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 that we propose allow to construct, from classical recursive definition of elastic 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 weaker than those proposed in earlier works, although a recursive regularizing term is required to get the proof of the positive definiteness as a direct consequence of the Haussler's convolution theorem. Furthermore, the positive definiteness is maintained when a symmetric corridor is used to reduce the search space and thus the algorithmic complexity, which, is quadratic in the worse case . The classification experiment we conducted on three classical time warp distances (two of which being metrics), using Support Vector Machine classifier, leads to the conclusion that, when the pairwise distance matrix obtained from the training data is *far* from definiteness, the positive definite recursive elastic kernels outperform in general the distance substituting kernels for several classical elastic distances we have tested.

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

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 in different domains, such as physics, chemistry, finance, bio-informatics, network monitoring, etc, have demonstrated the benefits of using elastic measures. A natural question to ask at this point is whether we can develop and use *elastic* kernel methods based from such elastic distances. This requires to address the existence and construction of Reproducing Edit Distance Hilbert Spaces for a given elastic measure, i.e. a functional vector space endowed with the dot product in which edit distances or time warping algorithms can be defined. 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 nevertheless, we can propose (close) regularized variants for which such kernel construction construction is possible.

This paper aims at (re-)exploring this issue and, following earlier works ([1], [2], [3], [4], [5]) proposes Recursive Edit Distance Kernels (REDK) 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 a new method to construct positive definite kernels from classical time-warp or edit measures. This method is quite general and less restrictive than previous ones, although it requires to introduce a recursive regularizing term that allows to prove the definiteness of our kernels as a direct consequence of the Haussler's convolution theorem,
3) we show that the regularizing approach we develop applies also when a symmetric *corridor* is used to limit the search space used to evaluate the elastic measure,
4) we experiment and compare the proposed kernels on some time series classification tasks using a large variety of time series data sets to estimate in practice the benefit we can expect from such kernels.

The paper is organized as follows: the second section synthesizes the related works; the third section introduces the notation and mathematical background that is used throughout the paper. The fourth section develops the construction of a general REDK and details some instantiations derived from classical elastic measures. The fifth section gathers classification experimentation on a wide range of time series data and compares REDK accuracies with classical elastic measures. The sixth section proposes a conclusion and further research

• *P.F. Marteau and Sylvie Gibet are with the Univ. Bretagne Sud, UMR 6074, IRISA, F-56000 Vannes, France.*
E-mail: {Pierre-Francois.Marteau, Sylvie.Gibet}(AT)univ-ubs.fr

perspectives. Appendix A states the indefiniteness of classical elastic measures, and appendix B presents 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 [6] implementation of the symbolic edit distance [7] by Wagner and Fisher [8], the Smith and Waterman (SW) algorithm [9] 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 [10] or FASTA [11]. Similarly, dynamic time warping measures have been developed to evaluate similarity between digital time series or timestamped data [12], [13], and more recently [14], [15] propose elastic metrics dedicated to such digital data.

Our ability to construct kernels with elastic or time-warp properties from such *elastic distances* allowing to embed time series into vector spaces (Euclidean or not) has attracted attention (e.g. [2][16][17]). Indeed, significant benefits can be expected from 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) [18], [19], [20] 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 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 [2][17][21], 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 [22], [2]. 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. These methods [23] [24] 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 minimizing the distance between convex hulls formed from the two categories of data embedded in the pE space [17]. 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 [25], [26] address 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.

Other approaches try to use directly the elastic distance into the kernel construction, without any approximation or resampling process.These works are based on the work of Haussler on convolution kernels [1] 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 [1], Saigo et al [27] define, from the Smith and Waterman algorithm [9], 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. [5] have adapted this approach to times series alignments covering the DTW elastic distance. In fact, these previous studies have very general implications, the first one being that classical elastic measures can also be understood as the limit of a series of definite convolution kernels.

In this paper we tackle a new approach to the positive definiteness regularization of elastic kernels that follows directly the lines of the Haussler's theorem on convolu-

tion kernels. The condition to obtain positive definiteness is weaker than the one proposed in previous works [5], although it requires the addition of an explicit recursive regularizing term. This term is easy to evaluate and has the same complexity than the recursive equation that defines the elastic distance. Our regularization strategy is thus quite general, independent from the data (although a parameter can be optimized), and can be applied to a very large family of editing or dynamic time warping like distances. It applies also to some of their variants which exploit a symmetric *corridor* to speed-up the computation.

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 [28], [20] and [29] present a large synthesis of major results. We give hereinafter some basic definitions and some results derived from these previous references.

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

Definition 3.2: 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 3.3: 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 3.4: 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, according to [28], $\psi(x, y) = \varphi(x, x_0) + \overline{\varphi(y, x_0)} - \varphi(x, y) - \varphi(x_0, x_0)$ 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 result 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 $A_i^j = A(i)A(i+1)...A(j)$. A_i^j with $i > j$ is the null time series, e.g. Ω. Furthermore, if $i > |A|$ then by convention, $A(i) = \Omega$.

3.3 General Edit/Elastic distance on a sequence set

Definition 3.7: An edit operation is a pair $(a, b) \in ((S \times T) \cup \{\Omega\})^2$ written $a \rightarrow b$. The 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 sub-sequences σ and τ. We call $a \rightarrow b$ a substitution operation if $a \neq \Omega$ and $b \neq \Omega$, a delete operation if $b = \Omega$, an insert operation if $a = \Omega$.

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 Ω, 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 $\Gamma_{A(i) \rightarrow \Omega, B, j}(A_1^p, B_1^q)$ or $\Gamma_{\Omega, A, i \rightarrow B(j)}(A_1^p, B_1^q) \in \mathbb{R}$. To simplify this notation, we will simply write $\Gamma(A(i) \rightarrow B(j))$, $\Gamma(A(i) \rightarrow \Omega_B(j))$ or $\Gamma(\Omega_A(i) \rightarrow B(j))$ with the understanding that the suppression cost $\Gamma(A(i) \rightarrow \Omega_B(j))$ (respectively the insertion cost $\Gamma(\Omega_A(i) \rightarrow B(j))$)

may depend on the current location j in sequence B (respectively on the location i in sequence A).

Hence we consider that $\Omega_X(k)$ is a function from $\mathbb{U} \times \mathbb{N}$ to $(S \times T) \cup \{\Omega\}$.

Definition 3.8: 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

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

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. $\Gamma(A(p) \rightarrow \Omega_B(q)) = \Gamma(\Omega_A(p) \rightarrow B(q)) = 1$ while the *substitution* cost is null if $A(p) = B(q)$ or 1 otherwise. Thus, for this distance, we consider that the functional term $\Omega_X(k) = \Omega$ for all X and k, since the suppression and insertion costs do not depend on the suppressed or inserted element respectively.

3.3.2 Dynamic time warping

The DTW similarity measure δ_{dtw} [12][13] is defined according to the previous notations such as:

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

where and $d_{LP}(x, y)$ is the LP norm of vector $(x - y)$ in S, and so for DTW, $\Gamma(A(p) \rightarrow \Omega_B(q)) = \Gamma(A(p) \rightarrow B(q)) = \Gamma(\Omega_A(p) \rightarrow B(q)) = d_{LP}(a(p), b(q))$. Thus, for this distance, $\Omega_X(k) = X(k)$ for all X and k. 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 $S \times T$. Furthermore, δ_{dtw} does not comply with the triangle inequality as shown in [14].

3.3.3 Edit Distance with real penalty

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

with

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

where g is a constant in S. For this distance, $\Omega_X(k) = g$ for all X and k.

