



HAL
open science

Interpretable time series kernel analytics by pre-image estimation

Thi Phuong Thao Tran, Ahlame Douzal-Chouakria, Saeed Varasteh Yazdi,
Paul Honeine, Patrick Gallinari

► **To cite this version:**

Thi Phuong Thao Tran, Ahlame Douzal-Chouakria, Saeed Varasteh Yazdi, Paul Honeine, Patrick Gallinari. Interpretable time series kernel analytics by pre-image estimation. *Artificial Intelligence*, 2020, 286, pp.103342. 10.1016/j.artint.2020.103342 . hal-03088295

HAL Id: hal-03088295

<https://normandie-univ.hal.science/hal-03088295>

Submitted on 26 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 Interpretable time series kernel analytics by pre-image
2 estimation

Thao Tran Thi Phuong, Ahlame Douzal, Saeed Varasteh Yazdi,
Paul Honeine, and Patrick Gallinari

3 **Abstract**

Kernel methods are known to be effective to analyse complex objects by implicitly embedding them into some feature space. To interpret and analyse the obtained results, it is often required to restore in the input space the results obtained in the feature space, by using pre-image estimation methods. This work proposes a new closed-form pre-image estimation method for time series kernel analytics that consists of two steps. In the first step, a time warp function, driven by distance constraints in the feature space, is defined to embed time series in a metric space where analytics can be performed conveniently. In the second step, the time series pre-image estimation is cast as learning a linear (or a nonlinear) transformation that ensures a local isometry between the time series embedding space and the feature space. The proposed method is compared to the state of the art through three major tasks that require pre-image estimation: 1) time series averaging, 2) time series reconstruction and denoising and 3) time series representation learning. The extensive experiments conducted on 33 publicly-available datasets show the benefits of the pre-image estimation for time series kernel analytics.

4 **1. Introduction**

5 Kernel methods [24] are well known to be effective in dealing with nonlin-
6 ear machine learning problems in general, and are often required for machine
7 learning tasks on complex data as sequences, time series or graphs. The main
8 idea behind kernel machines is to map the data from the input space to a
9 higher dimension feature space (i.e., kernel space) via a nonlinear map, where
10 the mapped data can be then analysed by linear models. While the mapping
11 from input space to the feature space is of primary importance in kernel meth-
12 ods, the reverse mapping of the obtained results from the feature space back to
13 the input space (called the pre-image problem) is also very useful. Estimating
14 pre-images is important in several contexts for interpretation and analysis pur-
15 poses. From the beginning, it has been often considered to estimate denoised
16 and compressed results of a kernel Principal Component Analysis (PCA). Other
17 tasks are of great interest, since the pre-image estimation allows, for instance,
18 to obtain the reverse mapping of the centroids of a kernel clustering, and of the
19 atoms as well as of the sparse representations in kernel dictionary learning.
20

21 In view of the importance of the pre-image estimation issue and of its bene-
22 fits in machine learning, several major propositions have been developed. First,
23 in Mika et al. [23], the problem is formalised as a nonlinear optimisation prob-
24 lem and, for the particular case of the Gaussian kernel, a fixed-point iterative
25 solution to estimate the reverse mapping is proposed. To avoid numerical insta-
26 bilities of the latter approach, in Kwok et al.[18], the relationship between the
27 distances in feature and input spaces is established for standard kernels, and
28 then used to approximate pre-images by multidimensional scaling. In Bakir et
29 al. [3], the pre-image estimation problem is cast as a regression problem be-
30 tween the input and the mapped data, the learned regression model is then
31 used to predict pre-images. Honeine and Richard proposed in [15] an approach
32 where the main idea is to estimate, from the mapped data, a coordinate system
33 that ensures an isometry with the input space; this approach has the advantage
34 to provide a closed-form solution, to be independent of the kernel nature and
35 to involve only linear algebra. More recently, task-specific estimation has been
36 studied, such as the resolution of the pre-image problem for nonnegative matrix
37 factorisation in [32].

38 All the proposed methods for pre-image estimation are either based on op-
39 timisation schema, such as gradient descent or fixed-point iterative solution, or
40 based on ideas borrowed from dimensionality reduction methods. In particu-
41 lar, these methods were developed for Euclidean input spaces, as derivations
42 are straightforward owing to linear algebra (see [16] for a survey on the resolu-
43 tion of the pre-image problems in machine learning). A major challenge arises
44 when dealing with non-Euclidean input spaces, that describe complex data as
45 sequences, time series, manifolds or graphs. Some recent works address that
46 pre-image problem on that challenging data. For instance, in Cloninger et al.
47 [9] the pre-image problem is addressed for Laplacian Eigenmaps under L_1 reg-
48 ularisation and in Bianchi et al. [6] an encoder-decoder is used to learn a latent
49 representation driven by the kernel similarities, where the pre-image estimation
50 is explicitly given by the decoder side.

51 For temporal data, while kernel machinery has been increasingly investigated
52 with success for time series analytics [13, 27, 31, 30, 29, 19], the pre-image
53 problem for temporal data remains in its infancy. In addition, time series data,
54 that involve varying delays and/or different lengths, are naturally lying in a
55 non-Euclidean input space, making the above existing pre-image approaches for
56 static data inapplicable. This work aims to fill this gap, by proposing a pre-
57 image estimation approach for time series kernel analytics, that consists of two
58 steps. In the first step, a time warp function, driven by distance constraints
59 in the feature space, is defined to embed time series in a metric space where
60 analytics can be performed conveniently. In the second step, the time series
61 pre-image estimation is cast as learning a linear or a nonlinear transformation
62 that ensures a local isometry between the time series embedding space and the
63 feature space. The relevance of the proposed method is studied through three
64 major tasks that require pre-image estimation: 1) time series averaging, 2) time
65 series reconstruction and denoising under kernel PCA, and 3) time series repre-
66 sentation learning and dictionary learning under kernel k -SVD. The benefits of

67 the proposed method are assessed through extensive experiments conducted on
 68 33 publicly-available time series datasets, including univariate and multivariate
 69 time series that may include varying delays and be of the different lengths.

70

71 The main contributions of this paper are:

- 72 1. We propose a time warp function, driven by distance constraints in the
 73 feature space, that embeds time series into an Euclidean space
- 74 2. We cast the time series pre-image estimation approach as learning a linear
 75 or nonlinear transformations in the feature space
- 76 3. We propose a closed-form solution for pre-image estimation by preserving
 77 a local isometry between the temporal embedded space and the feature
 78 space
- 79 4. We conduct wide experiments to compare the proposed approach to the
 80 major alternative pre-image estimation methods under three crucial tasks:
 81 1) time series averaging, 2) time series reconstruction and denoising and
 82 3) time series representation and dictionary learning.

83 The remainder of the paper is organised as follows. Section 2 gives a brief
 84 introduction to kernel PCA and kernel k -SVD and Section 3 presents the major
 85 pre-image estimation methods. In Section 4, we formalise the pre-image esti-
 86 mation problem for time series and develop the proposed solution as well as the
 87 corresponding algorithm. We detail the experiments conducted and discuss the
 88 obtained results in Section 5.

89 2. Kernel PCA and kernel k -SVD

90 Kernel methods [24] rely on embedding samples $\mathbf{x}_i \in \mathbb{R}^d$ with $\Phi(\mathbf{x}_i)$ into a
 91 feature space \mathcal{H} , a Hilbert space of arbitrary large and possibly infinite dimen-
 92 sion. The map function Φ needs not to be explicitly defined, since computations
 93 conducted in \mathcal{H} can be carried out by a kernel function that measures the inner
 94 product in that space, namely $\kappa(\mathbf{x}_i, \mathbf{x}_{i'}) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_{i'}) \rangle$ for all $\mathbf{x}_i, \mathbf{x}_{i'}$. Given
 95 a set of input samples $\{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^d$, let K be the Gram matrix related
 96 to the kernel, namely $K_{ii'} = \kappa(\mathbf{x}_i, \mathbf{x}_{i'})$. Let $X = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ be the
 97 description of N input samples $\mathbf{x}_i \in \mathbb{R}^d$ and, with some abuse of notation, let
 98 $\Phi(X)$ be the matrix of entries $\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_N)$.

99

100 In the following, we describe two well-known kernel methods, kernel PCA
 101 and kernel k -SVD, as nonlinear extensions of the well-known PCA and k -SVD.
 102 While both methods estimate a linear combination for optimal reconstruction
 103 of the input samples, the former forces the orthogonality of the atoms that leads
 104 to an orthonormal basis, and the latter forces the sparsity while relaxing the
 105 orthogonality condition.

