C}$ 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 3.4: Let U be a non empty set. A function $\varphi : U \times U \rightarrow \mathbb{C}$ 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 3.5: 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 PD and CPD for positive definite and conditionally positive definite to characterize either a kernel or a matrix having these properties.

Constructing PD kernels from CPD kernels is quite straightforward. For instance, if $-\varphi$ is a CPD 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 PD kernel, so are $e^{(\psi(x,y))}$ and $e^{-\varphi(x,y)}$. The converse is also true.
Furthermore, $e^{-t \cdot \varphi(x,y)}$ is PD for $t > 0$ if $-\varphi$ is CPD. We will precisely use this last results to construct PD kernels from classical elastic distances.

3.3 Sequence set

Definition 3.6: 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 3.7: 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 $A(i) = (a(i), t_{a(i)})$ where $a(i) \in S$ and $t_{a(i)} \in T$, with the condition that $t_{a(i)} > t_{a(j)}$ 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.

3.4 General Edit/Elastic distance on a sequence set

Definition 3.8: An edit operation is a pair $(a, b) \neq (\Lambda, \Lambda)$ of sequence elements, written $a \rightarrow b$. Sequence B results from the application of the edit operation $a \rightarrow b$ into sequence A, written $A \Rightarrow B$ via $a \rightarrow b$, if $A = \sigma a \tau$ and $B = \sigma b \tau$ for some subsequences σ and τ. We call $a \rightarrow b$ a match 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 $\Gamma_{A(i) \rightarrow B(j)}(A_1^p, B_1^q)$, or $\Gamma_{A(i) \rightarrow \Lambda, j}(A_1^p, B_1^q)$ or $\Gamma_{\Lambda, i \rightarrow B(j)}(A_1^p, B_1^q) \in \mathbb{R}$. To simplify this writing we will simply write $\Gamma(A(i) \rightarrow B(j))$, $\Gamma(A(i) \rightarrow \Lambda)$ or $\Gamma(\Lambda \rightarrow B(j))$ although this will not be fully appropriate in general.

Definition 3.9: A function $\delta : \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}$ 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

$$\delta(A_1^p, B_1^q) = \\ \text{Min} \begin{cases} \delta(A_1^{p-1}, B_1^q) + \Gamma(A(p) \rightarrow \Lambda) & delete \\ \delta(A_1^{p-1}, B_1^{q-1}) + \Gamma(A(p) \rightarrow B(q)) & match \\ \delta(A_1^p, B_1^{q-1}) + \Gamma(\Lambda \rightarrow B(q)) & insert \end{cases}$$

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

3.4.1 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. $\Gamma(A(p) \rightarrow \Lambda) = \Gamma(\Lambda \rightarrow B(q)) = 1$ while the *match* cost is null if $A(p) = B(q)$ or 1 otherwise.

3.4.2 Dynamic time warping

The DTW similarity measure δ_{dtw} [30][22] is defined according to the previous notations 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].

3.4.3 Edit Distance with real penalty

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

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.

3.4.4 Time warp edit distance

Time Warp Edit Distance (TWED) [16], [17] is defined similarly to the edit distance defined for string [15][31]. The similarity between any two time series A and B of finite length, respectively p and q is defined as:

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

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, $d(A(p), B(q)) = d_{LP}(a(p), b(q)) + \nu \cdot d_{LP}(t_{a(p)}, t_{t_b(q)})$ 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.

3.5 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 [21] and more recently for dynamic time warping kernels for time series alignment [8]. 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 Summative Time Warped Kernels

Definition 4.1: A function $< .;. >: \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}$ is called a Summative Time Warp Kernel (STWK) 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:

$$< A_1^p; B_1^q >= \sum \begin{cases} < A_1^{p-1}, B_1^q > \star f(\Gamma(A(p) \rightarrow \Lambda)) & delete \\ < A_1^{p-1}, B_1^{q-1} > \star f(\Gamma(A(p) \rightarrow B(q))) & match \\ < A_1^p, B_1^{q-1} > \star f(\Gamma(\Lambda \rightarrow B(q))) & insert \end{cases}$$

Where $\star$ is either the addition or the multiplication. This recursive definition requires to define an initialization. To that end we set $< \Omega, \Omega >= \xi$, where ξ is a real constant, and Ω the null sequence.

4.1.1 Interpretation of STWK

To interpret STWK we need first to introduce the concept of alignment path between two sequences or time series.

Definition 4.2: An (N, M)-warping path is a sequence $p = (p_1, \cdots, p_L)$ with $p_i = (n_i, m_i) \in \{1, \cdots, N\} \times \{1, \cdots, M\}$ for $i \in \{1, \cdots, L\}$ satisfying the following three conditions

i) Boundary condition: $p_1 = (1, 1)$ and $p_L = (N, M)$.
ii) Monotonicity condition: $n_1 \leq n_2, \cdots \leq n_L$ and $m_1 \leq m_2 \cdots \leq m_L$.
iii) Step size condition: $m_{i+1} > m_i$ or (inclusive) $n_{i+1} > n_i$ for $i \in \{1, \cdots, L-1\}$.

Summative refers to the $\sum$ operator replacing the $\min$ or $\max$ usually used. The recursion is initialized using $< A_1^0, B_1^0 >=< \Omega, \Omega >= \xi \in \mathbb{R}$.
This type of kernel sums up the multiplication or addition of the local quantities $f(\Gamma(A(i) \rightarrow B(j)))$ for all the possible alignment paths between the two time series.

Definition 4.3: If $\star$ is the addition, the STWK is called additive, otherwise it will be called multiplicative.

4.1.2 Definiteness of STWK

The following theorem, which generalizes the one presented in [8] establishes sufficient conditions on $f(\Gamma(a \rightarrow b))$ for an STWK to be definite and thus is a basis for the construction of definite STWK.