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 [14], 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) [30], [15] is defined similarly to the edit distance defined for string [7][8]. The similarity between any two time series A and B of finite length, respectively p and q is defined as:

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

with

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

where λ is a positive constant that represents a gap penalty. 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 ν is a non negative constant which characterizes the *stiffness* of the δ_{twed} elastic measure along the *time* axis. Furthermore, for this distance, we consider that the functional term $\Omega_X(k) = \Omega$ for all X and k, since the suppression and insertion costs do not depend on the suppressed or inserted element respectively.

3.4 Indefiniteness of elastic distance kernels

In appendix A, we give counter examples, one for each previously defined elastic distance, 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 makes possible, 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 summation ($\sum$) operator. Instead of keeping only one of the best alignment paths, the new kernel will sum up the costs of all the existing sub-sequence alignment paths with some weighting factor that will favor *good* alignments while penalizing bad alignments. In addition, this weighting factor can be optimized. This principle has been applied successfully to the Smith and Waterman symbolic distance which is also known to be indefinite [27], and more recently to dynamic time warping kernels for time series alignment [5]. If, basically, we are following the same objective, the approach that we propose is quite different since it is based on a regularization principle which is simply and directly expressed into the recursive equation defining the elastic distance. First we replace the *min* (or *max*) operator by a summation operator and introduce a symmetric *corridor* function to optionally limit the summation. Then we add a regularizing recursive term such that the proof of the positive definiteness property can be understood as a direct consequence of the Haussler's convolution theorem, as shown in appendix B. Basically, all is reduced to the proper inventory of pairs of symmetric alignment paths.

4.1 Recursive accumulation of *f-cost* products

Definition 4.1: A function $\mathcal{C}(.,.) : \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}$ is called a Recursive Accumulation of *f-cost* Products (RAfP) if, for any pair of sequences A_1^p, B_1^q, there exist a function $f : \mathbb{R} \rightarrow \mathbb{R}$ and three indicator functions h_d, h_s and $h_i : \mathbb{N}^2 \rightarrow \{0, 1\}$ such that the following recursive equation is satisfied:

$$\mathcal{C}(A_1^p, B_1^q) = \frac{1}{c} \sum \begin{cases} h_d(p,q)\mathcal{C}(A_1^{p-1}, B_1^q) f(\Gamma_d(A,p,B,q)) & del \\ h_s(p,q)\mathcal{C}(A_1^{p-1}, B_1^{q-1}) f(\Gamma_s(A,p,B,q)) & sub \\ h_i(p,q)\mathcal{C}(A_1^p, B_1^{q-1}) f(\Gamma_i(A,p,B,q)) & ins \end{cases} \quad (1)$$

Where c is a non negative constant. This recursive definition relies on an initialization. To that end we set $\mathcal{C}(\Omega, \Omega) = \xi$, where ξ is a real non negative constant, typically $\xi = 1$, and Ω the null sequence. Note that if no path satisfying the indicator functions exists when aligning time series A and B, then $\mathcal{C}(A, B) = 0$.

Furthermore, according to the indicator functions h_d, h_s and h_i, this type of construction sums up the multiplication of the local quantities $f(\Gamma(A(i) \rightarrow B(j)))$ that we call *f-costs* for all or for some of the possible alignment paths that exist between the two time series, the concept of alignment path being precisely defined in appendix B (see definition B.3).

In this paper we are specifically concerned with the following two symmetric RAfP families. The first one (Eq. 2), characterized by $\forall (p,q) \in \mathbb{N}^2$, $h_d(p,q) = h(p-1,q)$, $h_i(q,p) = h(p,q-1)$ and $h_s(p,q) = h(p-1,q-1)$, accumulates the *f-cost* products along all the alignment paths between the two time series (Eq.2) that exist inside the symmetric *corridor* defined by $h(p,q)$. For the case depicted in Figure 1, $h(p,q) = 1$, whenever $|p-q| \leq 3$, $h(p,q) = 0$ otherwise. We denote this RAfP function $\mathcal{C}_h$.

$$\mathcal{C}_h(A_1^p, B_1^q) = \frac{1}{c} \sum \begin{cases} h(p-1,q)\mathcal{C}_h(A_1^{p-1}, B_1^q) f(\Gamma(A(p) \rightarrow \Omega_B(q))) \\ h(p-1,q-1)\mathcal{C}_h(A_1^{p-1}, B_1^{q-1}) f(\Gamma(A(p) \rightarrow B(q))) \\ h(p,q-1)\mathcal{C}_h(A_1^p, B_1^{q-1}) f(\Gamma(\Omega_A(p) \rightarrow B(q))) \end{cases} \quad (2)$$

The second RAfP function (Eq. 3) accumulates the *f-cost* products of all mappings characterized by simultaneous insertions, deletions and substitutions. Hence the substitution, deletion and insertion operations are simultaneously performed on the same index value. This function is thus characterized by $\forall (p,q) \in \mathbb{N}^2$, $h_d(p,q) = h(p-1,q)$, $h_i(q,p) = h(p,q-1)$ where h is a symmetric function, and $h_s(p,q) = \delta_{p,q} h(p-1,q-1)$, where δ is the Kronecker's symbol (Eq.3). We denote this second RAfP function $\mathcal{C}_{h,\delta}$.

$$\mathcal{C}_{h,\delta}(A_1^p, B_1^q) = \frac{1}{c} \sum \begin{cases} h(p-1,q)\mathcal{C}_{h,\delta}(A_1^{p-1}, B_1^q) f(\Gamma(\Omega_A(p) \rightarrow \Omega_B(p)) \\ \delta_{p,q} h(p-1,q-1) \cdot \\ \qquad \mathcal{C}_{h,\delta}(A_1^{p-1}, B_1^{q-1}) f(\Gamma(A(p) \rightarrow B(q))) \\ h(p,q-1)\mathcal{C}_{h,\delta}(A_1^p, B_1^{q-1}) f(\Gamma(A(q) \rightarrow B(q)) \end{cases} \quad (3)$$

4.2 Recursive Edit Distance Kernel - Main result

Definition 4.2: We call Recursive Edit Distance Kernel (REDK) the function $\mathcal{K}(.,.) = \mathcal{C}_h(.,.) + \mathcal{C}_{h,\delta}(.,.) : \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}$.

The following theorem, that relates to the ones presented in [5] and [31], establishes sufficient conditions on $f(\Gamma(a \rightarrow b))$ for a REDK function to be definite positive, and thus is a basis for the construction of definite REDK. The proof of this theorem, that is given in Appendix B, is a direct consequence of the Hausslers *R-convolution* theorem [1]. The sufficient condition that is proposed here is quite general and less restrictive than those proposed by [5] and [31]. The avenue that is taken here only requires that the local kernel $k(x,y)$ is PD. In previous works, the condition for positive definiteness is that the local kernel $k(x,y)$ and the ratio $k(x,y)/(k(x,y)+1)$ are jointly PD. Our new condition is obviously weaker than the one proposed in [9]. Furthermore, our REDK construction applies to a wider range of edit or time warp distances developed either for symbolic sequence or time series matching.

Theorem 4.3: Definiteness of REDK:

REDK is a positive definite kernel on $\mathbb{U} \times \mathbb{U}$ iff the local kernel $k(x,y) = f(\Gamma(x \rightarrow y))$ is a positive definite kernel on $((S \times T) \cup \{\Omega\})^2$.

A sketch of proof for theorem 4.3 is given in the appendix B.