106 2.1. Kernel PCA

107 Kernel PCA extends standard PCA to find principal components that are
 108 nonlinearly related to the input variables. For that, the principal components

109 are rather determined in the feature space. For the sake of clarity, we assume
 110 for now that we are dealing with centred mapped data, namely $\sum_{i=1}^N \Phi(\mathbf{x}_i) = \mathbf{0}$.
 111 The covariance matrix in the feature space takes then the form

$$C = \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T. \quad (1)$$

112 Similarly to standard PCA, the objective comes to find the pairs of eigenvalue
 113 $\lambda_j \geq 0$ and corresponding eigenvector $\mathbf{u}_j \in \mathcal{H} \setminus \mathbf{0}$ that satisfy

$$\lambda_j \mathbf{u}_j = C \mathbf{u}_j, \quad (2)$$

114 namely for each $\Phi(\mathbf{x}_i)$

$$\lambda_j \langle \mathbf{u}_j, \Phi(\mathbf{x}_i) \rangle = \langle C \mathbf{u}_j, \Phi(\mathbf{x}_i) \rangle. \quad (3)$$

115 As each eigenvector \mathbf{u}_j lies in the span of $\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_N)$, there exist coeffi-
 116 cients $\alpha_{ij}, \dots, \alpha_{Nj}$ such that

$$\mathbf{u}_j = \sum_{i=1}^N \alpha_{ij} \Phi(\mathbf{x}_i). \quad (4)$$

117 From Eq. (3) and Eq. (4), and by simple developments, the problem in Eq. (2)
 118 remains to find the solution of the eigendecomposition problem:

$$\lambda_j \boldsymbol{\alpha}_j = \frac{1}{N} K \boldsymbol{\alpha}_j. \quad (5)$$

119 Let $\lambda_1 \geq \dots \geq \lambda_p$ be the p highest non-zero eigenvalues of $\frac{1}{N} K$ and $\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_p$
 120 their corresponding eigenvectors. The principal components in the feature space
 121 are then given by computing the projections $P_j(\Phi(\mathbf{x}))$ of the sample \mathbf{x} onto the
 122 eigenvector $\mathbf{u}_j = \Phi(X) \boldsymbol{\alpha}_j$:

$$P_j(\Phi(\mathbf{x})) = \langle \mathbf{u}_j, \Phi(\mathbf{x}) \rangle = \sum_{i=1}^N \alpha_{ij} \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle = \mathbf{k}_x \boldsymbol{\alpha}_j \quad (6)$$

123 with $\mathbf{k}_x = [\kappa(\mathbf{x}_1, \mathbf{x}), \dots, \kappa(\mathbf{x}_N, \mathbf{x})]$. By denoting $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_p]$, the description
 124 $P(\Phi(\mathbf{x}))$ of $\Phi(\mathbf{x})$ into the subspace of the p first principal components is then

$$P(\Phi(\mathbf{x})) = (\mathbf{k}_x \boldsymbol{\alpha})^T \quad (7)$$

125 Two considerations should be taken. First, the eigenvector expansion coeffi-
 126 cients $\boldsymbol{\alpha}_j$ should be normalised by requiring $\lambda_j \langle \boldsymbol{\alpha}_j, \boldsymbol{\alpha}_j \rangle = 1$. Secondly, as $\Phi(\mathbf{x}_i)$
 127 are assumed centred, both the Gram matrix K in Eq. (5) and \mathbf{k}_x used in Eq.
 128 (6) and Eq. (7) need to be substituted with their centred counterparts, namely

$$\tilde{K}_{ij} = (K - \mathbf{1}_N K - K \mathbf{1}_N + \mathbf{1}_N K \mathbf{1}_N)_{ij} \quad (8)$$

$$\tilde{\mathbf{k}}_x = \left(\mathbf{k}_x - \frac{1}{N} \mathbf{1}_N^T K \right) (I_N - \mathbf{1}_N) \quad (9)$$

129 with $(\mathbf{1}_N)_{ij} = 1/N$ for all i, j , I_N the identity matrix and $\mathbf{1}_N \in \mathbb{R}^N$ the unit
 130 vector.

131 *2.2. Kernel k -SVD*

132 Sparse coding and dictionary learning become popular methods for a vari-
 133 ety of tasks as feature extraction, reconstruction, denoising, compressed sensing
 134 and classification [17, 4, 5]. k -SVD [1] is among the most-known and tractable
 135 dictionary learning approach to learn a dictionary and to sparse represent the
 136 input samples as a linear combination of the dictionary atoms. When dealing
 137 with complex data, kernel k -SVD may be required to learn, in the feature
 138 space, the dictionary and the sparse representations of the mapped samples as
 139 a nonlinear combination of the dictionary atoms [28]. Let us introduce a brief
 140 description of kernel k -SVD.

141 Let $D = [\mathbf{d}_1, \dots, \mathbf{d}_L] \in \mathbb{R}^{d \times L}$ be the dictionary composed of L atoms $\mathbf{d}_j \in \mathbb{R}^d$.
 142 The embedded dictionary $\Phi(D) = \Phi(X)\mathcal{B}$ is defined as a linear representation
 143 of $\Phi(X)$, since the atoms lie in the subspace spanned by the $\Phi(X)$. The kernel
 144 dictionary learning problem takes the form
 145

$$\min_{\mathcal{B}, \mathcal{A}} \|\Phi(X) - \Phi(X)\mathcal{B}\mathcal{A}\|_{\mathcal{F}}^2 \quad (10)$$

$$\text{s.t. } \|\mathbf{a}_i\|_0 \leq \tau \quad \forall i = 1, \dots, N, \quad (11)$$

146 where $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm¹, the matrix $\mathcal{B} = [\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_L] \in \mathbb{R}^{N \times L}$
 147 gives the representation of the embedded atoms into the base $\Phi(X)$ and $\mathcal{A} =$
 148 $[\mathbf{a}_1, \dots, \mathbf{a}_N] \in \mathbb{R}^{L \times N}$ gives the sparse representations of $\Phi(X)$, with the sparsity
 149 level τ imposed by the above constraint.

150 The kernel k -SVD algorithm iteratively cycles between two stages. In the
 151 first stage, the dictionary is assumed fixed with \mathcal{B} known and a kernel orthog-
 152 onal matching pursuit (OMP) technique [20] is deployed to estimate \mathcal{A} . As in
 153 standard OMP, given a sample \mathbf{x} , we select the first atom that best reconstructs
 154 $\phi(\mathbf{x})$; namely we select the j -th atom that maximises
 155

$$(\mathbf{k}_{\mathbf{x}} - \mathbf{a}_{\mathbf{x}}^T \mathcal{B}^T K) \boldsymbol{\beta}_j. \quad (12)$$

156 The sparse codes are then updated by the projections onto the subspace of the
 157 yet selected atoms

$$\mathbf{a}_{\mathbf{x}} = (\mathcal{B}_{\Omega}^T K \mathcal{B}_{\Omega})^{-1} (\mathbf{k}_{\mathbf{x}} \mathcal{B}_{\Omega})^T, \quad (13)$$

158 where \mathcal{B}_{Ω}^T is the submatrix of \mathcal{B} limited to the yet selected atoms. The proce-
 159 dure is reiterate until the selection of τ atoms.

160 Once the sparse codes \mathcal{A} of the N samples estimated, the second stage of the
 161 kernel k -SVD is performed to update both \mathcal{B} and \mathcal{A} . For that, the reconstruction
 162

¹ $\|M\|_{\mathcal{F}} = \sqrt{\sum_{i=1}^p \sum_{j=1}^q (m_{ij})^2}$ is the Frobenius norm of the matrix $M \in \mathbb{R}^{p \times q}$ of general term m_{ij}

163 error is defined as

$$\min_{\mathcal{B}, \mathcal{A}} \|\Phi(X) - \Phi(X) \sum_{j=1}^L \beta_j \mathbf{a}_j\|_{\mathcal{F}}^2, \quad (14)$$

164 with $\mathbf{a}_j \in \mathbb{R}^N$ referencing the j -th row of \mathcal{A} , namely

$$\min_{\mathcal{B}, \mathcal{A}} \|\Phi(X) E_k - \Phi(X) \beta_k \mathbf{a}_k\|_{\mathcal{F}}^2, \quad (15)$$

165 with $E_k = I_N - \sum_{j \neq k} \beta_j \mathbf{a}_j$. the error of reconstruction matrix when removing
166 the k -th atom. An eigendecomposition is then performed to get

$$(E_k^R)^T K(E_k^R) = V \Sigma V^T, \quad (16)$$

167 where E_k^R is the error of reconstruction restricted to the samples that have
168 involved the k -th atom. The sparse codes β_k and \mathbf{a}_k^R are updated by using the
169 first eigenvector \mathbf{v}_1 with

$$\mathbf{a}_k^R = \sigma_1 \mathbf{v}_1^T \text{ and } \beta_k = \sigma_1^{-1} E_k^R \mathbf{v}_1. \quad (17)$$

170 3. Related works on pre-image estimation

171 From the representer theorem [22] any result $\varphi \in \mathcal{H}$ obtained by some kernel
172 method may be expressed in the form $\varphi = \sum_{i=1}^N \gamma_i \Phi(\mathbf{x}_i)$; that is as a linear
173 combination of the mapped training samples $\{\Phi(\mathbf{x}_i)\}_{i=1}^N$. In general, finding
174 the exact pre-image \mathbf{x} such that $\Phi(\mathbf{x}) = \varphi$ is an ill-posed problem, that is often
175 addressed by providing an approximate solution, namely by estimating \mathbf{x}^* such
176 that $\Phi(\mathbf{x}^*) \approx \varphi$. In the following, we describe three major methods to estimate
177 the pre-image \mathbf{x}^* of a given $\varphi \in \mathcal{H}$.

178 3.1. Pre-image estimation under distance constraints

179 The main idea proposed in Kwok et al. [18] is to use the distances between
180 φ and $\Phi(\mathbf{x}_i)$ and their relation to the distances between the pre-image \mathbf{x}^* and
181 \mathbf{x}_i . The main steps of the proposed approach are:

- 182 1. Let $\tilde{d}^2(\varphi, \Phi(\mathbf{x}_i)) = \langle \varphi, \varphi \rangle - 2\langle \varphi, \Phi(\mathbf{x}_i) \rangle + \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_i) \rangle$ be the square dis-
183 tance between φ and any $\Phi(\mathbf{x}_i)$. In practice, only neighbouring elements
184 are considered. Let $\Phi(\hat{\mathbf{x}}_1), \dots, \Phi(\hat{\mathbf{x}}_n)$ denote the n -th closest elements to φ .
- 185 2. For an isotropic kernel², the relation $d^2(\mathbf{x}_i, \mathbf{x}_j) = g(\tilde{d}^2(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)))$ be-
186 tween the distances in the input and the feature spaces can be established.

²an isotropic kernel is a function of the form $k(\mathbf{x}, \mathbf{y}) = f(\|\mathbf{x} - \mathbf{y}\|)$ that depends only on the norm of the lag vector between two samples.

