A Kernel for Time Series Classification: Application to Atmospheric Pollutants

Marta Arias[1], Alicia Troncoso[2], and José C. Riquelme[3]

[1] Universitat Politècnica de Catalunya, Catalunya, Spain
marias@lsi.upc.edu
[2] Pablo de Olavide University, Seville, Spain
ali@upo.es
[3] University of Seville, Seville, Spain
riquelme@us.es

Abstract. In this paper a kernel for time-series data is presented. The main idea of the kernel is that it is designed to recognize as similar time series that may be slightly shifted with one another. Namely, it tries to focus on the shape of the time-series and ignores the fact that the series may not be perfectly aligned. The proposed kernel has been validated on several datasets based on the UCR time-series repository [1]. A comparison with the well-known Dynamic Time Warping (DTW) distance and Euclidean distance shows that the proposed kernel outperforms the Euclidean distance and is competitive with respect to the DTW distance while having a much lower computational cost.

1 Introduction

Time-series analysis is an important problem with application in domains as diverse as engineering, medicine, astronomy or finance [2,3]. In particular, the problem of time-series classification is attracting a lot of attention among researchers [4]. Among the most successful and popular methods for classification are kernel-based methods such as support vector machines. Despite their popularity, there seem to be only a handful of kernels designed for time-series. This paper tries to fill this void, and proposes a kernel exclusively designed for time-series. Moreover, using a standard trick, we are able to convert our kernel into a distance metric for time-series, therefore allowing us to use our kernel in distance-based algorithms as well.

A crucial aspect when dealing with time-series is to find a good measure, either a kernel similarity or a distance, that captures the essence of the time-series according to the domain of application. For example, Euclidean distance between time-series is commonly used due to its computational efficiency; however, it is very brittle and small shifts in of one time-series can result in huge changes in the Euclidean distance. Therefore, more sophisticated distances have been devised designed to be more robust to small fluctuations of the input time-series. Notably, Dynamic Time Warping (DTW) is held as the state-of-the-art method for comparing time-series. Unfortunately, computing the DTW distance is prohibitely costly for many practical applications. Therefore, researchers are coming up with distances for time-series that approach the DTW at lower computational costs [5,4,6]. In a sense, the kernel-derived distance that is proposed here tries to fix the brittleness of Euclidean distance without incurring in the high computational costs of DTW. At a high level, our proposed distance is a combination of the Euclidean distances obtained by using several smoothed versions or the original time-series.

Many other distances have been proposed depending on the invariants required by the domain. For example, [7] define a distance between two time-series representing the convexities/concavities of two shape contours. In [8] the authors modify the Euclidean distance with a correction factor based on the complexity of the input time-series.

It is known that the DTW distance is not a distance in a strict sense as it does not fulfill the triangular inequality and, therefore, it can not be used to define a positive definite kernel. A more general theory of learning instead of positive semi-definite kernels and the relationship between good kernels and similarity functions is presented in [9]. In [10] a new kernel is defined by global alignments from the DTW distance. In particular, the kernel is defined as the sum of the exponential function of the distances for all possible alignments. However, this kernel has a high computational cost and constraints on alignments, similar to that of [5], are presented to speed-up the computation in [11]. The same authors present in [12] a kernel based on the idea that similar time series should be fit well by the same models. They use autoregressive models and thus the name of autoregressive kernel. A kernel for periodic time-series arising in the field of astronomy is presented in [13]. This kernel is similar to a global alignment kernel as it consists in the sum of the exponential function of the inner products for all possible shifting of a time series instead of computing the best alignment. Another kernel for time-series is proposed in [14]. In particular, the time series are represented with a summarizing smooth curve in a Hilbert space and the learning method of the kernel is based on Gaussian processes.

The paper is structured as follows. Section 2 describes our time-series kernel and its corresponding derived distance. Section 3 presents an empirical comparison using 20 different datasets. Finally, Section 5 concludes with a summary of our main contributions and possible directions of future work.

2 Kernel Description

This Section presents the notation used in this paper and also provides the definitions underlying the proposed kernel.