Theorem 4.4: Definiteness of STWK:

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

ii) An additive STWK is negative definite on $\mathbb{U}$ if the local kernel $k(x, y) = f(\Gamma(x \rightarrow y))$ is negative definite on $((S \times T) \cup \{\Lambda\})^2$ and $\xi \leq 0$.

iii) An additive STWK is conditionally positive definite if the local kernel $k(x, y) = f(\Gamma(x \rightarrow y))$ is conditionally positive definite on $((S \times T) \cup \{\Lambda\})^2$ and if $\xi \geq 0$.

(1)A sketch of proof for theorem 4.4 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 [23]. 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 [18] and thus forms a infinite dimension feature space.

4.1.3 Computational cost of STWK

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 $STWK(A, B)$ leads to a quadratic computational cost t$O(|A| \cdot |B|)$, 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 [22] 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.

4.2 Some instances of additive and multiplicative STWK

4.2.1 Multiplicative exponentiated STWK

Definition 4.5:

$$< A_1^p, B_1^q >_{me}= \tfrac{1}{3} \cdot \sum \begin{cases} < A_1^{p-1}, B_1^q >_{me} .e^{-\nu' \cdot \Gamma(A(p) \rightarrow \Lambda)} \\ < A_1^{p-1}, B_1^{q-1} >_{me} .e^{-\nu' \cdot \Gamma(A(p) \rightarrow B(q))} \\ < A_1^p, B_1^{q-1} >_{me} .e^{-\nu' \cdot \Gamma(\Lambda \rightarrow B(q))} \end{cases} \quad (2)$$

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, $\lim_{\nu' \rightarrow +\infty} -1/\nu' \cdot log(< A, B >_{me})$ does not coincide with the elastic distance δ that involves the same corresponding elementary costs.

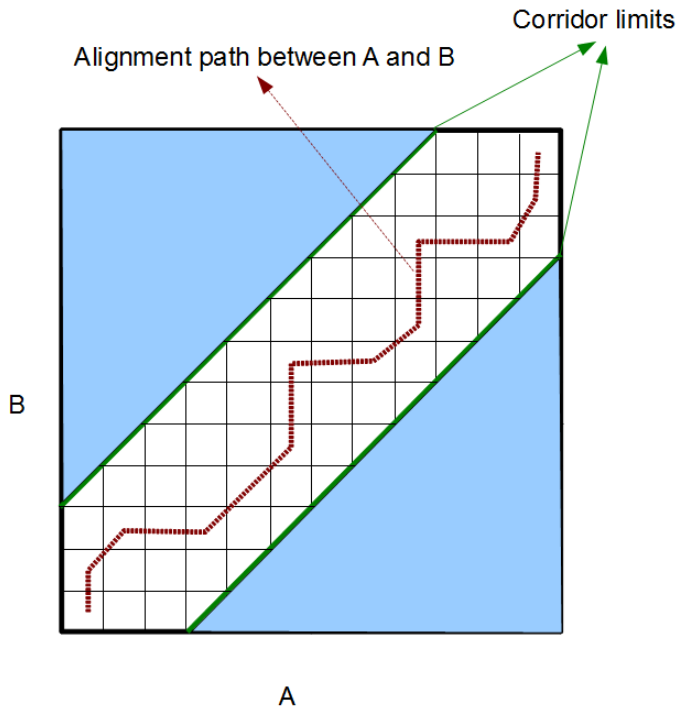

Fig. 1. Example of a symmetric corridor used to reduced the number of admissible alignment paths

We suggest setting $\xi = 1$ for this kind of multiplicative STWK.

Proposition 4.6: Definiteness of the multiplicative exponentiated STWK $< ., . >_{me}$:
$< ., . >_{me}$ 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 multiplicative SWTKs constructed from these distances are referred respectively to $STWK_{lev}$, $STWK_{erp}$, $STWK_{dtw}$, $STWK_{twed}$ in the rest of the paper.

The proof of proposition 4.6 is straightforward and is omitted.

4.2.2 Interpretation of multiplicative STWK

For multiplicative STWK each alignment path is assigned with a cost that is the multiplication of the local costs attached to each edge of the path. For multiplicative exponentiated STWK, the local cost, e.g. $e^{-\nu' \cdot \Gamma(A(p) \rightarrow B(q))}$ can be interpreted, up to a normalizing constant, as a probability to align symbol $A(p)$ with symbol $B(q)$, and the cost affected to each path can be interpreted as the probability of a specific alignment between two sequences. In that case the STWK, 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 multiplicative STWK 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 [21], the main drawbacks of these kind of kernels is the vanishing of the product of local costs (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.

4.2.3 Additive STWK

Although a very large family of distinct additive STWK exists, we present below two simple instances of additive STWK that correspond to generalizations of the Euclidean inner product.

Definition 4.7:

$$< A_1^p, B_1^q >_{twip_1} = \frac{1}{3} \cdot \sum \begin{cases} < A_1^{p-1}, B_1^q >_{twip_1} \\ < A_1^{p-1}, B_1^{q-1} >_{twip_1} + e^{-\nu . d(t_{a(p)}, t_{b(q)})} (a(p) \cdot b(q)) \\ < A_1^p, B_1^{q-1} >_{twip_1} \end{cases} \quad (3)$$

where d is a distance, and ν a stiffness parameter.

Definition 4.8:

$$< A_1^p, B_1^q >_{twip_2} = \frac{1}{1+2 \cdot e^{-\nu}} \cdot \sum \begin{cases} e^{-\nu} \cdot < A_1^{p-1}, B_1^q >_{twip_2} \\ < A_1^{p-1}, B_1^{q-1} >_{twip_2} + e^{-\nu . d(t_{a(p)}, t_{b(q)})} (a(p) \cdot b(q)) \\ e^{-\nu} \cdot < A_1^p, B_1^{q-1} >_{twip_2} \end{cases} \quad (4)$$

where d is a distance, and ν a stiffness parameter.

We propose taking $\xi = 0$ for these two additive SWTK.