187 3. A solution is then deployed to determine the pre-image \mathbf{x}^* such that

$$[d^2(\mathbf{x}^*, \hat{\mathbf{x}}_1), \dots, d^2(\mathbf{x}^*, \hat{\mathbf{x}}_n)] = [g(\tilde{d}^2(\varphi, \Phi(\hat{\mathbf{x}}_1))), \dots, g(\tilde{d}^2(\varphi, \Phi(\hat{\mathbf{x}}_n)))] \quad (18)$$

188 For that, an SVD is deployed on the centred version of the submatrix
189 $X_n = [\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_n]$, namely

$$X_n(I_n - \mathbf{1}_n) = U\Lambda V^T, \quad (19)$$

190 where $U = [\mathbf{u}_1, \dots, \mathbf{u}_q]$ is the $d \times q$ matrix of the left-singular vectors.
191 Let $Z = [\mathbf{z}_1, \dots, \mathbf{z}_n] = \Lambda V^T$ be the $q \times n$ matrix giving the projections of
192 $\hat{\mathbf{x}}_i$ on the \mathbf{u}_j 's orthonormal vectors. The pre-image estimation \mathbf{x}^* is then
193 obtained as:

$$\mathbf{x}^* = U\mathbf{z}^* + \frac{1}{n}X_n\mathbf{1}_n, \quad (20)$$

194 where \mathbf{z}^* , the projection of \mathbf{x}^* on the $[\mathbf{u}_1, \dots, \mathbf{u}_q]$ coordinate system, is

$$\mathbf{z}^* = -\frac{1}{2}(ZZ^T)^{-1}Z(\mathbf{d}^2 - \mathbf{d}_0^2), \quad (21)$$

195 with $\mathbf{d}_0^2 = [\|\mathbf{z}_1\|^2, \dots, \|\mathbf{z}_n\|^2]^T$, $\mathbf{d}^2 = [d_1^2, \dots, d_n^2]^T$ and $d_i^2 = g(\tilde{d}^2(\varphi, \Phi(\hat{\mathbf{x}}_i)))$.

196 3.2. Pre-image estimation by isometry preserving

197 To learn the pre-image \mathbf{x}^* of a given $\varphi = \sum_{i=1}^N \gamma_i \Phi(\mathbf{x}_i)$, the proposed
198 approach in [15] proceeds in two steps. First, a coordinate system, spanned by
199 the feature vectors $\{\Phi(\mathbf{x}_i)\}_{i=1}^N$ is learned to ensure an isometry with the input
200 space; subsequently, the coordinate system is used to estimate the pre-image \mathbf{x}^*
201 of φ . These two main steps are summarised in the followings:

202 1. Let $\Psi = \{\psi_1, \dots, \psi_p\}$ ($p \leq N$) be a coordinate system in the feature space,
203 with $\psi_k = \Phi(X)\alpha_k$. The projection of $\Phi(X)$ onto the coordinate system
204 is $P(\Phi(X)) = (KA)^T$, where $A = [\alpha_1, \dots, \alpha_p]$. To estimate the coordinate
205 system that is isometric with the input space, the following optimisation
206 problem is solved

$$\arg \min_A \|X^T X - KAA^T K\|_{\mathcal{F}}^2 + \lambda \|KA\|_{\mathcal{F}}^2, \quad (22)$$

207 where the second term controls through the regularisation parameter λ
208 the smoothness of the solution. The solution of this problem satisfies
209 $AA^T = K^{-1}(X^T X - \lambda K^{-1})K^{-1}$.

210 2. Based on this result, the pre-image estimation \mathbf{x}^* takes the form

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \|X^T \mathbf{x} - (X^T X - \lambda K^{-1})\boldsymbol{\gamma}\|_{\mathcal{F}}^2 \quad (23)$$

211 with $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N)^T$. This problem defines a standard overdetermined
212 equation system ($N \gg d$) that can be resolved as a least-square min-
213 imisation problem (i.e., any technique such as the pseudo-inverse or the
214 eigendecomposition). The pre-image estimation is then:

$$\mathbf{x}^* = (XX^T)^{-1}X(X^T X - \lambda K^{-1})\boldsymbol{\gamma}. \quad (24)$$

215 *3.3. Pre-image estimation by kernel regression*

216 In Bakir et al. [3], the pre-image estimation consists in learning a kernel
 217 regression function that maps all the $\Phi(\mathbf{x}_i)$ in the feature space \mathcal{H} (related to
 218 the kernel κ) to \mathbf{x}_i in the input space \mathbb{R}^d . For that, first a kernel PCA is deployed
 219 to embed $\Phi(X)$ into the subspace spanned by the eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_p$ defined
 220 in (4), with $\mathbf{u}_j = \Phi(X)\boldsymbol{\alpha}_j$. Then, a kernel regression is learned from the set of
 221 the projections onto the kernel PCA subspace and X as

$$\arg \min_B \|X - B\widehat{K}\|_{\mathcal{F}}^2 + \lambda\|B\|_{\mathcal{F}}^2, \quad (25)$$

222 where $B \in \mathbb{R}^{d \times N}$ is the regression coefficient matrix and \widehat{K} is the Gram matrix
 223 with entries $\widehat{K}_{ij} = \widehat{\kappa}(P(\Phi(\mathbf{x}_i)), P(\Phi(\mathbf{x}_j)))$. The solution to the problem (25) is
 224 then

$$B = X\widehat{K}(\widehat{K}^2 + \lambda I_N)^{-1} \quad (26)$$

225 For a given result $\boldsymbol{\varphi} = \Phi(X)\boldsymbol{\gamma} \in \mathcal{H}$, its pre-image \mathbf{x}^* is then estimated as:

$$\mathbf{x}^* = B\widehat{\mathbf{k}}_{P(\boldsymbol{\varphi})}^T, \quad (27)$$

226 with

$$\widehat{\mathbf{k}}_{P(\boldsymbol{\varphi})} = [\widehat{\kappa}(P(\boldsymbol{\varphi}), P(\Phi(\mathbf{x}_1))), \dots, \widehat{\kappa}(P(\boldsymbol{\varphi}), P(\Phi(\mathbf{x}_N)))]], \quad (28)$$

227 where $\widehat{\kappa}(P(\boldsymbol{\varphi}), P(\Phi(\mathbf{x}_i))) = \widehat{\kappa}(\boldsymbol{\alpha}^T K \boldsymbol{\gamma}, \boldsymbol{\alpha}^T \mathbf{k}_{\mathbf{x}_i}^T)$ and $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_p]$.

229 *3.4. Overview*

230 To sum up, the three major methods presented above define three different
 231 approaches for pre-image estimation problem. First of all, all the methods in-
 232 volve only linear algebra and propose solutions that don't suffer from numerical
 233 instabilities. In Kwok et al. [18], the solution is mainly requiring the definition
 234 of a relation between the distances into the input and the kernel feature spaces.
 235 That requirement limite the Kwok et al. [18] approach to linear or isotropic
 236 kernels. Honeine et al. [15] alleviate that point by proposing a closed-form solu-
 237 tion that is applicable to any type of kernels. Furthermore, while in Honeine et
 238 al. [15] the pre-image estimation is obtained by learning a linear transformation
 239 into the feature space that preserves the isometry between the input and the
 240 feature space, in Bakir et al. [3], the pre-image estimation is obtained by using
 241 a non linear kernel regression that predicts the input samples from their images
 242 into the feature space. Finally, while both [15] and [3] proposals involve the
 243 whole training samples for pre-image estimation, Kwok et al. [18] uses only the
 244 samples on the neighborhood of $\boldsymbol{\varphi}$, which offers a significant speed-up; highly
 245 valuable in the case of large scale data.

246 **4. Proposed pre-image estimation for time series kernel analytics**

247 Let $\{\mathbf{x}_i\}_{i=1}^N$ be a set of N time series, where each $\mathbf{x}_i \in \mathbb{R}^{d \times t_i}$ is a mul-
 248 tivariate time series of length t_i that may involve varying delays. Let $\Phi(\mathbf{x}_i)$
 249 be the Φ -mapping of the time series \mathbf{x}_i into the Hilbert space \mathcal{H} related to a
 250 temporal kernel κ that involves dynamic time alignments such as DTAK [25],
 251 KDTW [2] and KGA [10]. Let K be a the corresponding Gram matrix, with
 252 entries $K_{ii'} = \kappa(\mathbf{x}_i, \mathbf{x}_{i'})$. Given $\varphi = \sum_{i=1}^N \gamma_i \Phi(\mathbf{x}_i)$ a result generated in \mathcal{H} ,
 253 the objective is to estimate the time series $\mathbf{x}^* \in \mathbb{R}^{d \times t^*}$ that is the pre-image
 254 of φ . This problem is particularly challenging since, under varying delays, the
 255 time series are not longer lying in a metric space, which makes inapplicable the
 256 related work on pre-image estimation.

257 We tackle this problem in two parts. In the first part, we formalise the
 258 pre-image estimation problem as the estimation of a linear transformation in
 259 the feature space, that ensures an isometry between the input and the feature
 260 spaces. Moreover, this result is extended to the estimation of a nonlinear trans-
 261 formation in the feature space, shown powerful on challenging data in Section
 262 5. Subsequently, we propose a local time-warp mapping function to embed
 263 time series into a vector space where the pre-image estimation can be estimated
 264 conveniently.
 265

266 *4.1. Learning linear and nonlinear transformations for pre-image estimation*

267 Let $X = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{t \times N}$ be a matrix giving the description of N univari-
 268 ate³ time series \mathbf{x}_i that we assume first lying in the metric space \mathbb{R}^t ; Section 4.2
 269 addresses the general case of time series that are lying into nonEuclidean space.
 270 The proposed pre-image method relies on learning a linear transformation R
 271 in the feature space that ensures an isometry between X and $\Phi(X)$. We first
 272 describe the method as a linear transformation, and then extend it to nonlinear
 273 transformations.