As the cost function Γ is, in general, conditionally negative definite, choosing for $f(.)$ the exponential ensures that $f(\Gamma(x \rightarrow y))$ is a positive definite kernel [32].

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 [33] and thus forms an infinite dimension feature space. Note here that Cuturi et al. [5] state that to ensure the definiteness of the DTW-REDK kernel, not only $f(\Gamma(x \rightarrow y))$, but also $f(\Gamma(x \rightarrow y))/(1 + f(\Gamma(x \rightarrow y)))$ need to be PD kernels, which forms a stronger condition than the one we propose.

4.2.1 *Computational cost of REDK*

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 any RAfP function and therefore of $\mathcal{K}(A,B)$ lead 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 [13] 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. Note that any kind of corridor can be specified using the indicator functions h_d, h_s and h_i.

4.3 *Exponentiated* REDK

Definition 4.4: The following equations define the two RAfP function instances entering into the construction of the exponentiated REDK, considering $f(\Gamma(. \rightarrow .)) = \frac{1}{c}.e^{-\nu'\Gamma(. \rightarrow .)}$ with $c > 0$

$$\mathcal{C}_{e,h}(A_1^p, B_1^q) = \frac{1}{c} \sum \begin{cases} h(p-1,q)\mathcal{C}_{e,h}(A_1^{p-1}, B_1^q)\ e^{-\nu'\Gamma(A(p) \rightarrow \Omega_B(q))} \\ h(p-1,q-1)\mathcal{C}_{e,h}(A_1^{p-1}, B_1^{q-1})\ e^{-\nu'\Gamma(A(p) \rightarrow B(q))} \\ h(p,q-1)\mathcal{C}_{e,h}(A_1^{p}, B_1^{q-1})\ e^{-\nu'\Gamma(\Omega_A(p) \rightarrow B(q))} \end{cases} \quad (4)$$

$$\mathcal{C}_{e,h,\delta}(A_1^p, B_1^q) = \frac{1}{c} \sum \begin{cases} h(p-1,q)\mathcal{C}_{e,h,\delta}(A_1^{p-1}, B_1^q) e^{-\nu'\Gamma(\Omega_A(p) \rightarrow \Omega_B(p))} \\ \delta_{pq} h(p-1,q-1)\mathcal{C}_{e,h,\delta}(A_1^{p-1}, B_1^{q-1}) e^{-\nu'\Gamma(A(p) \rightarrow B(q))} \\ h(p,q-1)\mathcal{C}_{e,h,\delta}(A_1^{p}, B_1^{q-1}) e^{-\nu'\Gamma(A(q) \rightarrow B(q))} \end{cases} \quad (5)$$

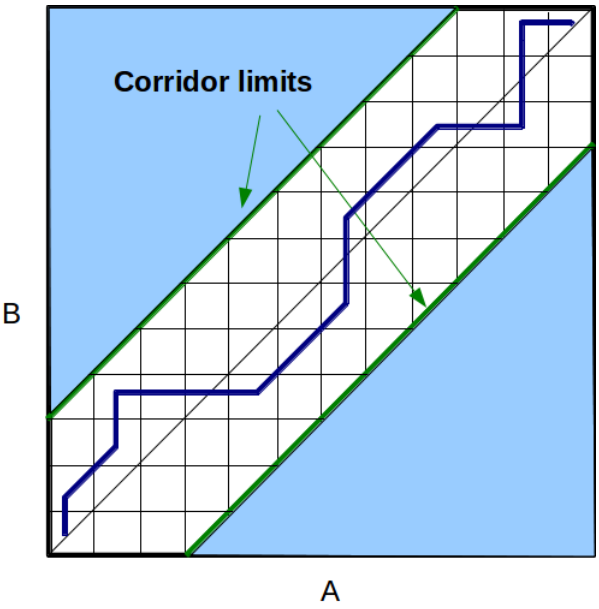

Fig. 1. Example of alignment paths existing when matching time series A and B. In red dotted line a path whose editing *f-cost* product is accumulated by $\mathcal{C}_h$ and $\mathcal{C}_{h,\delta}$, in blue plain line, a path whose editing cost product is accumulated by $\mathcal{C}_h$ but not $\mathcal{C}_{h,\delta}$. A symmetric corridor of any form can be used to reduce the number of admissible alignment paths that are inventoried by a RAfP

where ν' is a *stiffness* parameter that weights 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, therefore $\lim_{\nu' \rightarrow +\infty} -1/\nu' log(\mathcal{K}_e(A,B))$ does not coincide with the elastic distance δ that involves the same corresponding elementary costs, although we expect it to be close.

The constant $1/c$ (typically $1/3$) is used to maintain the global accumulation functions $\mathcal{C}_{e,h}$ and $\mathcal{C}_{e,h,\delta}$ in a range that is computationally acceptable.

The alignment cost of two null time series being 0, we set $\mathcal{C}_{e,h}(\Omega,\Omega) = \mathcal{C}_{e,h,\delta}(\Omega,\Omega) = 1$ in this exponentiation context.

Proposition 4.5: Definiteness of the exponentiated REDK instance, $\mathcal{K}_e$:
$\mathcal{K}_e(.,.)$ is positive definite for the cost functions $\Gamma(A(p) \rightarrow \Omega_B(q))$, $\Gamma(A(p) \rightarrow B(q))$ and $\Gamma(\Omega_A(p) \rightarrow B(q))$ involved in the computation of the δ_{lev}, δ_{dtw} and δ_{erp} distances.

The proof of proposition 4.5 is straightforward and is omitted.

The local cost function $\Gamma(\Omega_A(p) \rightarrow B(q))$ involved in the computation of the δ_{twed} does not respect the

condition of theorem 4.3 and thus the local kernel is not guaranteed yet to be p.d. for this distance.

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

4.3.1 Interpretation of the exponentiated REDK

In a REDK kernel, each alignment path is assigned with a product of *f-costs* that is the multiplication of the local *f-cost* functions attached to each edge of the path. For exponentiated REDK, the local *f-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 this case the REDK, 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 REDK 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 [27], 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 distance-matrix (obtained from pairwise 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

The purpose of this experiment is to estimate the benefit one can expect from using PD elastic kernels instead of indefinite ones in SVM classification. It is clear that the the Sequential Minimal Optimization (SMO) involved in the SVM learning procedure [34] [35] is able to handle indefinite kernels, but, as the convergence toward a global extremum is not guaranteed (the optimization is problem is not convex), we can expect some better accuracy for definite kernels especially when the pairwise distance matrix derived from the train data set is far from definiteness. This is precisely what our experiment targets to demonstrate.

To that end, we empirically evaluate the effectiveness of some REDK kernels comparatively to Gaussian Radial Basis Function (RBF) Kernels or elastic distance substituting kernels [2] 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 [36]. 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 [19], [37] 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 10-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 10 stratified subsets $TRAIN_1, TRAIN_2, \cdots, TRAIN_{10}$; 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 10 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 [38] to implement the SVM classifiers.

5.1 Experimenting with REDK

We tested the exponentiated REDK kernel based on the $\delta_{erp}, \delta_{dtw}, \delta_{twed}$ distance costs. We consider respectively the positive definite $REDK_{erp}, REDK_{dtw}, REDK_{twed}$ 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 [21], e.g. SVM δ_{erp}, SVM δ_{dtw}, SVM δ_{twed},

- and their corresponding REDK kernel, SVM $REDK_{erp}$, SVM $REDK_{dtw}$, SVM $REDK_{twed}$.