4.2.4 Interpretation of additive STWK

For additive STWK each alignment path is assigned with a cost that is the addition of the local costs attached to each edge of the path. We cannot maintain a probabilistic interpretation for local costs of the form $e^{-\nu \cdot \Gamma(A(p) \rightarrow B(q))}$. Nevertheless the additive STWK does not suffer from the diagonal dominance problem mentioned above and can be easily normalized. Furthermore, additive STWK can be viewed as a generalization of the standard inner product. In particular, note that for $\nu \rightarrow \infty$, $twip_2$, when applied to a pair of time series of equal lengths and identically sampled, identifies with the inner product in Euclidean spaces.

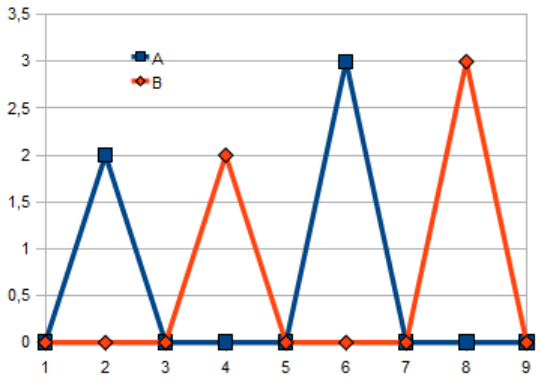

Fig. 2. When considering the discrete time series $A = (0,1)(2,2)(0,3)(0,4)(0,5)(3,6)(0,7)(0,8)(0,9)$ and $B = (0,1)(0,2)(0,3)(2,4)(0,5)(0,6)(0,7)(3,8)(0,9)$, the Euclidean inner product is null, while, for $\nu = .1$, the $twip_1$ inner product of A and B equals to $.459$, and the $twip_2$ inner product of A and B equals to $.475$. For $\nu = 100$, both TWIP give a null value.

Let $\mathbb{U}^N$ be the set of all time series sequences whose lengths is N and whose elements are selected in $S \times \{t_1 < t_2 < \cdots < t_N\}$, and consider the additive operator $\oplus$ and $\otimes$ defined below:

Definition 4.9: For all $A \in \mathbb{U}^N$ and all $\lambda \in \mathbb{R}$, $C = \lambda \otimes A \in \mathbb{U}^N$ is such that for all $0 \leq i \leq N$, $C(i) = (\lambda.a(i), t_i)$ and thus $|C| = |A| = N$.

Definition 4.10: For all $(A, B) \in (\mathbb{U}^N)^2$, $C = A \oplus B \in \mathbb{U}^N$ is defined such that for all $0 \leq i \leq N$, $C(i) = (a(i) + b(i), t_i)$ and thus $|C| = |A| = |B| = N$.

Proposition 4.11: Definiteness and inner product structure of the additive STWK:

i) $\forall N \in \mathbb{N}^+$, $< .,. >_{twip1}$ and $< .,. >_{twip2}$ are positive definite on $\mathbb{U}^N$.

ii) Furthermore, $< .,. >_{twip1}$ and $< .,. >_{twip2}$ are inner products on $(\mathbb{U}^N, \oplus, \otimes)$, that we call Time Warp Inner Products (TWIP), where $\oplus$ and $\otimes$ are defined in definition 4.10 and definition 4.9 respectively,

iii) The Euclidean inner product $< .,. >_E$ on a set of time series of constant lengths and uniformly sampled is the limit when $\nu \to \infty$ of $< .,. >_{twip2}$ on this same set.

Note that any discrete time series space $\mathbb{U}^N$ (a set of time series of length N that are uniformly sampled), when provided with the $\oplus$ and $\otimes$ operators and the metric (norm) induced by a TWIP, is a Hilbert space.
The proof of proposition 4.11 is straightforward and is omitted.

5 Classification Experiments

We empirically evaluate the effectiveness of some STWK comparatively to Gaussian Radial Basis Function (RBF) Kernels or elastic distance substituting kernels [9] 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 [13]. 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 optimize the meta parameters of the considered comparability measure.
- the second one is a SVM classifier [4], [29] 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.

TABLE 1
Dataset sizes and meta parameters used in conjunction with K_{ed} and $STWK_{twip_2}$ kernels

DATASET	length	#class	#train	#test	$K_{ed} : C, \sigma$	$STWK_{twip_2} : \nu, C, \sigma$
Synthetic control	60	6	300	300	1.0;0.125	0.1;2.0;0.0625
Gun-Point	150	2	50	150	256;1.0	1e-5;2048;2.0
CBF	128	3	30	900	8;1.0	0.01;1.0;0.0312
Face (all)	131	14	560	1690	4;0.5	1e-5;16.0;0.062
OSU Leaf	427	6	200	242	2;0.125	0.01;16;0.25
Swedish Leaf	128	15	500	625	128;0.125	1.0;16;0.125
50 Words	270	50	450	455	32;0.5	0.01;32;0.25
Trace	275	4	100	100	8;0.0156	1e-5;512;0.25
Two Patterns	128	4	1000	4000	4.0;0.25	0.01;2.0;0.0625
Wafer	152	2	1000	6174	4.0;0.5	0.1;8.0;0.0625
face (four)	350	4	24	88	8.0;2.0	1000;8.0;2.0
Ligthing2	637	2	60	61	2.0;0.125	.01;1.0;0.125
Ligthing7	319	7	70	73	32.0;256.0	0.1;512.0;8.0
ECG	96	2	100	100	8.0;0.25	1000.0, 8.0, 0.25
Adiac	176	37	390	391	1024.0;0.125	1000;1024.0;0.125
Yoga	426	2	300	3000	64.0;0.125	1.0;16.0;0.0625
Fish	463	7	175	175	64.0;1.0	10.0;64.0;1.0
Coffee	286	2	28	28	128.0;4.0	1000.0;128.0;4.0
OliveOil	570	4	30	30	2.0;0.125	0.01;4.0;0.125
Beef	470	5	30	30	128.0;4.0	1000.0;128.0;4.0