Definition 1 (Time-series). *A time-series X is a set of temporally sorted real-valued data. In this work, $X = \{x_1, ..., x_N\}$, where N is the length of the time-series.*

Definition 2 (Subsequence time-series). *A subsequence of length k of a time-series $X = \{x_1, ..., x_N\}$ is a time-series $X_j = \{x_j, x_{j+1}, ..., x_{j+k-1}\}$ for $1 \leq j \leq N - k + 1$.*

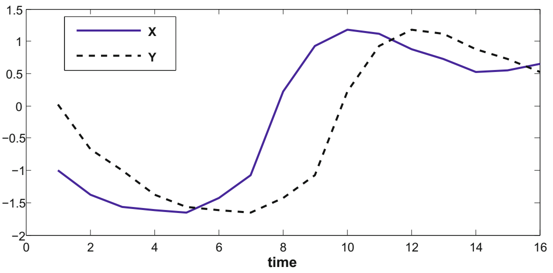

Fig. 1. Example of two shifted time-series

Definition 3 ($\{k, j\}$–Order partial sum). *A $\{k, j\}$–order partial sum of a time-series X, $s_{k,j}^X$, is the sum of the values of the X_j subsequence time-series of length k. That is:*

$$s_{k,j}^X = x_j + x_{j+1} + \cdots + x_{j+k-1}.$$

Definition 4 (k–Order partial sum time-series). *A k–order partial sum time-series is a time-series S_k^X whose values are $s_{k,j}^X$ for $1 \leq j \leq N - k + 1$, that is, the sum of all the values of the subsequences of length k of the time-series X.*

$$S_k^X = \{s_{k,1}^X, s_{k,2}^X, ..., s_{k,N-k+1}^X\}.$$

For example, the $\{k, j\}$–order partial sums and the k–order partial sum time-series for the $X = \{3, 2, 4, 1\}$ time-series are:

$$\begin{array}{ll}
s_{2,1}^X = 3 + 2 = 5 & \\
s_{2,2}^X = 2 + 4 = 6 & S_2^X = \{5,6,5\} \\
s_{2,3}^X = 4 + 1 = 5 & \\
s_{3,1}^X = 3 + 2 + 4 = 9 & S_3^X = \{9,7\} \\
s_{3,2}^X = 2 + 4 + 1 = 7 & \\
s_{4,1}^X = 3 + 2 + 4 + 1 = 10 & S_4^X = \{10\}
\end{array}$$

2.1 Motivation

The main motivation in the definition of the kernel proposed here is to obtain a similarity measure for time-series that yields high values when two time-series X and Y have the same shape but may be shifted of one another. Notice that the Euclidean distance does not take this into account as illustrated in Figure 1. It can be observed that both time-series are very similar (the Y time-series is obtained by shifting X). However, the Euclidean distance between the two time-series is very high. As a consequence, bad results could be obtained were the Euclidean distance to be used in distance-based classification algorithms, for instance. The purpose of this work is to propose a kernel that yields high similarity for time-series that have similar shapes.

The kernel proposed here is obtained by adding the inner products of partial sum time-series for all orders. It is not necessary to discover the best alignment between two time-series, in contrast with the DTW distance, as all partial sums will be included in the kernel definition.

2.2 Definition of the Kernel

Let X and Y be two time-series of length N. Let U^X and U^Y be two upper triangular matrices defined as:

$$\begin{aligned} U^X &= [U_1^X, ..., U_N^X] \\ U^Y &= [U_1^Y, ..., U_N^Y] \end{aligned}$$

where U_i^X and U_i^Y are the i-th rows of the matrices U^X and U^Y, respectively, which are defined by:

$$U_{ij}^X = \begin{cases} s_{i,j}^X & \text{if } 1 \leq j \leq N - i + 1 \\ 0 & \text{if } j > N - i + 1 \end{cases} \tag{1}$$

$$U_{ij}^Y = \begin{cases} s_{i,j}^Y & \text{if } 1 \leq j \leq N - i + 1 \\ 0 & \text{if } j > N - i + 1 \end{cases} \tag{2}$$

Finally, the kernel is defined as the sum of the scalar products among the rows of the U^X and U^Y matrices. That is,

$$Kernel(X, Y) = \sum_{i=1}^{N} < U_i^X, U_i^Y > \tag{3}$$

where U^X and U^Y are defined by Equations (1) and (2) and $< \cdot, \cdot >$ is the scalar product of two vectors in $\mathbb{R}^N$. It is obvious that the function defined by Equation (3) is indeed a kernel as it can be represented by a inner product in the high–dimensional feature space $\phi(\cdot)$ defined as follows:

$$Kernel(X, Y) =< \phi(X), \phi(Y) > \tag{4}$$

where

$$\begin{aligned} \phi : \mathbb{R}^N &\longrightarrow \mathbb{R}^{N^2} \\ X &\longrightarrow \phi(X) = (U_1^X, ..., U_N^X) \end{aligned}$$