For δ_{erp}, δ_{twed}, $REDK_{erp}$ and $REDK_{twed}$ we used the L1-norm, while the L2-norm has been implemented for δ_{dtw} and $REDK_{dtw}$, a classical choice for DTW [39].

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 $REDK_{erp}$ 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 $REDK_{twed}$ kernel.

The kernels exploited by the SVM classifiers are the Gaussian Radial Basis Function (RBF) kernels $K(A, B) = e^{\delta(A,B)^2/(2\sigma^2)}$ where δ stands for δ_{erp}, δ_{dtw}, δ_{twed}, $REDK_{erp}$, $REDK_{dtw}$, $REDK_{twed}$. To limit the search space for the SVM meta parameters, the pairwise distances or kernels values are normalized using the *min* and *max* values calculated on the TRAIN data, basically

- $\delta = (\delta - \delta_{min})/(\delta_{max} - \delta_{min})$ for $\delta = \delta_{erp}$, δ_{dtw} or δ_{twed}, and
- $\delta = exp\big((log(\delta) - log(\delta_{min}))/(log(\delta_{max}) - log(\delta_{min}))\big)$ for $\delta = REDK_{erp}$, $REDK_{dtw}$, $REDK_{twed}$.

Meta parameter C is selected from $\{2^{-5}, 2^{-4}, ..., 1, 2, ..., 2^{10}\}$, and σ^2 from $\{2^{-5}, 2^{-4}, ..., 1, 2, ..., 2^{14}\}$. The best values are obtained using the 10-folds cross validation procedure.

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

The optimization procedure is as follows:

- for each value in $\mathcal{G}$, we train a SVM $REDK_*$ classifier on the training dataset using the previously described 10-folded cross validation procedure to select the SVM meta parameters C 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.

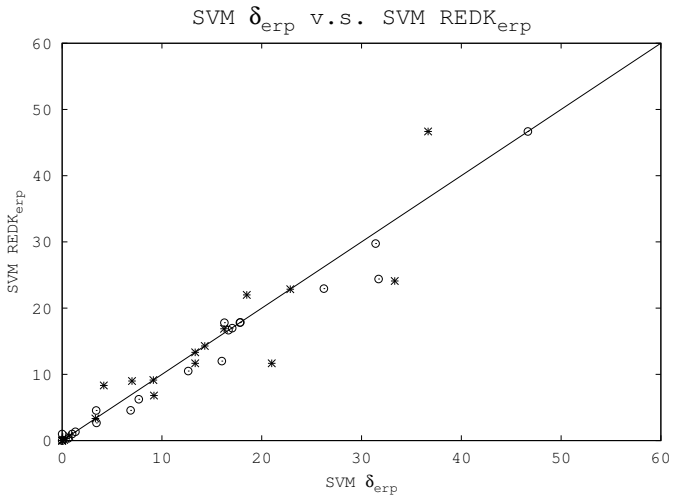

Fig. 2. Comparison of error rates (in %) between two SVM classifiers, the first one based on the δ_{erp} substituting kernel (SVM δ_{erp}), and the second one based on a REDK kernel induced by the δ_{erp} distance (SVM $REDK_{erp}$). The straight line has a slope of 1.0 and dots correspond, for the pair 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 $REDK_{erp}$ has a lower (resp. higher) error rate than distance SVM δ_{erp}

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

5.2 Discussion

5.2.1 REDK 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 REDK kernels (SVM $REDK_{erp}$, SVM $REDK_{dtw}$ and SVM $REDK_{twed}$).

In this experiment, we show that the SVM classifiers clearly outperform in general the 1-NN classifiers that are used as a baseline. But the interesting results reported in tables 2 and 3 and figures 2, 3 and 4 is that SVM $REDK_{erp}$ and SVM $REDK_{twed}$ perform slightly better than SVM δ_{erp} and SVM δ_{twed} respectively, and the SVM $REDK_{dtw}$ 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}, but another explanation could be related to the the SMO optimization involved into the SVM learning. SVM δ_{dtw} poorly behaves compared to the other tested classifiers, probably because the SVM opti-

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

DATASET	$\delta_{erp} : g; C; \sigma$	$REDK_{erp} : g; \nu'; C; \sigma$	$\delta_{dtw} : C; \sigma$	$REDK_{dtw} : \nu'; C; \sigma$	$\delta_{twed} : \Omega; \nu; C; \sigma$	$REDK_{twed} : \Omega; \nu; \nu'; C; \sigma$
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, $REDK_{erp}$, dtw and $REDK_{dtw}$ 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. Bold faces indicates the lowest error rate between the pairs of SVMs (K_{erp}, $REDK_{erp}$) and (K_{dtw}, $REDK_{dtw}$).

DATASET	1-NN δ_{erp}	SVM δ_{erp}	SVM $REDK_{erp}$	1-NN δ_{dtw}	SVM δ_{dtw}	SVM $REDK_{dtw}$
Synthetic control	0.67|3.7	**0.33**|**1**	**0.33**|**1**	0|1.33	**0**|1.33	**0**|**1**
Gun-Point	6.12|4	**0**|**1.33**	**0**|**1.33**	18.36|9.3	6|8	**0**|**.67**
CBF	0|0.33	**3.33**|3.44	**3.33**|**2.67**	0|0.33	3.33|1.67	**0**|**0.22**
Face (all)	10.73|20.18	**.71**|17.04	**.71**|**16.98**	6.8|19.23	4.29|14.97	**1.25**|**7.89**
OSU Leaf	30.15|40.08	**18.5**|31.41	22|**29.75**	29|44.63	28|34.3	**21**|**25.62**
Swedish Leaf	11.02|12	9.2|7.68	**6.8**|**6.24**	24.65|20.8	20.6|17.92	**6.8**|**6.72**
50 Words	19.38|28.13	**16.22**|**16.26**	16.89|17.80	33.18|31	30.89|38.90	**16.44**|**25.93**
Trace	10.01|17	**0**|**0**	**0**|1	0|0	**0**|**0**	**0**|**0**
Two Patterns	0|0	**0**|**0**	**0**|**0**	0|0	**0**|**0**	**0**|**0**
Wafer	.1|0.9	.2|0.62	**0.1**|**0.34**	1.4|2.01	2.4|3.11	**0.2**|**0.42**
face (four)	4.35|10.2	**4.17**|**3.41**	8.33|4.55	26.09|17.05	12.5|25	**8.33**|**3.41**
Ligthing2	11.86|14.75	13.33|26.23	**11.67**|**22.95**	13.56|13.1	16.67|**8.2**	**8.33**|16.39
Ligthing7	23.19|30.1	**22.86**|**17.81**	**22.86**|**17.81**	33.33|27.4	30.00|26.03	**12.86**|**17.81**
ECG	10.01|13	**7**|16	9|**12**	23.23|23	12|19	**6**|**11**
Adiac	35.99|37.85	33.33|31.71	**24.10**|**24.4**	40.62|39.64	36.41|35.29	**21.79**|**20.97**
Yoga	14.05|14.7	21|12.63	**11.67**|**10.5**	16.37|16.4	20.33|17.03	**11.67**|**11.47**
Fish	16.09|12	**9.14**|6.86	**9.14**|**4.57**	26.44|16.57	25.71|22.29	**7.43**|**4.57**
Coffee	25.93|25	**14.29**|**17.86**	**14.29**|**17.86**	14.81|17.86	**7.14**|**7.14**	**7.14**|**7.14**
OliveOil	17.24|16.67	**13.33**|**16.67**	**13.33**|**16.67**	13.79|13.33	**10**|**13.33**	13.33|**13.33**
Beef	68.97|50	**36.67**|**46.67**	46.67|**46.67**	46.67 |50	33.33|53.33	**30**|**46.67**
# Best Scores	-	**15**|10	**15**|**17**	-	5|5	**19**|**19**
# Uniquely Best Scores	-	3|4	**5**|**10**	-	1|1	**15**|**15**

mization process does not perform well. Nevertheless, the $REDK_{dtw}$ kernel based on δ_{dtw} greatly improves the classification results. To further explore the potential impact of indefiniteness on classification rates, we give in table 4 two quantified hints of deviation to conditionally definiteness for the distance-matrices corresponding to the δ_{dtw}, δ_{erp} and δ_{twed} distances. Since to be conditionally definite (negative) a pairwise distance-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 distance 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) matrix should be such that simultaneously $\#Pev = 1$ and $\Delta_p = 0$.