TABLE 2
Meta parameters used in conjunction with δ_{erp}, $STWK_{erp}$, δ_{dtw}, $STWK_{dtw}$, δ_{twed} and $STWK_{twed}$ kernels

DATASET	$\delta_{erp} : g; C; \sigma$	$STWK_{erp} : g; \nu'; C; \sigma$	$\delta_{dtw} : C; \sigma$	$STWK_{dtw} : \nu'; C; \sigma$	$\delta_{twed} : \lambda; \nu; C; \sigma$	$STWK_{twed} : \lambda; \nu; \nu'; C; \sigma$
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 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 K_{ed} and $STWK_{twip_2}$ 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 ED	1-NN $STWK_{twip_2}$	SVM K_{ed}	SVM $STWK_{twip_2}$
Synthetic control	8.7|12	**.33**|**.67**	2.33|2	**1.67**|**.33**
Gun-Point	**4.08**|**8.67**	**4.08**|**8.67**	6|6	**2**|**3.33**
CBF	17.24|14.8	**6.9**|**2.33**	3.3|10.89	**3.33**|**3.44**
Face (all)	11.27|28.64	**6.98**|**24.37**	6.07|**16.63**	**4.29**|23.79
OSU Leaf	38.19|48.34	**33.17**|**45.04**	32|**43.80**	**29.5**|44.21
Swedish Leaf	25.05|21.12	**23.25**|**20.19**	23.25|20.19	**12.8**|**8.8**
50 Words	36.52|36.92	**33.18**|**34.28**	32.45|30.10	**31.11**|**29.67**
Trace	16.16|24	**11.11**|**23**	7|24	**3**|**10**
Two Patterns	8.61|9.32	**4.8** |**4.17**	8.6|7.45	**5.5**|**4.1**
Wafer	0.7|**0.45**	**.4**|.67	.8|**.52**	**.2**|0.82
face (four)	**34.78**|**21.59**	**34.78**|**21.59**	25|**14.77**	**20.83**|22.73
Ligthing2	28.81|24.59	**22.03**|**16.39**	21.77|32.79	**20**|**21.31**
Ligthing7	36.23|42.47	**28.98**|**31.51**	37.14|**36.98**	**30**|**36.98**
ECG	**14.14**|**12**	**14.14**|**12**	**8**|**9**	**8**|**9**
Adiac	**39.59**|**38.87**	**39.59**|**38.87**	**25.13**|25.83	**25.13**|**24.29**
Yoga	23.08|16.97	**20**|**16.83**	16|14.87	**15.33**|**14.7**
Fish	**24.14**|**21.71**	**24.14**|**21.71**	**13.14**|13.14	**13.14**|**12.57**
Coffee	22.22|25	**18.52**|**17.85**	**0**|**0**	**0**|3.57
OliveOil	13.79|**13.33**	**13.33**|**13.33**	10|**13.33**	**6.37**|**13.33**
Beef	51.72|**46.67**	**48.28**|50	**30**|**30**	**30**|**30**
# Best Scores	4|8	**20**|**18**	4|9	**20**|**15**
# Uniquely Best Scores	0|2	**15**|**12**	0|5	**15**|**11**

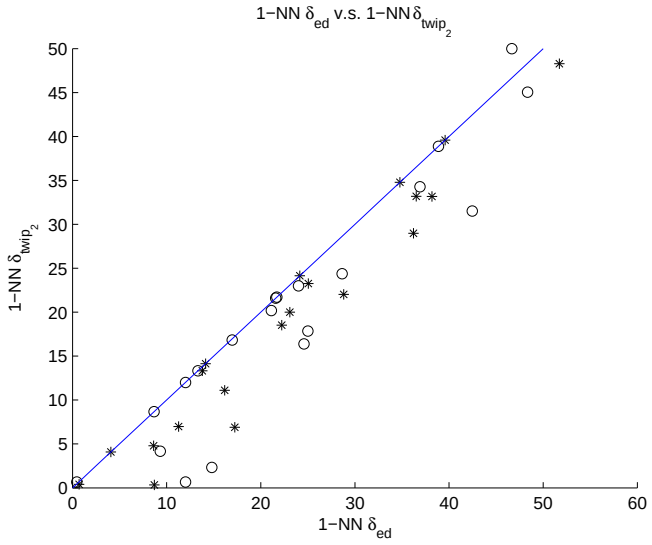

Fig. 3. Comparison of error rates (in %) between two 1-NN classifiers based on the Euclidean Distance (1-NN ED), δ_{ED}, and the distance δ_{twip_2} induced by a time-warp inner product (1-NN TWIP). The straight line has a slope of 1.0 and dots correspond, for the pair of classifiers, to the error rates on the train (star) or test (circle) data sets. A dot below (resp. above) the straight line indicates that distance δ_{twip_2} has a lower (resp. higher) error rate than distance δ_{ED}

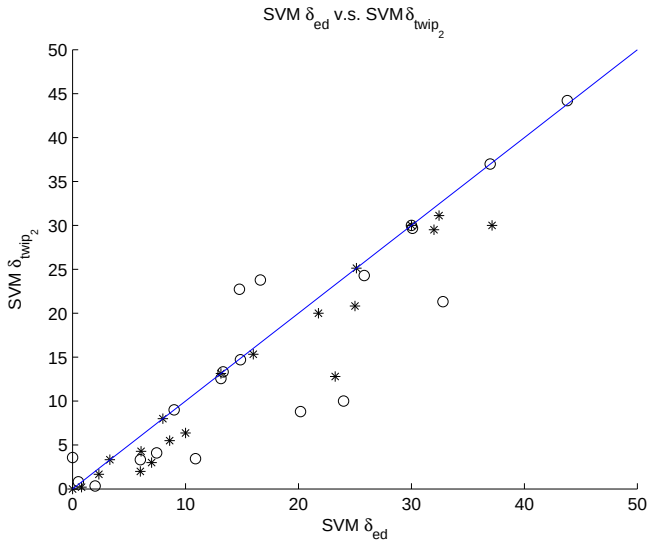

Fig. 4. Comparison of error rates (in %) between two SVM classifiers, the first one based on the Euclidean Distance Gaussian kernel (SVM K_{ed}), and the second one based on a Gaussian kernel induced by a time-warp inner product (SVM $STWK_{twip_2}$). The straight line has a slope of 1.0 and dots correspond, 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 $STWK_{twip_2}$ has a lower (resp. higher) error rate than distance SVM K_{ed}