Next, we show an illustrative example for the time-series $X = \{3, 2, 4, 1\}$ and $Y = \{1, -1, 0, 2\}$. Firstly, the U^X and U^Y matrices comprising the partial sums of the X and Y time-series have to be computed. The 2-order partial sums for X and Y are $S_2^X = \{5, 6, 5\}$ and $S_2^Y = \{0, -1, 2\}$, respectively. Analogously, the 3 and 4 order partial sums are $S_3^X = \{9, 7\}$, $S_3^Y = \{0, 1\}$, $S_4^X = \{10\}$ and $S_4^Y = \{2\}$. Therefore, the matrices are:

$$U^X = \begin{bmatrix} 3 & 2 & 4 & 1 \\ 5 & 6 & 5 & 0 \\ 9 & 7 & 0 & 0 \\ 10 & 0 & 0 & 0 \end{bmatrix} \quad U^Y = \begin{bmatrix} 1 & -1 & 0 & 2 \\ 0 & -1 & 2 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$

The second step consists in calculating the scalar products of the rows of the U^X matrix and the corresponding rows of the U^Y matrix. That is,

$$
\begin{aligned}
&< U_1^X, U_1^Y > = 3 \cdot 1 + 2 \cdot (-1) + 4 \cdot 0 + 1 \cdot 2 = 3 \\
&< U_2^X, U_2^Y > = 5 \cdot 0 + 6 \cdot (-1) + 5 \cdot 2 = 4 \\
&< U_3^X, U_3^Y > = 9 \cdot 0 + 7 \cdot 1 = 7 \\
&< U_4^X, U_4^Y > = 10 \cdot 2 = 20
\end{aligned}
$$

where U_i^X and U_i^Y are the i rows of the U^X and U^Y matrices, respectively.

Finally, the kernel is defined as the sum of the above-mentioned scalar products. Therefore,

$$Kernel(X, Y) = (3 + 4 + 7 + 20) = 34.$$

It should be noted that a distance metric can be obtained from any positive definite kernel Ker using the standard transformation described in the following equation [15]:

$$d(u, v) = Ker(u, u) + Ker(v, v) - 2 \cdot Ker(u, v).$$

Therefore, when this work refers to the proposed kernel as a distance it really means the derived distance from the kernel.

3 Results

This section presents the results obtained by the application of the proposed kernel to the classification of multi–class time series. Section 3.1 provides a detailed description of all datasets used in the experiments. In Section 3.2 the kernel has been applied to twenty time-series to validate its potential for separating classes in time-series. Finally, a real-world dataset composed by ozone time series is considered in Section 3.3.

3.1 Description of Datasets

The new kernel has been initially tested on several datasets from the UCR time-series repository [1]. Time series lengths in our datasets range from 60 to 637, with the average and the median being 282.1 and 272.5, respectively. Computation times are highly sensitive to the time-series length, especially for the DTW algorithm, which is quadratic in this parameter. Relevant information about these datasets is summarized in Table 1.

Table 1. Datasets from UCR time-series Repository [1]

Dataset	Num. Instances	Num. Classes	Length of Series	Dataset	Num. Instances	Num. Classes	Length of Series
50Words	450	50	270	Lighting-2	20	2	637
Adiac	296	37	176	Lighting-7	63	7	319
Beef	45	5	470	OSU Leaf	54	6	427
CBF	21	3	128	OliveOil	40	4	570
Coffee	18	2	286	Swedish Leaf	105	15	128
ECG	14	2	96	Trace	36	4	275
Fish	63	7	463	Two Patterns	28	4	128
Face (All)	112	14	131	Synthetic Control	36	6	60
Face (Four)	36	4	350	Wafer	16	2	152
Gun-Point	16	2	150	Yoga	18	2	426

3.2 Validation

A statistic based on pair-wise distances has been developed to show how well the proposed kernel is able to separate classes in time-series.

Let D be a labeled dataset of M time-series of the same length N. Let $c(X)$ be the class of the time-series $X \in D$. Then, the SM separation measure is defined as follows,

$$SM = \frac{INTRA - INTER}{MAX}$$

where $INTRA$ and $INTER$ are the average pair-wise distance of time series belonging to the same and to different classes, respectively, and MAX is the maximum pair-wise distance over the whole dataset. Namely, let $A = \{(X,Y)|X,Y \in D, c(X) = c(Y)\}$ and $B = \{(X,Y)|X,Y \in D, c(X) \neq c(Y)\}$. That is, A is the set of pairs of time series that belong to the same class, and B is the set of pairs of time-series that belong to different classes. Then,

$$INTRA = \frac{1}{|A|} \sum_{(X,Y)\in A} d(X,Y)$$

$$INTER = \frac{1}{|B|} \sum_{(X,Y)\in B} d(X,Y)$$

$$MAX = \underset{X,Y\in D}{max}\, d(X,Y)$$

where d is any distance defined in $\mathbb{R}^N$ x $\mathbb{R}^N$ and N is the length of the time-series in D.