274 *Linear transformation*

275 The main idea to solve the pre-image problem is the isometry preserving,
 276 in the same spirit as the method described in Section 3.2. For this purpose,
 277 we formalise the pre-image problem as the estimation of the square matrix R
 278 that establishes an isometry between X and $\Phi(X)$, by solving the optimisation
 279 problem

$$R^* = \arg \min_R \|X^T X - \Phi(X)^T R \Phi(X)\|_{\mathcal{F}}^2. \quad (29)$$

280 By using a kernel PCA where a relevant subspace is considered, an explicit de-
 281 scription $P(\Phi(X)) \in \mathbb{R}^{p \times N}$ of $\Phi(X)$ is given and Eq. (29) can thus be rewritten

³For multivariate time series, simply define $X \in \mathbb{R}^{d \times t \times N}$.

282 as:

$$R^* = \arg \min_R \|X^T X - P(\Phi(X))^T R P(\Phi(X))\|_{\mathcal{F}}^2. \quad (30)$$

283 As $P(\Phi(X))P(\Phi(X))^T$ is invertible, a closed-form solution is given by:

$$R^* = (P(\Phi(X))P(\Phi(X))^T)^{-1} P(\Phi(X))X^T X P(\Phi(X))^T (P(\Phi(X))P(\Phi(X))^T)^{-1}. \quad (31)$$

284 The estimation of the time series \mathbf{x}^* , as the pre-image of $\varphi = \Phi(X)\gamma$, is then
285 given by:

$$\begin{aligned} \mathbf{x}^* &= (X X^T)^{-1} X P(\Phi(X))^T R^* P(\varphi) \\ &= (X X^T)^{-1} X P(\Phi(X))^T R^* \alpha^T K \gamma, \end{aligned} \quad (32)$$

286 with $P(\varphi) = \alpha^T K \gamma$ and α defined in Eq. (7).

287

288 One can easily include some regularisation terms in the optimisation prob-
289 lems (29) and (30), which can be easily propagated to the pre-image expression.
290 For example, in the case of non-invertible $X X^T$, a regularisation term is intro-
291 duced in Eq. (32) as:

$$\mathbf{x}^* = (X X^T + \lambda I_t)^{-1} X P(\Phi(X))^T R^* \alpha^T K \gamma, \quad (33)$$

292 for some positive regularisation parameter λ .

293 *Nonlinear transformation*

294 In the following, we propose to extend the above result to learn nonlinear
295 transformations for pre-image estimation. Let $\hat{\kappa}$ be a kernel defined on the
296 feature space \mathcal{H} , and $\hat{\Phi}$ the corresponding embedding function that maps any
297 element of \mathcal{H} into the Hilbert space defined by $\hat{\kappa}$. With some abuse of notation,
298 we denote $\hat{\Phi}(\Phi(X))$ the matrix of all mapped elements $\hat{\Phi}(\Phi(\mathbf{x}_i))$, for $i = 1, \dots, N$.
299 Let \hat{K} be the Gram matrix of general term $\hat{\kappa}(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j))$.

300

301 The pre-image estimation problem can be then defined as learning a nonlin-
302 ear transformation that defines an isometry between X and $\hat{\Phi}(\Phi(X))$ as:

$$R^* = \arg \min_R \|X^T X - \hat{\Phi}(\Phi(X))^T R \hat{\Phi}(\Phi(X))\|_{\mathcal{F}}^2. \quad (34)$$

303 Similarly, a closed-form solution for R^* can be obtained as:

$$\begin{aligned} R^* &= (P(\hat{\Phi}(\Phi(X)))P(\hat{\Phi}(\Phi(X)))^T)^{-1} P(\hat{\Phi}(\Phi(X))) \\ &\quad X^T X P(\hat{\Phi}(\Phi(X)))^T (P(\hat{\Phi}(\Phi(X)))P(\hat{\Phi}(\Phi(X)))^T)^{-1}, \end{aligned} \quad (35)$$

304 and

$$P(\hat{\Phi}(\Phi(X))) = \hat{\alpha}^T \hat{K}. \quad (36)$$

305 To estimate \widehat{K} , an indirect manner is to use a kernel PCA, with $\widehat{\kappa}(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) \approx$
306 $\widehat{\kappa}(P(\Phi(\mathbf{x}_i)), P(\Phi(\mathbf{x}_j)))$. A simpler way is possible when dealing with kernels
307 that are radial basis functions. For example, for the well-known Gaussian ker-
308 nel $\widehat{\kappa}$, \widehat{K} is estimated directly from K as:

$$\begin{aligned} \widehat{\kappa}(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) &= \exp\left(-\frac{\|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\|^2}{2\sigma^2}\right) \\ &= \exp\left(-\frac{\langle\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_i)\rangle - 2\langle\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle + \langle\Phi(\mathbf{x}_j), \Phi(\mathbf{x}_j)\rangle}{2\sigma^2}\right) \\ &= \exp\left(-\frac{\kappa(\mathbf{x}_i, \mathbf{x}_i) - 2\kappa(\mathbf{x}_i, \mathbf{x}_j) + \kappa(\mathbf{x}_j, \mathbf{x}_j)}{2\sigma^2}\right) \end{aligned} \quad (37)$$

309 The estimation of the pre-image of $\varphi = \sum_{i=1}^N \gamma_i \Phi(\mathbf{x}_i)$ is then given by the
310 time series \mathbf{x}^* :

$$\mathbf{x}^* = (XX^T)^{-1} X P(\widehat{\Phi}(\Phi(X)))^T R^* P(\widehat{\Phi}(\varphi)), \quad (38)$$

311 with $P(\widehat{\Phi}(\varphi)) = (\widehat{\mathbf{k}}_\varphi \widehat{\alpha})^T$, where $\widehat{\mathbf{k}}_\varphi$ is the vector whose i -th entry is

$$\widehat{\kappa}(\varphi, \Phi(\mathbf{x}_i)) = \exp\left(-\frac{\gamma^T K \gamma - 2\gamma^T \mathbf{k}_{\mathbf{x}_i}^T + K_{ii}}{2\sigma^2}\right). \quad (39)$$

312 The above proposed formulations and results for pre-image estimation (Sec-
313 tion 4.1) present some similarities and differences with the method proposed
314 in [16] and presented in Section 3.2. First of all, both approaches propose for-
315 mulations and solutions that only require linear algebra and are independent
316 of the type of kernel. To establish the isometry, in [16] a linear transforma-
317 tion restricted to the form $R = \Phi(X)AA^T\Phi(X)^T$ is estimated, whereas in our
318 proposal the estimated R may be linear Eq.(30) or non linear Eq.(34) and is im-
319 portantly unconstrained, namely of general form which enlarges its potential to
320 deal with complex structures. Finally, while in [16] the solution Eq.(24) involves
321 the kernel information through the regularisation term, which may be canceled
322 for lower values of λ , in the proposed solutions Eq.(33) and Eq.(38) the kernel
323 information is entirely considered regardless of the regularisation specifications.
324

325 4.2. Learning a metric space embedding for time series pre-image estimation

326 In Section 4.1, time series are assumed of the same length and lying in a met-
327 ric space. However, in general $X = \{\mathbf{x}_i\}_{i=1}^N$ is instead composed of time series
328 \mathbf{x}_i of different lengths t_i that are located in a non-metric space, rendering the
329 previous results as well as the pre-image estimation related works not applicable.
330

331 To address the pre-image estimation for such challenging time series, we de-
332 fine an embedding function that allows to represent the time series in a metric

333 space, where the previous linear and nonlinear transformations method for pre-
 334 image estimation can be performed conveniently.

335
 336 For this purpose, first we define \mathcal{N}_φ in \mathcal{H} and \mathcal{N}_φ^{-1} as the set of the n -closest
 337 neighbours of φ and its pre-image, given as:

$$\begin{aligned} \mathcal{N}_\varphi &= \left\{ \Phi(\mathbf{x}_i) \mid \langle \Phi(\mathbf{x}_i), \varphi \rangle = \sum_{j=1}^N \gamma_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \text{ is among the } n \text{ highest values} \right\}, (40) \\ \mathcal{N}_\varphi^{-1} &= \{ \mathbf{x}_i \mid \Phi(\mathbf{x}_i) \in \mathcal{N}_\varphi \} \end{aligned} \quad (41)$$

338 Let $\Phi(\mathbf{x}_r)$ be the representative of \mathcal{N}_φ with $\mathbf{x}_r \in \mathbb{R}^{t^*}$ defined as:

$$\Phi(\mathbf{x}_r) = \arg \max_{\Phi(\mathbf{x}_i) \in \mathcal{N}_\varphi} \sum_{\Phi(\mathbf{x}_j) \in \mathcal{N}_\varphi} \kappa(\mathbf{x}_i, \mathbf{x}_j). \quad (42)$$

To resorb the arising delays, a temporal alignment between each \mathbf{x}_i and \mathbf{x}_r is then performed by dynamic programming. An alignment $\boldsymbol{\pi}$ of length $|\boldsymbol{\pi}| = m$ between \mathbf{x}_i and \mathbf{x}_r is defined as the set of m increasing couples

$$\boldsymbol{\pi} = ((\pi_1(1), \pi_2(1)), (\pi_1(2), \pi_2(2)), \dots, (\pi_1(m), \pi_2(m))),$$

339 where the applications π_1 and π_2 defined from $\{1, \dots, m\}$ to $\{1, \dots, t_i\}$ and $\{1, \dots, t^*\}$
 340 respectively obey to the following boundary and monotonicity conditions:

$$\begin{aligned} 341 \quad & 1 = \pi_1(1) \leq \pi_1(2) \leq \dots \leq \pi_1(m) = t_i \\ 342 \quad & 1 = \pi_2(1) \leq \pi_2(2) \leq \dots \leq \pi_2(m) = t^* \end{aligned}$$

343 and $\forall l \in \{1, \dots, m\}$, $\pi_1(l+1) \leq \pi_1(l) + 1$ and $\pi_2(l+1) \leq \pi_2(l) + 1$, $(\pi_1(l+1) -$
 344 $\pi_1(l)) + (\pi_2(l+1) - \pi_2(l)) \geq 1$.