TABLE 4
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, $STWK_{erp}$, dtw and $STWK_{dtw}$ 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** $STWK_{erp}$	**1-NN** δ_{dtw}	**SVM** δ_{dtw}	**SVM** $STWK_{dtw}$
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**

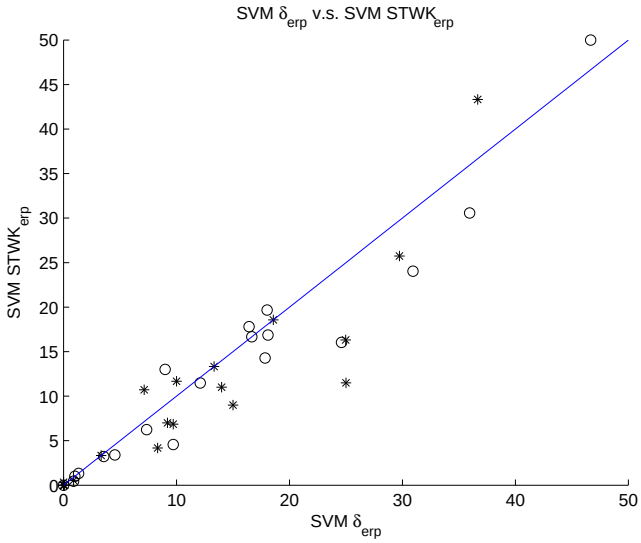

Fig. 5. Comparison of error rates (in %) between two SVM classifiers, the first one based on the δ_{erp} substituting kernel (SVM δ_{erp}), and the second one based on an additive time-warp kernel induced by the ERP distance (SVM $STWK_{erp}$). The straight line has a slope of 1.0 and dots correspond, 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 $STWK_{erp}$ has a lower (resp. higher) error rate than distance SVM δ_{erp}

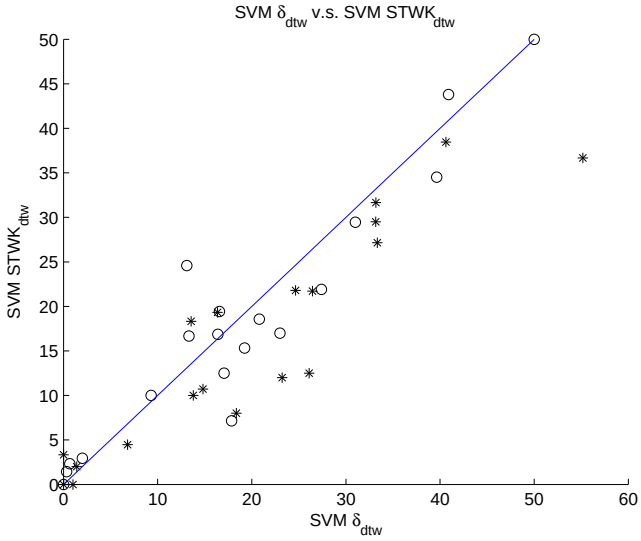

Fig. 6. Comparison of error rates (in %) between two SVM classifiers, the first one based on the δ_{dtw} substituting kernel (SVM δ_{dtw}), and the second one based on an additive time-warp kernel induced by the δ_{dtw} distance (SVM $STWK_{dtw}$)). The straight line has a slope of 1.0 and dots correspond, 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 $STWK_{dtw}$ has a lower (resp. higher) error rate than distance δ_{dtw}

TABLE 5
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 $STWK_{twed}$ 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 $STWK_{twed}$
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**

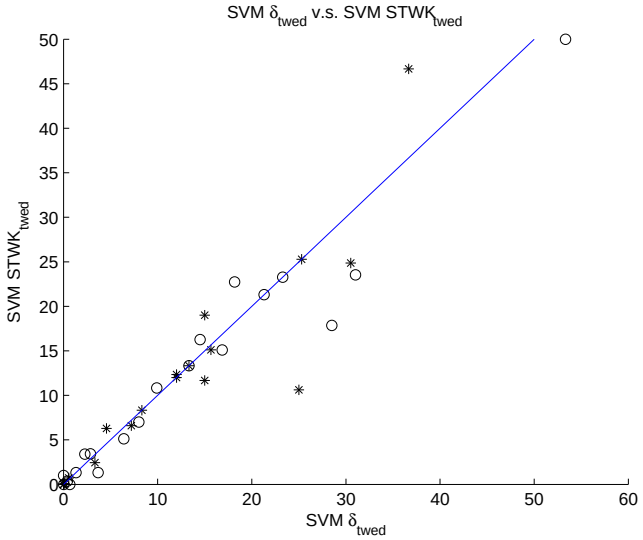

Fig. 7. Comparison of error rates (in %) between two SVM classifiers, the first one based on the δ_{twed} substituting kernel (SVM δ_{twed}), and the second one based on an additive time-warp kernel induced by the ERP distance (SVM $STWK_{twed}$). The straight line has a slope of 1.0 and dots correspond, 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 $STWK_{twed}$ has a lower (resp. higher) error rate than distance SVM δ_{twed}

5.1 Additive STWK

We tested the additive STWK based on the Time Warp Inner Product $< A, B >_{twip_2}$ (Eq.4) (we choose to test $twip_2$ because for large ν it tends towards the Euclidean inner product). Precisely, we used the time-warp distance induced by $< A, B >_{twip_2}$, basically $\delta_{twip_2}(A, B) = (< A - B, A - B >_{twip_2})^{1/2}$.