In a sense, the SM measure is designed to show how well the proposed distance separates instances in different classes as opposed to instances within the same class. Since we are taking averages, it is a measure of the *global* separability ability of the distance. This distance is reminiscent to the cost function used in the problem of correlation clustering in weighted graphs [16,17], if we were to cluster all instances using their class.

Table 2 presents a comparison of the separation statistic and computation time of the following distances: the Euclidean distance, the one derived from the kernel proposed here (which we call Kernel-based distance), and the DTW distance. The comparison is over the 20 datasets from the UCR repository [1]. The distance that better separates the existing classes for each dataset is marked in bold style. It can be seen that the average of the separation measure for the proposed kernel is better than that of the Euclidean distance and similar to that of the DTW distance. When looking at the columns for computation times, it is very clear that the Euclidean distance is by far the fastest one to compute, followed by our proposed distance using (roughly) an order or magnitude extra CPU time. The DTW distance is by far the slowest, needing two more orders or magnitude than our kernel-based distance.

Table 3 further summarizes Table 2. On the table on the left, the reader can observe that the behavior of the kernel-based distance is better than that of DTW on average (1.65 versus 1.85), and both outperform the Euclidean distance (1.65 and 1.85 versus 2.40). The table on the right shows the wins matrix for pairs of distances over the 20 datasets. That is, in how many datasets a distance separates better than another distance.

Table 2. Separation measure among classes and computing times

	Separation Measure			CPU Times (in s.)		
Dataset	Eucl	Kernel	DTW	Eucl	Kernel	DTW
50Words	0.155	0.196	**0.498**	177.4	3653.3	176629.5
Adiac	-0.042	-0.040	**-0.027**	83.8	387.5	32702.9
Beef	0.696	**0.894**	0.557	1.8	93.7	5505.9
CBF	0.150	0.311	**0.557**	0.5	4.9	94.2
Coffee	-0.015	**0.111**	-0.015	0.3	5.8	316.9
ECG	0.045	0.054	**0.126**	0.2	3.8	22.8
FISH	**-0.004**	-0.008	-0.018	10.3	46.4	2690.4
Face (All)	0.007	0.110	**0.387**	1.1	46.7	1928.6
Face (Four)	-0.005	**0.016**	0.011	3.4	131.1	10576.4
Gun-Point	0.136	0.113	**0.345**	0.3	1.2	83.3
Lighting-2	0.126	0.152	**0.263**	0.4	29.0	1964.2
Lighting-7	0.137	0.268	**0.279**	3.6	67.1	4767.3
OSU Leaf	0.229	**0.346**	0.098	1.4	94.0	6396.3
OliveOil	-0.063	-0.041	**-0.027**	2.5	100.8	6459.6
Swedish Leaf	0.104	**0.144**	0.048	9.3	39.2	2286.0
Trace	0.300	**0.303**	0.091	1.2	3.1	65.1
Two Patterns	0.113	0.165	**0.585**	1.1	19.2	1195.1
Synthetic Control	0.103	0.293	**0.580**	0.7	3.9	174.5
Wafer	0.126	**0.172**	0.015	0.3	1.0	72.9
Yoga	-0.000	**0.073**	-0.005	0.3	11.0	724.1
Average	**0.115**	**0.182**	**0.202**	**15**	**237.13**	**12732.8**

Table 3. Left: comparison of the number of times each distance achieves the first, second and third positions over all datasets and average rank. Right: Win matrix for pairs of distances, it should be read as follows: if row i and column j contains number m, then distance i has beaten distance j a total of m times. For example, the Euclidean distance beats the Kernel-based distance in 2 datasets, and beats the DTW distance in 8 datasets.