345
 346 Intuitively, an alignment $\boldsymbol{\pi}$ between \mathbf{x}_i and \mathbf{x}_r describes a way to associate
 347 each element of \mathbf{x}_i to one or more elements of \mathbf{x}_r and vice-versa. Such an
 348 alignment can be conveniently represented by a path in the $t_i \times t^*$ grid, as
 349 shown in Figure 1 (left), where the above monotonicity conditions ensure that
 350 the path is neither going back nor jumping. The optimal alignment $\boldsymbol{\pi}^*$ between
 351 \mathbf{x}_i and \mathbf{x}_r is then obtained as:

$$\boldsymbol{\pi}^* = \arg \min_{\boldsymbol{\pi}=(\pi_1, \pi_2)} \|\mathbf{x}_i^{\pi_1} - \mathbf{x}_r^{\pi_2}\|^2 \quad (43)$$

352 where $\mathbf{x}_i^{\pi_1} = (x_{i \pi_1(1)}, \dots, x_{i \pi_1(m)})$ and $\mathbf{x}_r^{\pi_2} = (x_{r \pi_2(1)}, \dots, x_{r \pi_2(m)})$ are \mathbf{x}_i and
 353 \mathbf{x}_r aligned through $\boldsymbol{\pi}$.

354
 355 We define f_r , the temporal embedding function, to embed time series $\mathbf{x}_i \in$
 356 $\mathbb{R}^{d \times t_i}$ into a new temporal metric space as:

$$\begin{aligned} f_r : \quad X &\longrightarrow \tilde{X} \subset \tilde{\mathcal{I}} = \mathbb{R}^{d \times t^*} \\ \mathbf{x}_i &\longrightarrow f_r(\mathbf{x}_i) = \mathbf{x}_i W_{ir} N_{ir} \end{aligned} \quad (44)$$

357 where $W_{ir} \in \{0, 1\}^{t_i \times t^*}$ is the binary matrix related to the optimal tempo-
 358 ral alignment between \mathbf{x}_i and \mathbf{x}_r , as shown in Figure 1 (right). The matrix

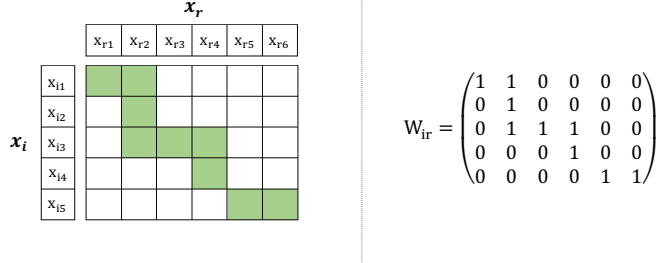


Figure 1: In the left, the temporal alignment between \mathbf{x}_i ($t_i = 5$) and \mathbf{x}_r ($t^* = 6$), the optimal alignment π^* is indicated in green. In the right, the adjacency binary matrix related to the optimal temporal alignment.

359 $N_{ir} = \text{diag}(W_{ir}^T \mathbf{1}_{t_i})^{-1}$ is the weight diagonal matrix of order t^* , of general term
360 $\frac{1}{|N_t|}$, that gives the weight of the element t of \mathbf{x}_r , where $|N_t|$ is the number of
361 time stamps of \mathbf{x}_i aligned to t . In particular, note that \mathbf{x}_r remains unchanged
362 by f_r , as $W_{rr} = N_{rr} = \text{diag}([1, 1, \dots, 1])$.

363
364 The set of embedded time series $\tilde{X} = \{f_r(\mathbf{x}_1), \dots, f_r(\mathbf{x}_N)\}$ is for now lying
365 in a metric space $\tilde{\mathcal{I}}$, where the delays are resorbed w.r.t. the representative
366 time series \mathbf{x}_r . The pre-image solution provided in the method described in
367 Section 4.1 can be developed to establish a linear or nonlinear transformations
368 to preserve an isometry between \tilde{X} and $\Phi(X)$. The algorithm for the proposed
369 solution TSPRIMA is summarised in Algorithm 1.

Algorithm 1 TSPRIMA: Pre-image estimation for time series

- 1: **Input:** $\{\mathbf{x}_i\}_{i=1}^N$ with $\mathbf{x}_i \in \mathbb{R}^{d \times t_i}$, κ (a temporal kernel), $\hat{\kappa}$ (a Gaussian kernel),
 - 2: γ (with $\varphi = \sum_{i=1}^N \gamma_i \Phi(\mathbf{x}_i)$), n (the neighbourhood size)
 - 3: **Output:** \mathbf{x}^* the pre-image estimation of φ
 - 4:
 - 5: Define \mathcal{N}_φ , \mathcal{N}_φ^{-1} and \mathbf{x}_r using respectively (40), (41) and (42)
 - 6: Embed \mathcal{N}_φ^{-1} into a temporal metric space by using Eq. (44), set $\tilde{N}_\varphi^{-1} = f_r(\mathcal{N}_\varphi^{-1})$
 - 7: Set $X = \tilde{N}_\varphi^{-1}$ and $\Phi(X) = \mathcal{N}_\varphi$
 - 8: Learn a linear (resp. nonlinear) transformation R by using Eq. (31) (resp. Eq. (35))
 - 9: Estimate the pre-image \mathbf{x}^* based on a linear (resp. nonlinear) transformation using Eq. (33) (resp. Eq. (38))
-

370 **5. Experiments**

371 In this section, we evaluate the efficiency of the proposed pre-image esti-
 372 mation method under three major time series analysis tasks: 1) time series
 373 averaging, 2) time series reconstruction and denoising and 3) time series rep-
 374 resentation learning. The proposed pre-image estimation method TsPRIMA is
 375 compared to three major alternative approaches introduced in Section 3, and
 376 referenced in the following as Honeine, Kwok and Bakir methods. The exper-
 377 iments are conducted on 33 public datasets (Table 1) including univariate and
 378 multivariate time series data, that may involve varying delays and be of the
 379 same or different lengths. The 25 first datasets in Table 1 are selected from the
 380 archive given in [8, 11] by using three selection criteria: a) have a reasonable
 381 number of classes (Nb. of Classes < 50), b) have a sufficient size for train and
 382 test samples (Train size <= 500 and Test size < 3000), c) avoid time series of
 383 extra large lengths (Time series length < 700). To obtain a manageable number
 384 of datasets, the 3 above selection criteria are applied on the top 40 datasets, in
 385 the order set out in [8]. The 25 obtained datasets are composed of univariate
 386 time series and half of the datasets include significant delays. We consider a
 387 dataset as including significant delays if the difference between the 1-NN Eu-
 388 clidean distance error and the 1-NN Dynamic time warping [21] error is greater
 389 than 5%. The 5 next datasets include univariate and multivariate time series
 390 covering local and noisy salient events as described in [30, 27, 14] and the three
 391 last datasets are related to handwritten digits and characters, they are described
 392 as multivariate time series of variable lengths [7]. In the following, we detail the
 393 evaluation process of the pre-image estimation methods then give and discuss
 394 the obtained results.

395 *5.1. Time series averaging*

396 Estimating the centroid of a set of time series is a major topic for many time
 397 series analytics as summarisation, prototype extraction or clustering. Time se-
 398 ries averaging has been an active area in the last decade, where the propositions
 399 mainly focus on tackling the tricky problem of multiple temporal alignments
 400 [14, 26, 27]. A suitable way to circumvent the problem of multiple temporal
 401 alignments is to use a temporal kernel method to evaluate the time series cen-
 402 troid in the feature space. The pre-image of the centroid is then estimated to
 403 obtain the time series averaging in the input space.

404 In that context, let $\{\mathbf{x}_i\}_{i=1}^N$ and $\{\Phi(\mathbf{x}_i)\}_{i=1}^N$ be, respectively, a set of time
 405 series and their mapped images into the Hilbert space \mathcal{H} related to the temporal
 406 kernel DTAK [25]. Let $\varphi = \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i)$ be the centroid of the mapped time
 407 series in the feature space and \mathbf{x}^* its pre-image in the input space. The quality
 408 of the obtained centroids is given by the within-class similarity $\sum_i \text{DTAK}(\mathbf{x}^*, \mathbf{x}_i)$;
 409 the higher the within-class similarity, the better is the estimated centroid.

410 To evaluate the efficiency of each pre-image estimation method, the time
 411 series centroid is estimated for each class of the studied datasets and the induced
 412
 413

Table 1: Data Description

Dataset	Nb. Classes	Train size	Test size	Time series length	Univariate
CC	6	300	300	60	✓
GunPoint	2	50	150	150	✓
CBF	3	30	900	128	✓
OSULeaf	6	200	242	427	✓
SwedishLeaf	15	500	625	128	✓
Trace	4	100	100	275	✓
FaceFour	4	24	88	350	✓
Lighting2	2	60	61	637	✓
Lighting7	7	70	73	319	✓
ECG200	2	100	100	96	✓
Adiac	37	390	391	176	✓
FISH	7	175	175	463	✓
Beef	5	30	30	470	✓
Coffee	2	28	28	286	✓
OliveOil	4	30	30	570	✓
DiatomSizeR	4	16	306	345	✓
ECG5Days	2	23	861	136	✓
FacesUCR	14	200	2050	131	✓
ItalyPowerD	2	67	1029	24	✓
MedicalImages	10	381	760	99	✓
MoteStrain	2	20	1252	84	✓
SonyAIBOII	2	27	953	65	✓
SonyAIBO	2	20	601	70	✓
Symbols	6	25	995	398	✓
TwoLeadECG	2	23	1139	82	✓
SPIRAL1	1	50	50	101	✗
SPIRAL2	1	50	50	300	✗
PowerCons	2	73	292	144	✓
BME	3	30	150	128	✓
UMD	3	36	144	150	✓
DIGITS	10	100	100	29~218	✗
LOWER	26	130	260	27~163	✗
UPPER	26	130	260	27~412	✗

414 within-class similarity is evaluated. The average within-class similarity is then
415 reported in Table 2 for each dataset and each pre-image estimation method; the
416 best values are indicated in bold (t-test at 5% risk). In addition, a Nemenyi
417 test [12] is performed to compare the significance of the obtained results, with
418 the related critical difference diagram given in Figure 2. The estimated time
419 series centroids for some challenging classes are shown in Figure 3, where we
420 retain particularly SPIRAL1 and the handwritten digits and characters datasets

421 (DIGITS, LOWER and UPPER) as they are more intuitive to visually evaluate the
422 quality of the estimated time series centroids.