5.1.1 Meta parameters

δ_{twip_2} is characterized by the meta parameter ν (the *stiffness* parameter) that is optimized for each dataset on the train data by minimizing the classification error rate of a first near neighbor classifier. For this kernel, ν is selected in $\{100, 10, 1, .1, .01, ..., 1e-5\}$.

To explore the potential benefits of TWIP against the Euclidean inner product, we also tested the Euclidean Distance δ_{ed} which is the limit when $\nu \to \infty$ of δ_{twip_2}.

The kernels exploited by the SVM classifiers are the Gaussian kernels $STWK_{twip_2}(A, B) = e^{\delta_{twip_2}(A,B)^2/(2\cdot\sigma^2)}$ and $K_{ed}(A, B) = e^{\delta_{ed}(A,B)^2/(2\cdot\sigma^2)}$. The meta parameters C is selected into the discrete set $\{2^{-5}, 2^{-4}, ..., 1, 2, ..., 2^{10}\}$, and σ^2 into $\{2^{-5}, 2^{-4}, ..., 1, 2, ..., 2^{10}\}$.

Table 1 gives for each data set and each tested kernels (K_{ed} and $STWK_{twip_2}$) the corresponding optimized values of the meta parameters.

5.2 Multiplicative STWK

We tested the multiplicative exponentiated STWK based on the $\delta_{erp}, \delta_{dtw}, \delta_{twed}$ distance costs. We consider respectively the positive definite $STWK_{erp}, STWK_{dtw}, STWK_{twed}$ 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 STWK, e.g. SVM δ_{erp}, SVM $STWK_{erp}$, SVM δ_{dtw}, SVM $STWK_{dtw}$, SVM δ_{twed}, SVM $STWK_{twed}$.

For δ_{erp}, δ_{twed}, $STWK_{erp}$ and $STWKtwed$ we used the L1-norm, while the L2-norm has been implemented for δ_{dtw} and $STWK_{dtw}$, a classical choice for DTW [20].

5.2.1 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 $STWK_{me}(ERP)$ 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 $STWK_{twed}$ kernel.

The kernels exploited by the SVM classifiers are the Gaussian Radial Basis Function (RBF) kernels $K(A, B) = e^{\delta(A,B)^2/(2\cdot\sigma^2)}$ where δ stands for $\delta_{erp}, \delta_{dtw}, \delta_{twed}$, $STWK_{erp}$, $STWK_{dtw}$, $STWK_{twed}$. 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 $STWK_{erp}$, $STWK_{dtw}$ and $STWK_{twed}$ 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 $STWK_*$ 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 2 gives for each data set and each tested kernel (δ_{erp}, δ_{dtw}, δ_{twed}, $STWK_{erp}$, $STWK_{dtw}$ and $STWK_{twed}$) the corresponding optimized values of the meta parameters.

5.3 Discussion

5.3.1 Additive STWK experiment analysis

Table 3 shows the classification error rates obtained for the tested methods, e.g. the first near neighbor classifier based on the Euclidean Distance and the distance induced by the time-warp inner product (1-NN ED and 1-NN δ_{twip_2}), the Gaussian RBF kernel SVM based on the Euclidean distance and the distance induced by the time-warp inner product (SVM K_{ed} and SVM $STWK_{twip_2}$).

This experiment shows that the time-warp inner product is significantly more effective for the considered tasks comparatively to the Euclidean distance, since it exhibits, on average, the lowest error rates for most of the tested datasets for both the 1-NN and SVM classifiers, as shown in Table 3 and Figures 3 and 4. The

TABLE 6
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%

stiffness parameter in δ_{twip_2} seems to play a significant role in these classification tasks, and this for a quite large majority of data sets.

Only one dataset, *Beef*, is better classified by the 1-NN ED classifier on the test data, although the error rate on the train data is lower for the 1-NN δ_{twip_2} classifier. As the train data for the beef dataset is quite small (30 instances), the significance of this specific result is not clear. For the SVM classifiers, only two datasets, *Face (all)* and *Face (four)*, are significantly better classified on the test data by SVM K_{ed} classifiers. Nevertheless, for these two datasets, $STWK_{twip_2}$ reaches a better score on the train data. We are facing here the trade-off between learning and generalization capabilities. The meta parameter ν is selected such as to minimized the classification error on the train data. If this strategy is on average a winning strategy, some datasets show that it does not necessarily lead to a good trade-off, this is the case for *Face (all)* and *Face (four)* datasets.

5.3.2 Multiplicative STWK experiment analysis

Tables 4 and 5 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 STWK kernels (SVM $STWK_{erp}$, SVM $STWK_{dtw}$ and SVM $STWK_{twed}$).

In this experiment, we show that the SVM classifiers clearly outperform the 1-NN classifiers. But the interesting results reported in tables 4 and 5 and figures 5, 6 and 7 is that SVM $STWK_{erp}$ and SVM $STWK_{twed}$ perform slightly better than SVM δ_{erp} and SVM δ_{twed} respectively, and the SVM $STWK_{dtw}$ 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 $STWK_{dtw}$ kernel based on δ_{dtw} seems to correct greatly its drawbacks. To explore further the potential impact of indefiniteness on classification error rates, we give in Table 6 two quantified hints of deviation to conditionally definiteness for the gram-matrices corresponding to the δ_{dtw}, δ_{erp} and δ_{twed} distances. Since a conditionally (negative) definite gram-matrix is characterized by 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 when δ_{dtw} does not. The regularization brought by STWK is therefore more effective on δ_{dtw}. This is the case for instance on the *Beef* dataset for which, on the train data, δ_{twed} performs slightly better than the $STWK_{twed}$. In this case, both kernels lead to a definite gram-matrix, and the extra parameter ν' in use in the $STWK_{twed}$ kernel explains probably a poorer classification rate due to a lack of learning data. Nevertheless, similarly to the additive STWK, few datasets are better classified by SVM that use directly the distance kernel instead of the derived STWK kernel. The same reasons mentioned above in the case of additive STWK can be invoked here also. 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 [11] and local alignment kernels defined for string processing around the Smith and Waterman algorithm [27] [21], we propose summative time-warp kernels (STWK) applicable for string and time series processing. We give some simple sufficient conditions to build positive definite STWK. Our generalization leads us to propose additive and multiplicative STWK. For multiplicative STWK, we show that, for the exponentiated version we have tested, the sufficient 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. From the general additive STWK definition we have suggested a time-warp inner product (TWIP) from which a metric (or norm) that generalizes the Euclidean distance (or Euclidean norm) is induced. The experiments conducted on a variety of time series datasets show that the multiplicative positive definite STWKs outperforms the indefinite elastic distances they are derived from when considering 1-NN and SVM classification tasks, specifically when the gram-matrix associated to the elastic distance is far from definiteness.