Distance	#1st	#2nd	#3rd	Avg. Rank
Euclidean	1	7	11	2.40
Kernel	8	11	1	1.65
DTW	11	1	8	1.85

Distance	Euclidean	Kernel	DTW
Euclidean	–	2	8
Kernel	18	–	9
DTW	12	11	–

4 A Real Application: Classification of Ozone Concentration in Atmosphere

Finally, an environmental application related to atmospheric pollutants such as the tropospheric ozone is presented. The pattern recognition in ozone time data is an important task as it is neccessary activate environmental politics and alert protocols by the government when the ozone reaches high ozone concentration levels in atmosphere. Ozone time series have been retrieved from a meteorological station placed in the outskirts of Seville city (Spain), providing 312 times series composed of 168 hourly records each one. The dataset is classified into two classes corresponding to high and low ozone level periods (165 and 147 time series, respectively). The time series data have been split in training set (218 time series) and test set (94 time series) preserving the proportion between the two existing classes.

We have used the well-known nearest neighbor method (1-NN) with the three competing distances to classify the ozone time series into weeks of high or low ozone concentration. Table 4 shows the error in percentage and the time in seconds obtained from the application of the 1-NN method to classify the test set when using several distances. It can be observed that the kernel-based distance presents better results in both error and CPU time.

Table 4. Percentage of error and time in seconds required to classify the test set

Distance	Error	Time
Euclidean	9.5%	0.5
Kernel-based	**4.2%**	23.3
DTW	6.3%	3692.9

5 Conclusions and Future Work

In this paper we have presented a kernel for time-series data and its associated distance metric. Initial experiments show promise in detecting similarity between time-series. The proposed kernel has been compared to the Euclidean distance as a reference distance and the DTW distance as one of the most competitive distances that exist in the literature. The kernel is shown to efficiently separate different time-series classes, and also, its application to real-world data has been successful. In particular, it achieves low error in the classification of the ozone atmosphere concentration. This paper is an initial step in the study of this kernel and its possible variants. Further experimentation including comparison to other state-of-the-art kernels are underway in the context of classification with kernel-based methods. In the future, we plan to generalize our kernel to time-series that differ in length. We would also like to adapt our ideas so that they can be used in a *streaming* setting where time-series keep growing unboundedly.

References

1. Keogh, E.: UCR time series repository (2011), `http://www.cs.ucr.edu/~eamonn/`
2. Sedano, J., Curiel, L., Corchado, E., de la Cal, E., Villar, J.R.: A soft computing method for detecting lifetime building thermal insulation failures. Integr. Comput. Aided Eng. 17(2), 103–115 (2010)
3. Corchado, E., Arroyo, A., Tricio, V.: Soft computing models to identify typical meteorological days. Logic Journal of The Igpl / Bulletin of The Igpl 19, 373–383 (2011)
4. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.: Fast time series classification using numerosity reduction. In: Proceedings of the 23rd International Conference on Machine Learning, vol. 1040. ACM (2006)
5. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 26, 43–49 (1978)
6. Marteau, P.F., Ménier, G.: Speeding up simplification of polygonal curves using nested approximations. Pattern Anal. Appl. 12(4), 367–375 (2009)
7. Adamek, T., O'Connor, N.E.: A multiscale representation method for nonrigid shapes with a single closed contour. IEEE Transactions on Circuits and Systems for Video Technology 14(5), 742–752 (2004)
8. Gustavo, E.A.P.A., Batista, X.W., Keogh, E.J.: A complexity-invariant distance measure for time series. In: SIAM International Conference on Data Mining (2011)
9. Balcan, M., Blum, A., Srebro, N.: A theory of learning with similarity functions. Machine Learning 72(1), 89–112 (2008)
10. Cuturi, M., Vert, J.P., Birkenes, O., Matsui, T.: A kernel for time series based on global alignments. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007, vol. 2, pp. II–413–II–416 (2007)
11. Cuturi, M.: Fast global alignement kernels. In: Internacional Conference on Machine Learning (2011)
12. Cuturi, M., Doucet, A.: Autoregressive kernels for time series (2011) arXiv:1101.0673

13. Wachman, G., Khardon, R., Protopapas, P., Alcock, C.R.: Kernels for periodic time series arising in astronomy. In: European Conference on Machine Learning (2009)
14. Lu, Z., Leen, T., Huang, Y., Erdogmus, D.: A reproducing kernel Hilbert space framework for pairwise time series distances. In: Proceedings of the 25th International Conference on Machine Learning, pp. 624–631. ACM (2008)
15. Scholkopf, B.: The kernel trick for distances. In: Proceedings of the Conference on Neural Information Processing Systems (NIPS), pp. 301–307 (2000)
16. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning 56(1-3), 89–113 (2004)
17. Bonchi, F., Gionis, A., Ukkonen, A.: Overlapping correlation clustering. In: ICDM, pp. 51–60 (2011)