Table 2: Average within-class similarity of the estimated time series centroids

DataSet	TsPRIMA	Honeine	Kwok	Bakir
CC	0.744	0.709	0.721	0.709
GunPoint	0.902	0.910	0.882	0.886
CBF	0.798	0.737	0.755	0.737
OSULeaf	0.985	0.987	0.986	0.987
SwedishLeaf	0.910	0.920	0.920	0.920
Trace	0.998	0.992	0.991	0.992
FaceFour	0.981	0.980	0.981	0.98
Lighting2	0.918	0.876	0.859	0.875
Lighting7	0.964	0.930	0.930	0.931
ECG200	0.593	0.565	0.567	0.566
Adiac	0.997	0.997	0.996	0.997
FISH	0.996	0.995	0.994	0.995
Beef	0.900	0.892	0.898	0.890
Coffee	0.998	0.998	0.998	0.998
OliveOil	0.999	0.999	0.998	0.999
DiatomSizeR	0.997	0.997	0.997	0.997
ECG5Days	0.777	0.746	0.417	0.746
FacesUCR	0.721	0.699	0.648	0.700
ItalyPowerD	0.610	0.552	0.420	0.542
MedicalImages	0.671	0.644	0.637	0.646
MoteStrain	0.776	0.777	0.701	0.777
SonyAIBOII	0.749	0.740	0.716	0.740
SonyAIBO	0.960	0.962	0.955	0.962
Symbols	0.959	0.949	0.904	0.951
TwoLeadECG	0.980	0.977	0.911	0.977
SPIRAL1	0.831	0.823	0.799	0.824
SPIRAL2	0.947	0.940	0.934	0.940
PowerCons	0.458	0.328	0.436	0.330
BME	0.701	0.572	0.638	0.555
UMD	0.800	0.765	0.724	0.755
DIGITS	0.746	0.575	0.657	0.581
LOWER	0.713	0.544	0.645	0.545
UPPER	0.764	0.572	0.570	0.573
Nb. Best	28	9	4	8
Avg. Rank	1.5	2.68	3.24	2.58

423 *5.2. Time series reconstruction and denoising*

424 The reconstruction and denoising tasks represent a standard application con-
425 text for pre-image estimation. For the time series reconstruction task, a kernel

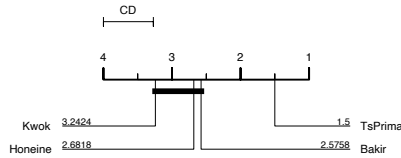


Figure 2: Nemenyi test: comparison of pre-image methods under centroid estimation task

426 PCA is performed on the training set, the reconstruction of a given test sample
 427 \mathbf{x} is then defined as the pre-image \mathbf{x}^* of its kernel PCA projection $P(\Phi(\mathbf{x}))$.
 428 The latter takes the form $\boldsymbol{\varphi} = \Phi(X)\boldsymbol{\gamma}$, with $\boldsymbol{\gamma}$ defined as:

$$\boldsymbol{\gamma} = (I_N - \mathbf{1}_N) \boldsymbol{\alpha} \boldsymbol{\alpha}^T \tilde{\mathbf{k}}_{\mathbf{x}} + \frac{1}{N} \mathbf{1}_N \quad (45)$$

429 The quality of the reconstruction is then measured as the similarity $\text{DTAK}(\mathbf{x}^*, \mathbf{x})$
 430 between each test sample \mathbf{x} and its reconstruction \mathbf{x}^* ; the higher the criterion,
 431 the better is the reconstruction. Table 3 gives the average quality of recon-
 432 struction obtained for each dataset and each method. Figure 4 gives the critical
 433 difference diagram related to the Nemenyi test for the average ranking compar-
 434 ison of the studied methods. Figure 5 shows the reconstructions obtained for
 435 some challenging time series of DIGITS, LOWER and UPPER datasets.

436
 437 For the time series denoising task, first a kernel PCA is performed on the
 438 training set, then a $(0, \sigma^2)$ Gaussian noise is added to the test samples \mathbf{x} to
 439 generate noisy samples $\tilde{\mathbf{x}}$ with different variances σ^2 . The denoised sample is
 440 obtained as the pre-image \mathbf{x}^* of its kernel PCA projection $P(\Phi(\tilde{\mathbf{x}}))$, with $\boldsymbol{\gamma}$
 441 defined as in Eq. (45). Similarly, the quality of the denoising is measured as the
 442 similarity $\text{DTAK}(\mathbf{x}^*, \mathbf{x})$ between \mathbf{x}^* and the initial \mathbf{x} . Table 4 gives, for different
 443 values of σ^2 , the average quality of the denoising for some datasets. Figure 6
 444 illustrates the denoising results for some challenging times series of the noisy
 445 SPIRAL2 data and of the class “M” of UPPER dataset.

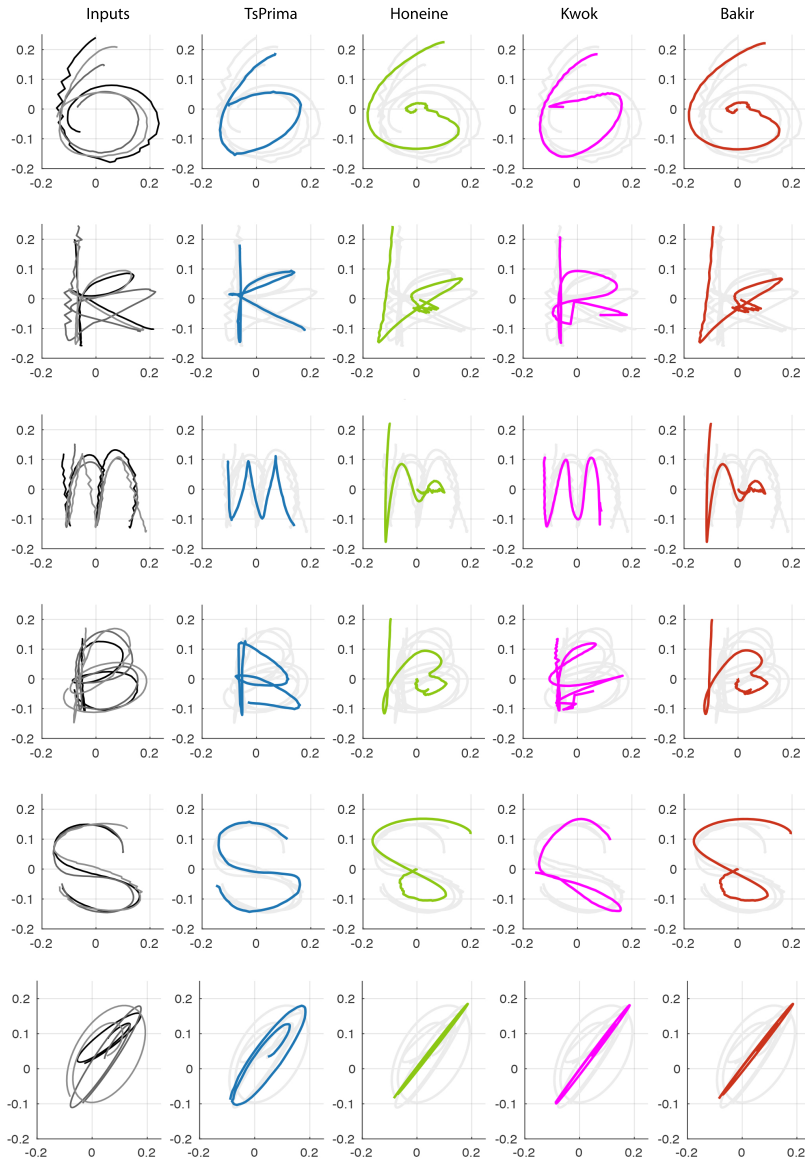


Figure 3: Time series centroids for some challenging classes of DIGITS, LOWER, UPPER and SPIRAL1 datasets

Table 3: Quality of the time series reconstruction under kernel PCA

DataSet	TSPRIMA	Honeine	Kwok	Bakir
CC	0.798	0.747	0.758	0.747
GunPoint	0.994	0.996	0.992	0.990
CBF	0.916	0.854	0.896	0.875
OSULeaf	0.997	0.998	0.995	0.998
SwedishLeaf	0.798	0.701	0.690	0.650
Trace	0.689	0.519	0.597	0.519
FaceFour	0.981	0.951	0.967	0.964
Lighting2	0.993	0.967	0.984	0.975
Lighting7	0.954	0.920	0.938	0.922
ECG200	0.965	0.979	0.959	0.962
Adiac	0.194	0.127	0.139	0.125
FISH	0.779	0.580	0.586	0.579
Beef	0.528	0.703	0.643	0.704
Coffee	0.584	0.595	0.570	0.559
OliveOil	0.150	0.125	0.141	0.121
DiatomSizeR	0.330	0.174	0.186	0.173
ECG5Days	0.996	0.996	0.995	0.995
FacesUCR	0.939	0.825	0.878	0.847
ItalyPowerD	0.831	0.892	0.023	0.851
MedicalImages	0.946	0.906	0.935	0.928
MoteStrain	0.971	0.987	0.970	0.979
SonyAIBOII	0.978	0.989	0.969	0.985
SonyAIBO	0.939	0.98	0.924	0.967
Symbols	0.885	0.822	0.724	0.761
TwoLeadECG	0.825	0.630	0.444	0.669
SPIRAL1	0.961	0.939	0.933	0.911
SPIRAL2	0.966	0.939	0.946	0.940
PowerCons	0.971	0.966	0.955	0.977
BME	0.896	0.800	0.858	0.666
UMD	0.885	0.855	0.904	0.797
DIGITS	0.840	0.721	0.798	0.726
LOWER	0.787	0.696	0.747	0.685
UPPER	0.856	0.678	0.787	0.687
Nb. Best	22	9	1	3
Avg. Rank	1.56	2.67	2.71	3.06