Our experiments also show that the additive STWK we constructed from the proposed instance of TWIP ($twip_2$)significantly outperforms the kernels derived from the Euclidean inner product. This time-warp inner product opens some interesting perspectives since it leads to reconsidering the notion of orthogonality in discrete time series spaces. In particular, in such spaces provided with a TWIP the discrete sine and cosine waveforms are no longer orthogonal. Is it therefore relevant to raise the issue of a discrete elastic Fourier transform?

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

$$M_{lev}^V = \begin{pmatrix} 0 & 3 & 2 & 1 & 2 \\ 3 & 0 & 2 & 2 & 1 \\ 2 & 2 & 0 & 3 & 3 \\ 1 & 2 & 3 & 0 & 3 \\ 2 & 1 & 3 & 3 & 0 \end{pmatrix} \quad (5)$$

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 $C \cdot M_{lev}^V \cdot C^T = 2/3 > 0$ and $D \cdot M_{lev}^V \cdot D^T = -4/3 < 0$, 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 $V = \{01, 012, 0123, 01234\}$.

Then the DTW empiric gram matrix evaluated on V is

$$M_{dtw}^V = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \end{pmatrix} \quad (6)$$

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 $C \cdot M_{dtw}^V \cdot C^T = 2/32 > 0$ and $D \cdot M_{dtw}^V \cdot D^T = -1/2 < 0$, 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:

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

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:

$$M^V_{erp} = \begin{pmatrix} 0 & 2 & 3 & 3 & 4 & 5 & 4 & 2 & 4 & 5 \\ 2 & 0 & 3 & 5 & 2 & 3 & 2 & 2 & 4 & 5 \\ 3 & 3 & 0 & 4 & 3 & 2 & 3 & 1 & 3 & 4 \\ 3 & 5 & 4 & 0 & 7 & 6 & 7 & 5 & 1 & 2 \\ 4 & 2 & 3 & 7 & 0 & 3 & 2 & 2 & 6 & 5 \\ 5 & 3 & 2 & 6 & 3 & 0 & 1 & 3 & 5 & 4 \\ 4 & 2 & 3 & 7 & 2 & 1 & 0 & 2 & 6 & 5 \\ 2 & 2 & 1 & 5 & 2 & 3 & 2 & 0 & 4 & 5 \\ 4 & 4 & 3 & 1 & 6 & 5 & 6 & 4 & 0 & 1 \\ 5 & 5 & 4 & 2 & 5 & 4 & 5 & 5 & 1 & 0 \end{pmatrix} \quad (8)$$

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
PROOFS OF MAIN PROPOSITIONS

B.1 Proof of theorem 4.4

i) Let us show that if the function:
$f(x,y) = f(\Gamma(x \to y)) : ((S \times T) \cup \{\Lambda\})^2 \to \mathbb{R}$ is positive definite and if $\xi > 0$, then an additive or multiplicative STWK is definite positive.

Let us first denote $< A, B >_{i,j} = < A^i_1, B^j_1 >$ the restriction of the STWK up to index i and j such that $0 \leq i \leq |A|$ and $0 \leq j \leq |B|$.

Furthermore let $\kappa_{A(i)\to B(j)}(A,B)$, $\kappa_{\Lambda\to B(j)}(A,B)$ and $\kappa_{A(i)\to\Lambda}(A,B)$ be local kernels defined on $\mathbb{U}^2$ as follows:

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

If $f(x,y) = f(\Gamma(x \to y)) : ((S \times T) \cup \{\Lambda\})^2 \to \mathbb{R}$ is PD on $(S \times T) \cup \{\Lambda\}$, we directly establish that the local kernels $\kappa_{A(i)\to B(j)}(A,B)$, $\kappa_{\Lambda\to B(j)}(A,B)$ and $\kappa_{A(i)\to\Lambda}(A,B)$ are PD kernels on $\mathbb{U} \times \mathbb{U}$.

Let us define an alignment path between any pair (A,B) of sequences in $\mathbb{U}$: the alignment of an element $A(i_k)$ of A with an element $B(j_k)$ of B is defined by a couple $p_k = (i_k, j_k)$, with $0 \leq i_k \leq |A|$ and $0 \leq j_k \leq |B|$. An alignment path for a pair of sequences (A,B) is defined by a sequence $\pi = \pi(1), \pi(2), \cdots, \pi(k), \cdots, \pi(K)$ such that the sequences $i_k, k = 1, \cdots, K$ and $j_k, k = 1, \cdots, K$ are non decreasing $(i_{k-1} \leq i_k, j_{k-1} \leq j_k)$ and verifies either $(i_{k-1} < i_k)$ or $(j_{k-1} < j_k)$ for all k.

Each alignment path $\pi = \pi(1), \pi(2), \cdots, \pi(K)$ uniquely characterizes a sequence of elementary editing operations $\gamma = \gamma(1), \gamma(2), \cdots, \gamma(K)$, where each $\gamma(k)$ belongs to either a match $(A(i_k) \to B(j_k))$, an insertion $(\Lambda \to B(j_k))$ or a deletion $(A(i_k) \to \Lambda)$.

And finally let $\mathcal{E}(A,B)$ denotes the set of all existing editing sequences between sequences A and B. Note that $\mathcal{E}(A,B)$ is finite iff A and B are finite sequences.