By examining the distance-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 further away from a conditionally defi-

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 $REDK_{twed}$ 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. Bold faces indicates the lowest error rate between the pair of SVMs (K_{twed}, $REDK_{twed}$).

DATASET	1-NN δ_{twed}	SVM δ_{twed}	SVM $REDK_{twed}$
Synthetic control	1|2.33	**0**|1.33	**0**|**1**
Gun-Point	0|1.33	2|**0.67**	**0**|**0.67**
CBF	0|0.67	**3.33**|3.33	**3.33**|**1.77**
Face (all)	1.43|18.93	**0.54**|17.10	0.71|**16.15**
OSU Leaf	17.59|24.79	**15**|**17.36**	17|20.66
Swedish Leaf	8.82|10.24	7|6.56	**6.4**|**5.76**
50 Words	18.26|18.9	16.22|**14.07**	**15.78**|16.04
Trace	1|5	**0**|**0**	**0**|1
Two Patterns	0|0.12	**0**|**0**	**0**|**0**
Wafer	0.1|.86	0.3|0.34	**0.1**|**0.2**
face (four)	8.7|3.41	**8.33**|**2.27**	**8.33**|3.41
Ligthing2	13.56|21.31	20|32.15	**10**|**21.31**
Ligthing7	24.64|24.66	**27.14**|**21.92**	**27.14**|**21.92**
ECG	13.13|10	**11**|8	12|**7**
Adiac	36.25|37.6	28.72|30.95	**24.85**|**24.30**
Yoga	19.06|12.97	11.67|**9.87**	**11.33**|10.46
Fish	12.07|5.14	**6.29**|**3.43**	6.86|**3.43**
Coffee	18.52|21.43	17.86|21.43	**14.29**|**17.86**
OliveOil	11.11|16.67	**13.33**|**13.33**	**13.33**|16.67
Beef	58.62|53.3	**36.67**|53.33	46.67|**50**
# Best Scores	-	12|10	**15**|**14**
# Uniquely Best Scores	-	5|5	**8**|**10**

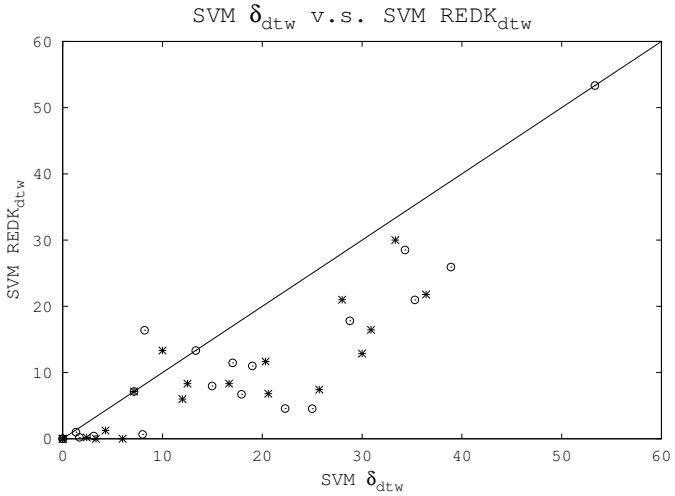

Fig. 3. Comparison of error rates (in %) between two SVM classifiers, the first one based on the δ_{dtw} substituting kernel (SVM δ_{dtw}), and the second one based on a REDK induced by the δ_{dtw} distance (SVM $REDK_{dtw}$)). The straight line has a slope of 1.0 and dots correspond, for the pair of 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 $REDK_{dtw}$ has a lower (resp. higher) error rate than distance δ_{dtw}

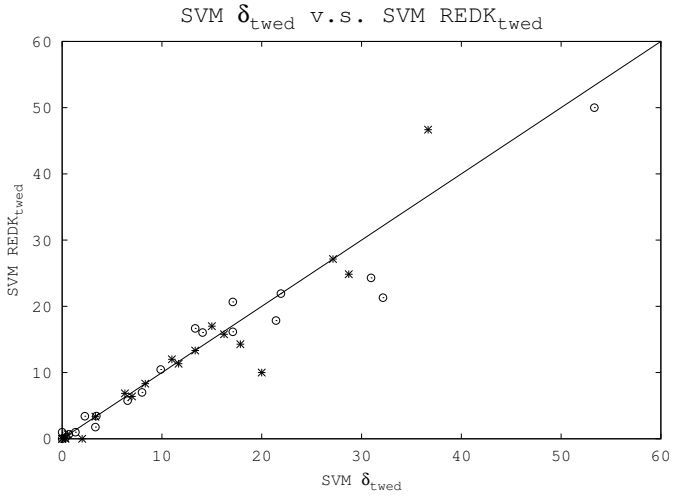

Fig. 4. Comparison of error rates (in %) between two SVM classifiers, the first one based on the δ_{twed} substituting kernel (SVM δ_{twed}), and the second one based on a REDK induced by the δ_{erp} distance (SVM $REDK_{twed}$). The straight line has a slope of 1.0 and dots correspond, for the pair 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 $REDK_{twed}$ has a lower (resp. higher) error rate than distance SVM δ_{twed}

TABLE 4
Analysis of the deviation to conditionally definiteness for the distance-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%

nite 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 quantifiable by computing the number of positive eigenvalues, as well as their amplitudes, of the pairwise distance matrices. For datasets such as *FaceAll, OSULeaf, SwedishLeaf, 50words, Adiac, Two_Patterns, Fish* etc., all the three distances lead to indefinite pairwise distance matrices. The REDK regularization brings in general a significant improvement, specifically when the number and amplitudes of the extra positive eigenvalues are high. Furthermore, for datasets of small sizes (such as *CBF*, *Beef*, *Coffee*, *OliveOil*, etc.), δ_{erp} and δ_{twed} produce conditionally definite matrices where δ_{dtw} does not. One can see that the regularization brought by REDK is much more effective on δ_{dtw} on these data set. Nevertheless, some datasets are better classified by SVM that use directly the distance substituting kernel instead of the derived REDK regularized kernel.

To conclude, our experiment clearly shows that for problems involving distance matrices *far* from definiteness (more than 5-10% of eigenvalues are positive, see Table 4), the REDK regularized version outperforms the original elastic distance substituting kernel. This is particularly true for δ_{dtw}. For δ_{twed}, or even δ_{erp} distances, most of the training sets lead to pairwise distance-matrices that are either conditionally definite negative or close to CND matrices (with very few extra positive eigenvalues). Nevertheless, in most cases, the REDK-δ_{erp} and REDK-δ_{twed} outperformed δ_{erp} and δ_{twed} respectively, showing some robustness of the REDK regularization. The main drawback of REDK is the extra parameter ν'. This 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 more complex to optimize.