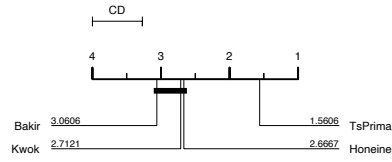


Figure 4: Nemenyi test: comparison of pre-image methods under kernel PCA reconstruction

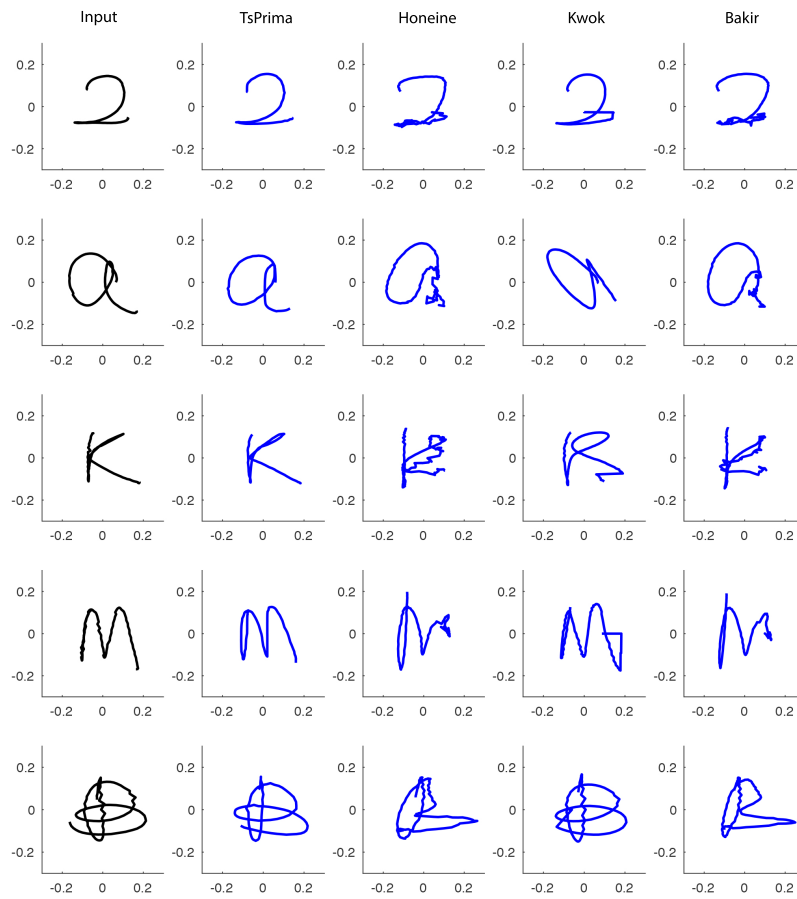


Figure 5: The time series reconstruction under kernel PCA of some samples of DIGITS, LOWER and UPPER datasets

Table 4: Quality of the denoising for several noise levels

DataSet	σ^2	TsPRIMA	Honeine	Kwok	Bakir
DIGITS	0.01	0.832	0.669	0.782	0.666
	0.05	0.808	0.619	0.742	0.627
	0.1	0.791	0.605	0.723	0.612
	0.15	0.783	0.598	0.719	0.606
LOWER	0.01	0.766	0.651	0.721	0.637
	0.05	0.746	0.614	0.689	0.606
	0.1	0.736	0.601	0.675	0.596
	0.15	0.729	0.594	0.670	0.591
UPPER	0.01	0.837	0.627	0.765	0.638
	0.05	0.806	0.579	0.712	0.600
	0.1	0.789	0.561	0.688	0.590
	0.15	0.782	0.554	0.679	0.586
Nb. Best		12	0	0	0
Avg. Rank		1.00	3.58	2.00	3.42

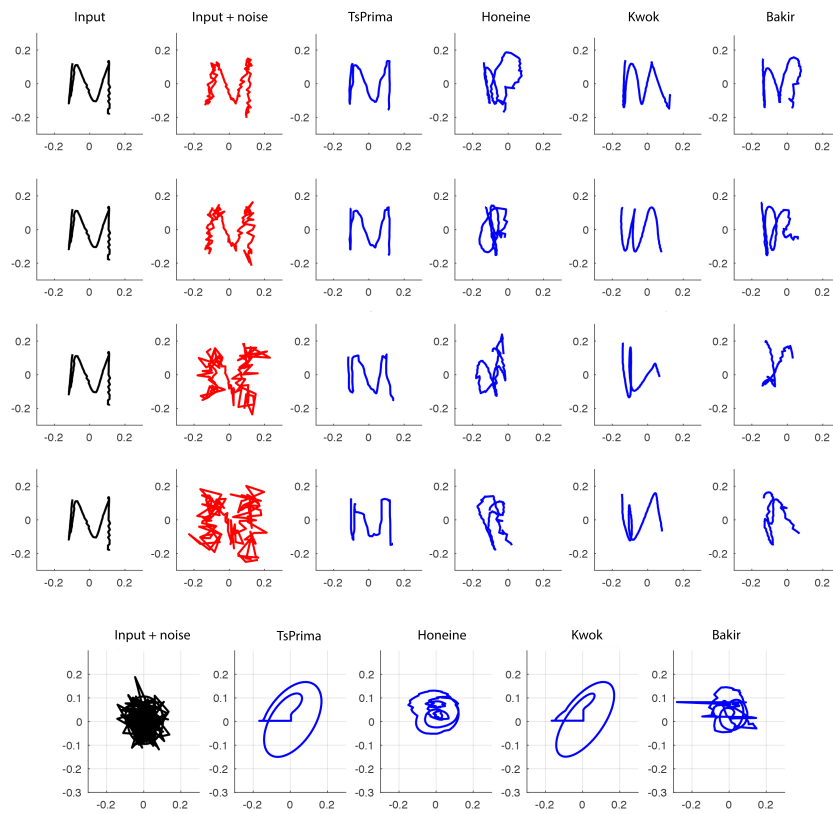


Figure 6: Time series denoising under kernel PCA of noisy samples of SPIRAL2 and of the class “M” of UPPER dataset.

446 5.3. Time series representation learning

447 For time series representation learning, the kernel k -SVD ($\tau = 5$) is used
 448 to learn, for each class of the considered datasets, the dictionary $\Phi(X)\mathcal{B}$ and
 449 the sparse representations $\mathcal{A} = [\mathbf{a}_1, \dots, \mathbf{a}_N]$ of its membership time series, as
 450 defined in Section 2.2. The pre-images D^* and X^* of the dictionary $\Phi(X)\mathcal{B}$ and
 451 of the sparse codes \mathcal{A} are then obtained by considering $\gamma = \mathcal{B}$ and $\gamma = \mathcal{B}\mathcal{A}$,
 452 respectively. The quality of the learned sparse representations is then measured
 453 as the similarity $\text{DTAK}(\mathbf{x}_i, \mathbf{x}_i^*)$ between each time series \mathbf{x}_i and the pre-image
 454 \mathbf{x}_i^* of the sparse representation $\Phi(X)\mathcal{B}\mathbf{a}_i$. Table 5 gives the average quality
 455 of the learned representations for each dataset and each pre-image estimation
 456 method. Figure 7 gives the critical difference diagram related to the Nemenyi
 457 test for the average ranking comparison of the studied methods. Figure 8 shows
 458 the learned representations for some time series of DIGITS, LOWER and UPPER
 459 datasets and Figure 9 illustrates, for a challenging sample of the class “k” of
 460 LOWER dataset, the learned representations as well as the top 3 atoms involved
 461 in its reconstruction.

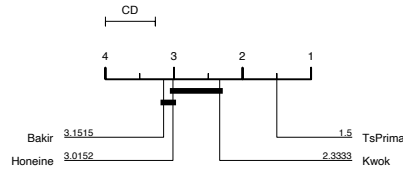


Figure 7: Nemenyi test: comparison of pre-image methods under kernel k -SVD representation learning

Table 5: Quality of the time series representation learning under Kernel k -SVD

DataSet	TSPRIMA	Honeine	Kwok	Bakir
CC	0.788	0.730	0.751	0.732
GunPoint	0.993	0.994	0.992	0.985
CBF	0.917	0.862	0.900	0.872
OSULeaf	0.996	0.996	0.995	0.996
SwedishLeaf	0.789	0.659	0.691	0.623
Trace	0.687	0.514	0.602	0.514
FaceFour	0.971	0.940	0.959	0.947
Lighting2	0.991	0.961	0.982	0.968
Lighting7	0.961	0.934	0.947	0.934
ECG200	0.953	0.957	0.950	0.941
Adiac	0.184	0.122	0.131	0.117
FISH	0.757	0.553	0.579	0.560
Beef	0.411	0.555	0.605	0.621
Coffee	0.596	0.607	0.586	0.560
OliveOil	0.145	0.133	0.152	0.120
DiatomSizeR	0.287	0.177	0.198	0.178
ECG5Days	0.996	0.996	0.995	0.994
FacesUCR	0.917	0.834	0.878	0.842
ItalyPowerD	0.800	0.781	0.034	0.728
MedicalImages	0.937	0.860	0.930	0.878
MoteStrain	0.969	0.970	0.971	0.970
SonyAIBOII	0.974	0.975	0.973	0.975
SonyAIBO	0.932	0.938	0.930	0.936
Symbols	0.811	0.785	0.794	0.755
TwoLeadECG	0.810	0.617	0.411	0.629
SPIRAL1	0.944	0.913	0.920	0.914
SPIRAL2	0.964	0.936	0.949	0.937
PowerCons	0.968	0.946	0.957	0.951
BME	0.872	0.734	0.843	0.622
UMD	0.888	0.842	0.905	0.788
DIGITS	0.822	0.699	0.793	0.706
LOWER	0.773	0.678	0.738	0.671
UPPER	0.840	0.664	0.797	0.675
Nb. Best	24	7	3	3
Avg. Rank	1.5	3.02	2.33	3.15