Proposition B.1: For any pair $(A,B) \in \mathbb{U}^2$, any (i,j) such that $0 \leq i \leq |A|$ and $0 \leq j \leq |B|$ we have

$$< A^i_1, B^j_1 > = \sum_{\gamma \in \mathcal{E}(A^i_1, B^j_1)} \xi \star \prod_{k=1\cdots|\gamma|}^{\star} \kappa_{\gamma(k)}(A^i_1, B^j_1) \quad (9)$$

where

$$\prod^{\star}$$

is either the product operator if $\star$ is the multiplication, or the sum operator if $\star$ is the addition.

We prove proposition B.1 by induction on $r = i + j$. The proposition is true for $r = 0$, since then $i = j = 0$ and then $< A^0_1, B^0_1 > = < \Lambda, \Lambda > = \xi$. The proposition is true for $r = 1$, since

- $< A^0_1, B^1_1 > = < A^0_1, B^0_1 > \star f(\Gamma(\Lambda \to B(1)))$
  $= \xi \star \kappa_{\Lambda\to B(1)}(A^0_1, B^1_1)$, and

- $< A_1^1, B_1^0 >=< A_1^0, B_1^0 > \star f(\Gamma(A(1) \rightarrow \Lambda))$
  $= \xi \star \kappa_{A(1)\rightarrow\Lambda}(A_1^1, B_1^0)$.

Let suppose proposition B.1 is true for all $n \leq r$ 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 $< ., . >$ we have

$$\begin{aligned} < A_1^i, B_1^j >=& < A_1^{i-1}, B_1^j > \star f(\Gamma(A(i) \rightarrow \Lambda)) \\ &+ < A_1^{i-1}, B_1^{j-1} > \star f(\Gamma(A(i) \rightarrow B(j))) \\ &+ < A_1^i, B_1^{j-1} > \star f(\Gamma(\Lambda \rightarrow B(j))). \end{aligned} \quad (10)$$

which rewrites

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

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

$$\begin{aligned} < A_1^{i-1}, B_1^j >=& \sum_{\gamma \in \mathcal{E}(A_1^{i-1}, B_1^j)} \xi \star \prod_{k=1\cdots|\gamma|}^{\star} \kappa_{\gamma(k)}(A_1^i, B_1^j) \\ < A_1^{i-1}, B_1^{j-1} >=& \sum_{\gamma \in \mathcal{E}(A_1^{i-1}, B_1^{j-1})} \xi \star \prod_{k=1\cdots|\gamma|}^{\star} \kappa_{\gamma(k)}(A_1^i, B_1^j) \quad (12) \\ < A_1^i, B_1^{j-1} >=& \sum_{\gamma \in \mathcal{E}(A_1^i, B_1^{j-1})} \xi \star \prod_{k=1\cdots|\gamma|}^{\star} \kappa_{\gamma(k)}(A_1^i, B_1^j) \end{aligned}$$

Recombining equations 11 and 12 and completing the editing sequences we get the expected decomposition for $< A_1^i, B_1^j >$.

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

$$\begin{aligned} < A_1^i, B_1^j >=& \xi \star f(\Gamma(A(1) \rightarrow \Lambda)) \cdots \star f(\Gamma(A(r) \rightarrow \Lambda)) \\ =& \xi \star \prod_{k=1\cdots|\gamma|}^{\star} \kappa_{A(k)\rightarrow\Lambda}(< A_1^i, B_1^j >) \end{aligned}$$

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

Proof of theorem:
For all finite subset $V = \{A_1, A_2, \cdots, A_{|V|}\}$ of $\mathbb{U}$ and all $(\alpha_1, \alpha_2, \cdots, \alpha_{|V|}) \in \mathbb{R}^{|V|}$, according to proposition B.1 we have

$$\begin{aligned} &\sum_{m,n} \alpha_m \alpha_n < A_m, A_n >= \\ &\sum_{m,n} \alpha_m \alpha_n \sum_{\gamma \in \mathcal{E}(A_{m,1}^{|A_m|}, A_{n,1}^{|A_n|})} \xi \prod_{k=1\cdots|\gamma|}^{\star} \kappa_{\gamma(k)}(A_{m,1}^{|A_m|}, A_{n,1}^{|A_n|}) \end{aligned}$$

Since $\xi \geq 0$, the decomposition of the STWK kernel results in a sum of product (if $\star$ is the multiplication, with $\xi > 0$) or a sum of sum (if $\star$ is the addition, with $\xi = 0$) of local kernels applying on the same arguments (proposition B.1) that are all positive definite by asumption, and since positive definite kernels are closed under summation or multiplication [3], the STWK is proved to be definite positive.

As the sum of CPD or ND kernels are also CPD or ND, similarly to i), proposition B.1 directly leads to a proof of ii) and iii). □

B.2 Proof of proposition 4.11

The proofs of i) and ii) are obtained using a similar recursion as the one used to prove theorem 4.4. iii) is immediate.

ACKNOWLEDGMENTS

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] 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.

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

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

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

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

[13] 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/.
[14] 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.
[15] 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).
[16] 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/.
[17] 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.
[18] C. A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive functions. *Constructive Approximation*, 2:11–22, 1986.
[19] W. Pearson. Rapid and sensitive sequence comparisons with fasp and fasta. *Methods Enzymol*, 183:63–98, 1990.
[20] 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.
[21] H. Saigo, J.P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20:1682–1689, 2004.
[22] 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.
[23] I. J. Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536, nov 1938.
[24] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
[25] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.
[26] 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.
[27] T. Smith and Waterman M. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
[28] Vladimir Vapnik. *Statistical Learning Theory*. Wiley-Interscience, ISBN 0-471-03003-1, 1989.
[29] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
[30] V. M. Velichko and N. G. Zagoruyko. Automatic recognition of 200 words. *International Journal of Man-Machine Studies*, 2:223–234, 1970.
[31] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21:168–173, 1973.
[32] 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.
[33] 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.