6 Conclusion

Following the work on convolution kernels [1] and local alignment kernels defined for string processing around the Smith and Waterman algorithm [9] [27], or defined for time series on the basis of the DTW similarity measure [5], we have addressed the definiteness of elastic distances through the construction of positive definite kernels built from the recursive definitions of elastic (editing) distances themselves and achieve some extension and generalization of these previous results.

Contrary to other approaches that rely on the regularization of the Gram-matrix [23][24], this approach does not depend at all on the data, excpet for the optimization of a regularizing parameter (ν'). By adding an extra recursive term we achieve some simple sufficient conditions, that are weaker than those proposed in [27] or [5], allowing to build positive definite exponentiated REDK. These conditions are basically satisfied by any classical elastic distance defined by a recursive equation. In particular they can be applied to the edit distance, the well known Dynamic Time Warping measure and to 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. Furthermore, our results apply when a symmetric *corridor* is used to limit the search space required to evaluate the elastic distance, thus reducing consequently

the computation time, which, in the worse case, i.e. without corridor, remains quadratic.

The experiments conducted on a variety of time series datasets show that the positive definite REDKs outperforms in general the indefinite elastic distances from which they are derived, when considering 1-NN and SVM classification tasks. This is mostly striking when the Gram-matrix evaluated on the train data sets is *far* from definiteness, which is the case when DTW is used. Recent experiment in isolated gesture recognition [40] corroborate this observation.

APPENDIX A
INDEFINITENESS OF SOME 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 [41]. Let us consider the subset of sequences $V = \{abc, bad, dab, adc, bcd\}$ that leads to the following distance matrix

$$M_{lev}^V = \begin{pmatrix} 0 & 3 & 2 & 1 & 2 \\ 3 & 0 & 2 & 2 & 1 \\ 2 & 2 & 0 & 3 & 3 \\ 1 & 2 & 3 & 0 & 3 \\ 2 & 1 & 3 & 3 & 0 \end{pmatrix} \tag{6}$$

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

$$M_{dtw}^V = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \end{pmatrix} \tag{7}$$

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 $\Omega = 0.0$ we get the following matrix:

$$M_{twed}^V = \begin{pmatrix} 0 & 2 & 7 & 9 & 6 & 7 & 5 & 5 & 10 & 9 \\ 2 & 0 & 5 & 9 & 4 & 5 & 3 & 3 & 8 & 9 \\ 7 & 5 & 0 & 6 & 7 & 4 & 6 & 2 & 5 & 10 \\ 9 & 9 & 6 & 0 & 13 & 10 & 12 & 8 & 1 & 4 \\ 6 & 4 & 7 & 13 & 0 & 5 & 3 & 5 & 12 & 9 \\ 7 & 5 & 4 & 10 & 5 & 0 & 2 & 6 & 9 & 6 \\ 5 & 3 & 6 & 12 & 3 & 2 & 0 & 4 & 11 & 8 \\ 5 & 3 & 2 & 8 & 5 & 6 & 4 & 0 & 7 & 10 \\ 10 & 8 & 5 & 1 & 12 & 9 & 11 & 7 & 0 & 5 \\ 9 & 9 & 10 & 4 & 9 & 6 & 8 & 10 & 5 & 0 \end{pmatrix} \tag{8}$$

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_{twed}^V has no definiteness.

A.4 The Edit Distance with Real Penalty

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

$$M_{erp}^V = \begin{pmatrix} 0 & 2 & 3 & 3 & 4 & 5 & 4 & 2 & 4 & 5 \\ 2 & 0 & 3 & 5 & 2 & 3 & 2 & 2 & 4 & 5 \\ 3 & 3 & 0 & 4 & 3 & 2 & 3 & 1 & 3 & 4 \\ 3 & 5 & 4 & 0 & 7 & 6 & 7 & 5 & 1 & 2 \\ 4 & 2 & 3 & 7 & 0 & 3 & 2 & 2 & 6 & 5 \\ 5 & 3 & 2 & 6 & 3 & 0 & 1 & 3 & 5 & 4 \\ 4 & 2 & 3 & 7 & 2 & 1 & 0 & 2 & 6 & 5 \\ 2 & 2 & 1 & 5 & 2 & 3 & 2 & 0 & 4 & 5 \\ 4 & 4 & 3 & 1 & 6 & 5 & 6 & 4 & 0 & 1 \\ 5 & 5 & 4 & 2 & 5 & 4 & 5 & 5 & 1 & 0 \end{pmatrix} \tag{9}$$

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_{erp}^V has no definiteness.

APPENDIX B
PROOF OF OUR MAIN RESULT

B.1 Proof of theorem 4.3

Definition B.1: Let $\mathbb{U}_n$ be the subset of $\mathbb{U}$ containing all the sequences whose lengths are lower or equal to n.

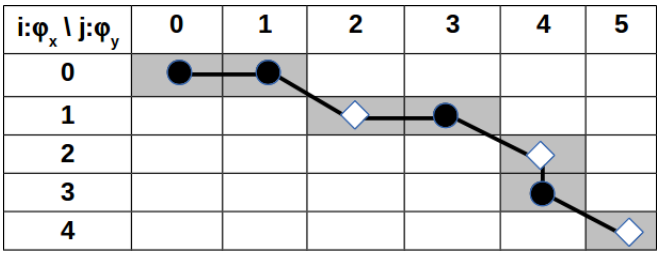

Fig. 5. Example of an alignment path corresponding to the alignment map $(0,0)(0,1)(1,2)(1,3)(2,4)(3,4)(4,5)$. The white squares correspond to substitution operations and black circles to either deletion or insertion operations.

Definition B.2: For all $(a,b) \in ((S \times T) \cup \{\Omega\})^2$ let $\kappa(a,b)$ be the local kernel defined as follows:
$\kappa(a,b) = f(\Gamma(a \rightarrow b))$
where Ω stands for the *null* sequence.

Definition B.3: Let π be an ordered alignment map between two finite non empty sequences of successive integers of the form $0,..,n$. Basically π is a finite sequence of pairs of integers $\pi(l) = (i_l, j_l)$ for $l \in \{0,...,|\pi|-1\}$, satisfying the following conditions

- $0 \leq i_l, \forall l \in 0,..,|\pi|-1$
- $i_l \leq i_{l-1}+1, \forall l \in 1,..,|\pi|-1$
- $j_l \leq j_{l-1}+1, \forall l \in 1,..,|\pi|-1$
- $i_{l-1} < i_l$ or $j_{l-1} < j_l, \forall l \in \{1,..,|\pi|-1\}$

$\pi_x(l) = i_l$ and $\pi_y(l) = j_l$ are the two coordinate access functions for the l^{th} pair of mapped integers so that $\pi(l) = (\pi_x(l), \pi_y(l))$.

For all $n \geq 1$, let $\mathcal{M}_n$ be the set of alignment maps π such that the two sets of integers mapped by π are $\{1 \cdots n\} \times \{1 \cdots n\}$. By convention we set $\mathcal{M}_0 = \emptyset$.

As shown in figure 5, there exists a direct correspondence between an alignment map and an alignment path, i.e. a finite sequence of editing operations.

Definition B.4: For all n we introduce two projections, basically two vectorized representations, $\varphi_{\pi_x} : \mathbb{U}_n \rightarrow ((S \times T) \cup \{\Omega\})^{2n}$ and $\varphi_{\pi_y} : \mathbb{U}_n \rightarrow ((S \times T) \cup \{\Omega\})^{2n}$ induced uniquely by any alignment map $\pi \in \mathcal{M}_n$.

The principle for the construction of these two unique projections, given any sequence A and any alignment map $\pi \in \mathcal{M}_n$, is straightforward. With the convention that if $k > |A|$ then $A(k) = \Omega_A(|A|)$ and $\Omega_A(k) = \Omega_A(|A|)$, during the course along π of length $L = |\pi|$ at step l, $l \in \{1 \cdots L\}$, we apply the following rules:

i) if both indexes $\pi_x(l)$ and $\pi_y(l)$ increase , then, we insert $A(\pi_x(l))$ in $\varphi_{\pi_x}(A)$ and $A(\pi_y(l))$ in $\varphi_{\pi_y}(A)$.

ii) if only index $\pi_x(l)$ increases, then we insert $A(\pi_x(l))$ in $\varphi_{\pi_x}(A)$ and $\Omega_A(\pi_y(l))$ in $\varphi_{\pi_y}(A)$, with the convention that, if $\pi_x(l) > |A|$, $A(\pi_x(l)) = \Omega_A(\pi_x(l))$

iii) if only index $\pi_y(l)$ increases, then we insert $\Omega_A(\pi_x(l))$ in $\varphi_{\pi_x}(A)$ and $A(\pi_y(l))$ in $\varphi_{\pi_y}(A)$.

iv) when we reach the end of π, if the lengths of $\varphi_{\pi_x}(A)$ (respectively $\varphi_{\pi_y}(A)$) is shorter than $2n$, then we insert Ω into the remaining dimensions.

For example, for a sequence A of length 5 the two projections in $((S \times T) \cup \{\Omega\})^{2\times 5}$ corresponding to the alignment path depicted in Figure 5 are
$\varphi_{\pi_x}(A) = \Omega(1)\mathbf{A(1)}\Omega(2)\mathbf{A(2)}A(3)\mathbf{A(4)}\Omega\Omega\Omega\Omega$
$\varphi_{\pi_y}(A) = A(1)\mathbf{A(2)}A(3)\mathbf{A(4)}\Omega(5)\mathbf{A(5)}\Omega\Omega\Omega\Omega$
where $\Omega(i) = \Omega_A(i)$ is the symbol used for an insertion or deletion operation. This symbol depends on the elastic distance. For DTW, $\Omega(i) = A(i)$ if $0 < i \leq |A|$, or $\Omega(i) = \Omega$ if $i > |A|$.

Then, for any $A \in \mathbb{U}_n$ and $\pi \in \mathcal{M}_n$, we call $\mathcal{P}_\pi(A) = \{\varphi_{\pi_x}(A), \varphi_{\pi_y}(A)\}$ the set of projections (or parts) for sequence A induced by π. Note that all these projections are sequences whose lengths are $2n$. Even

Proposition B.5: If the local kernel $k(x,y) = f(\Gamma(x \rightarrow y))$ is positive definite on $((S \times T) \cup \{\Omega\})^2$ then $\forall n \geq 1$ and $\forall \pi \in \mathcal{M}_n$

$$K_\pi(A,B) = \sum_{\varphi(A) \in \mathcal{P}_\pi(A)} \sum_{\varphi(B) \in \mathcal{P}_\pi(B)} \prod_{p=1}^{2n} k(\varphi(A)_p, \varphi(B)_p) \tag{10}$$

is a p.d. kernel on $(\mathbb{U}_n)^2$.

Proof of lemma B.5 is a direct consequence of the Haussler's *R-convolution* kernel theorem [1]. Indeed, since $k(x,y)$ is a p.d. kernel on $((S \times T) \cup \{\Omega\})^2$, and, considering the sets of parts $\mathcal{P}_\pi(A)$ and $\mathcal{P}_\pi(B)$ associated respectively to the sequences A and B, the conditions for the Haussler's *R-convolution* are satisfied.

Note that $K_\pi(A,B)$ simply rewrites as

$$\begin{aligned} K_\pi(A,B) = & \prod_{p=1}^{2n} k(\varphi_{\pi_x}(A)_p, \varphi_{\pi_y}(B)_p)) \\ & + \prod_{p=1}^{2n} k(\varphi_{\pi_y}(A)_p, \varphi_{\pi_x}(B)_p)) \\ & + \prod_{p=1}^{2n} k(\varphi_{\pi_x}(A)_p, \varphi_{\pi_x}(B)_p)) \\ & + \prod_{p=1}^{2n} k(\varphi_{\pi_y}(A)_p, \varphi_{\pi_y}(B)_p)) \end{aligned} \tag{11}$$

Let $\mathcal{M}_{n,h} \subset \mathcal{M}_n$ be the subset of paths that are restricted by the symmetric corridor induced by any symmetric indicator function h entering into the construction of the RAfP functions $\mathcal{C}_h(.,.)$ and $\mathcal{C}_{h\delta}(.,.)$.

Proposition B.6: For any $n \in \mathbb{N}$, any $\pi \in \mathcal{M}_{n,h}$, and any

$(A, B) \in (\mathbb{U}_n)^2$, we have

$$\mathcal{K}(A, B) = \frac{1}{2} \sum_{\pi \in \mathcal{M}_{n,h}} K_\pi(A, B) \tag{12}$$

We state first that $\forall \pi \in \mathcal{M}_{n,h}$, $\exists! \tilde{\pi} \in \mathcal{M}_{n,h}$ such that $\pi_x = \tilde{\pi}_y$ and $\pi_y = \tilde{\pi}_x$. $(\pi, \tilde{\pi})$ is a pair of symmetrical paths with respect to the main diagonal.

Thus, since $\mathcal{C}_h(.,.)$ evaluates the sum of the *cost-product* along all paths in $\mathcal{M}_{n,h}$, that is

$$\mathcal{C}_h(A, B) = \sum_{\pi \in \mathcal{M}_{n,h}} \prod_{p=1}^{2n} k(\varphi_{\pi_x}(A)_p, \varphi_{\pi_y}(B)_p) \tag{13}$$

we can rewrite

$$\begin{aligned}
\mathcal{C}_h(A, B) = \sum_{\pi \in \mathcal{M}_{n,h}} \Bigg(& \frac{1}{2} \prod_{p=1}^{2n} k(\varphi_{\pi_x}(A)_p, \varphi_{\pi_y}(B)_p) \\
& + \frac{1}{2} \prod_{p=1}^{2n} k(\varphi_{\tilde{\pi}_x}(A)_p, \varphi_{\tilde{\pi}_y}(B)_p) \Bigg) \\
= \frac{1}{2} \sum_{\pi \in \mathcal{M}_{n,h}} \Bigg(& \prod_{p=1}^{2n} k(\varphi_{\pi_x}(A)_p, \varphi_{\pi_y}(B)_p) \\
& + \prod_{p=1}^{2n} k(\varphi_{\pi_y}(A)_p, \varphi_{\pi_x}(B)_p) \Bigg)
\end{aligned}$$

Similarly, for $\mathcal{C}_{h\delta}(.,.)$ we get

$$\begin{aligned}
\mathcal{C}_{h,\delta}(A, B) = \frac{1}{2} \sum_{\pi \in \mathcal{M}_{n,h}} \Bigg(& \prod_{p=1}^{2n} k(\varphi_{\pi_x}(A)_p, \varphi_{\pi_x}(B)_p) \\
& + \prod_{p=1}^{2n} k(\varphi_{\pi_y}(A)_p, \varphi_{\pi_y}(B)_p) \Bigg)
\end{aligned}$$

Hence, the expected result.

Proof of Proposition 4.3 (Definiteness of REDK)

Proposition B.6 states that $\mathcal{K}(.,.)$ is a finite sum of p.d. kernels defined on $(\mathbb{U}_n)^2$. According to the closure properties under the addition of p.d. kernels, we show that $\mathcal{K}(.,.)$ is a p.d. kernel defined on $(\mathbb{U}_n)^2$.

As this result is true for all n, one can see $\mathcal{K}(.,.)$ as a point-wise limit as n tends toward infinity of a p.d. kernel defined on $(\mathbb{U}_n)^2$. This establishes that $\mathcal{K}(.,.B)$ is a p.d. kernel on $\mathbb{U}^2$ $\square$.

ACKNOWLEDGMENTS

The authors 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. The authors also thank Dr. E. Keogh's team at UC Riverside for making available the time series data sets that have been used in this study.

REFERENCES

[1] D. Haussler, "Convolution kernels on discrete structures," University of California at Santa Cruz, Santa Cruz, CA, USA, Technical Report UCS-CRL-99-10, 1999. [Online]. Available: http://citeseer.ist.psu.edu/haussler99convolution.html

[2] B. Haasdonk and C. Bahlmann, "Learning with distance substitution kernels," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, C. Rasmussen, H. Blthoff, B. Schlkopf, and M. Giese, Eds. Springer Berlin Heidelberg, 2004, vol. 3175, pp. 220–227.

[3] J. P. Vert, H. Saigo, and T. Akutsu, "Local alignment kernels for biological sequences," in *Kernel Methods in Computational Biology*, B. Scholkopf, K. Tsuda, and J. Vert, Eds. MIT Press, 2004, pp. 131–154.

[4] C. Cortes, P. Haffner, and M. Mohri, "Rational kernels: Theory and algorithms," *J. Mach. Learn. Res.*, vol. 5, pp. 1035–1062, Dec. 2004.

[5] M. Cuturi, J.-P. Vert, . Birkenes, and T. Matsui, "A kernel for time series based on global alignments," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2, April 2007, pp. II–413–II–416.

[6] R. Bellman, *Dynamic Programming*. Princeton Univ Press, 1957, new Jersey.

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

[8] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *Journal of the ACM (JACM)*, vol. 21, pp. 168–173, 1973.

[9] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, pp. 195–197, 1981.

[10] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "A basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, pp. 403–410, 1990.

[11] W. Pearson, "Rapid and sensitive sequence comparisons with fasp and fasta," *Methods Enzymol*, vol. 183, pp. 63–98, 1990.

[12] V. M. Velichko and N. G. Zagoruyko, "Automatic recognition of 200 words," *International Journal of Man-Machine Studies*, vol. 2, pp. 223–234, 1970.

[13] H. Sakoe and S. Chiba, "A dynamic programming approach to continuous speech recognition," in *Proceedings of the Seventh International Congress on Acoustics, Budapest*, vol. 3. Budapest: Akadémiai Kiadó, 1971, pp. 65–69.

[14] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, ser. VLDB '04. VLDB Endowment, September 2004, pp. 792–803.

[15] P.-F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 306–318, Feb 2009.

[16] A. Hayashi, Y. Mizuhara, and N. Suematsu, "Embedding time series data for classification," in *Machine Learning and Data Mining in Pattern Recognition*, ser. Lecture Notes in Computer Science, P. Perner and A. Imiya, Eds. Springer Berlin Heidelberg, 2005, vol. 3587, pp. 356–365.

[17] B. Haasdonk, "Feature space interpretation of svms with indefinite kernels," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 4, pp. 482–492, April 2005.

[18] V. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, ISBN 0-471-03003-1, 1989.

[19] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: ACM, July 1992, pp. 144–152.

[20] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[21] D. Zhang, W. Zuo, D. Zhang, and H. Zhang, "Time series classification using support vector machine with gaussian elastic metric kernel," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ser. ICPR '10. Washington, DC, USA: IEEE Computer Society, August 2010, pp. 29–32.

[22] A. Woznica, A. Kalousis, and M. Hilario, "Distances and (indefinite) kernels for sets of objects," in *Data Mining, 2006. ICDM '06. Sixth International Conference on*, Dec. 2006, pp. 1151–1156.

[23] G. Wu, E. Y. Chang, and Z. Zhang, "Learning with non-metric proximity matrices," in *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*. New York, NY, USA: ACM, Nov. 2005, pp. 411–414.
[24] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti, "Similarity-based classification: Concepts and algorithms," *J. Mach. Learn. Res.*, vol. 10, pp. 747–776, Jun. 2009.
[25] K. Sivaramakrishnan and C. Bhattacharyya, "Time series classification for online tamil handwritten character recognition a kernel based approach," in *Neural Information Processing*, ser. Lecture Notes in Computer Science, N. R. Pal, N. Kasabov, R. K. Mudi, S. Pal, and S. K. Parui, Eds. Springer Berlin / Heidelberg, 2004, vol. 3316, pp. 800–805.
[26] K. Kumara, R. Agrawal, and C. Bhattacharyya, "A large margin approach for writer independent online handwriting classification," *Pattern Recogn. Lett.*, vol. 29, pp. 933–937, May 2008.
[27] H. Saigo, J.-P. Vert, N. Ueda, and T. Akutsu, "Protein homology detection using string alignment kernels," *Bioinformatics*, vol. 20, no. 11, pp. 1682–1689, Jul. 2004.
[28] C. Berg, J. P. R. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, Apr. 1984, vol. 100.
[29] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004.
[30] P. F. Marteau, "Time warp edit distance," VALORIA, Universite de Bretagne Sud, Tech. Rep., 2008, technical Report valoriaUBS-2008-3v, http://hal.archives-ouvertes.fr/hal-00258669/fr/.
[31] K. Shin and T. Kuboyama, "A generalization of haussler's convolution kernel mapping kernel and its application to tree kernels," *Journal of Computer Science and Technology*, vol. 25, no. 5, pp. 1040–1054, 2010.
[32] I. J. Schoenberg, "Metric spaces and positive definite functions," *Transactions of the American Mathematical Society*, vol. 44, no. 3, pp. 522–536, nov 1938.
[33] C. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," *Constructive Approximation*, vol. 2, no. 1, pp. 11–22, 1986.
[34] J. C. Platt, "Advances in kernel methods," B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208.
[35] P.-H. Chen, R.-E. Fan, and C.-J. Lin, "A study on smo-type decomposition methods for support vector machines," *Neural Networks, IEEE Transactions on*, vol. 17, no. 4, pp. 893–908, July 2006.
[36] E. J. Keogh, X. Xi, L. Wei, and C. Ratanamahatana, "The UCR time series classification-clustering datasets," 2006, http://wwwcs.ucr.edu/ eamonn/time_series_data/.
[37] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
[38] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[39] 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)*, Apr. 2004, pp. 11–22.
[40] P.-F. Marteau and S. Gibet, "Down-Sampling coupled to Elastic Kernel Machines for Efficient Recognition of Isolated Gestures," in *Proceedings of the 2014 22th International Conference on Pattern Recognition*, IAPR, Ed. Stockholm, Sweeden: IEEE, Aug. 2014, p. to appear.
[41] C. Cortes, P. Haffner, and M. Mohri, "Positive definite rational kernels," in *Learning Theory and Kernel Machines*, ser. Lecture Notes in Computer Science, B. Schlkopf and M. Warmuth, Eds. Springer Berlin Heidelberg, 2003, vol. 2777, pp. 41–56.