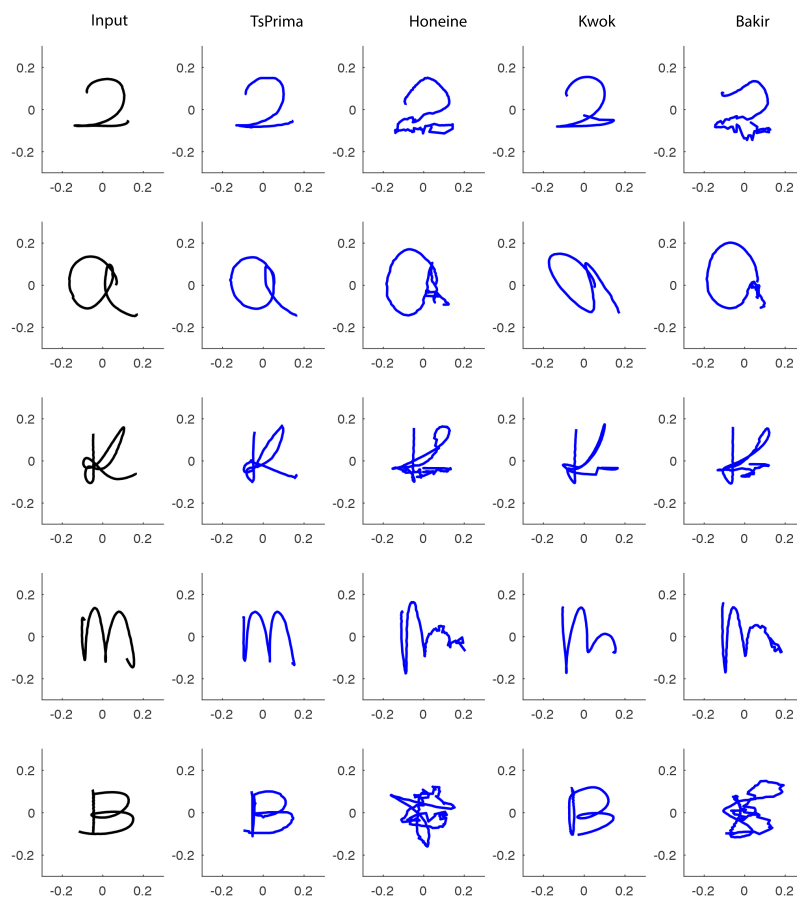


Figure 8: The learned time series representations under kernel k -SVD of some samples of DIGITS, LOWER, UPPER datasets

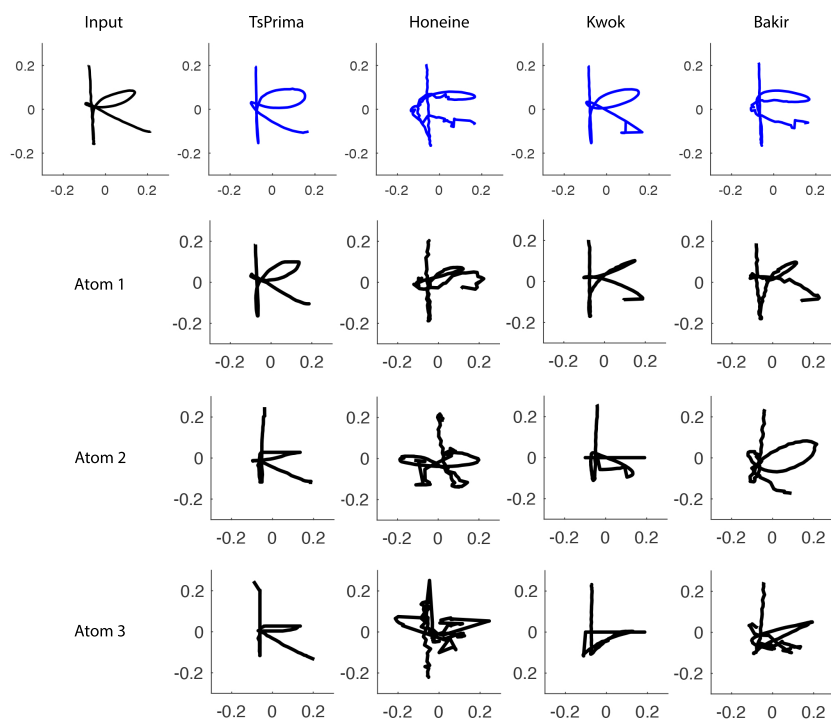


Figure 9: The sparse representation of a time series of the class “k” of LOWER dataset and the top 3 involved atoms for its reconstruction

462 5.4. Further comparison

463 In the previous experiments (Sections 5.1 to 5.3), we have evaluated the
 464 performances of TsPRIMA that are mainly due to two major ingredients : 1)
 465 the defined temporal embedding function f_r (Section 4.2) and 2) the proposed
 466 transformation R to preserve an isometry between the time series embedding
 467 space and the feature space (Section 4.1). In this last part, the aim is to evaluate
 468 the efficiency of the proposed transformation R , regardless of the effect of f_r .
 469 For that, TsPRIMA is compared to the alternative methods Honeine, Kwok and
 470 Bakir once all the time series embedded into the same metric space; namely,
 471 all the pre-image estimation methods are performed between the time series
 472 embedding space and the feature space. Similar experiments are performed on
 473 the 33 public datasets (Table 1), the results obtained for the three tasks are
 474 summarised into Table 6 and the related Nemenyi tests are given in Figure 10.

Table 6: Further comparisons for pre-image estimation

		TsPRIMA	Honeine	Kwok	Bakir
Averaging	Nb. Best	19	20	4	19
	Avg. Rank	2.23	2.21	3.35	2.21
Reconstruction (kernel PCA)	Nb. Best	24	10	0	1
	Avg. Rank	1.56	2.35	3.05	3.05
Denoising (kernel PCA)	Nb. Best	12	0	0	0
	Avg. Rank	1.50	3.25	2.62	3.12
Rep. Learning (kernel k SVD)	Nb. Best	25	8	1	2
	Avg. Rank	1.44	2.67	2.58	3.32

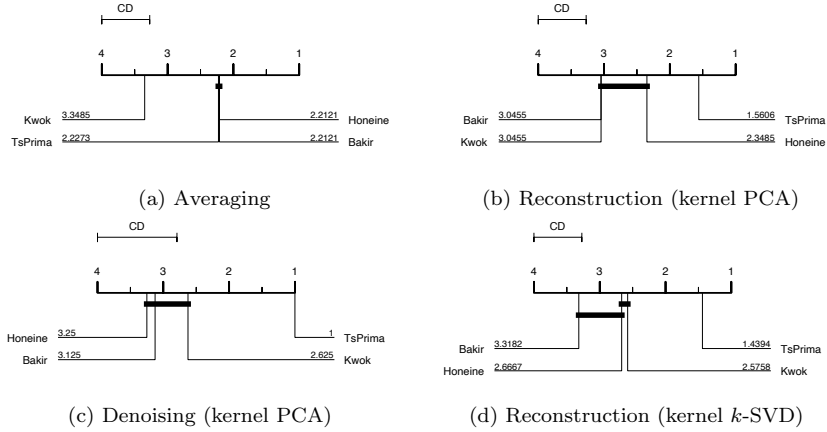


Figure 10: Nemenyi Tests.

475 *5.5. Overall analysis*

476 The experiments conducted in Sections 5.1 to 5.3 show that the proposed
477 method TSPRIMA leads on almost all the datasets and through the three stud-
478 ied tasks to the best results. On the other hand, the performances obtained by
479 the alternative methods seem slightly equivalent and lower than those obtained
480 by TSPRIMA.

481
482 In particular, for time series averaging task, we can see in Table 2 that the
483 centroids estimated by TSPRIMA lead to the highest within-class similarity on
484 almost all the datasets; namely, each centroid obtained by TSPRIMA is in gen-
485 eral the closest to the set of time series it represents. The analysis of the critical
486 difference diagram given in Figure 2 indicates that the next best results are ob-
487 tained respectively by Bakir, Honeine and Kwok methods. In addition, as the
488 state of the art methods are connected by a solid bold line, their performances
489 remain equivalent. From Figure 3, we can see that while all the methods succeed
490 to reconstitute the centroids of some input classes (shown on the left column) as
491 the class "6" of DIGITS and "S" of UPPER dataset, only TSPRIMA succeeds to
492 estimate the centroids of the most challenging classes, as the "k" class of LOWER
493 dataset and SPIRAL1.

494
495 For time series reconstruction, Table 3 shows that TSPRIMA leads to the
496 highest reconstruction accuracies through almost all the datasets, followed by
497 Honeine, Bakir and Kwok methods. Figure 4 indicates that there is no signifi-
498 cant difference between the performances of the three state of the art methods
499 (connected by a solid bold line). These results are assessed in Figure 5 that
500 shows, for some input time series, the quality of the reconstructions obtained
501 by TSPRIMA and the state of the art methods.

502
503 For the time series denoising task, we observe from Table 4 and for all the
504 methods that the quality of the denoising decreases when the intensity of noise
505 increases. This result is illustrated in Figure 6, that shows the denoising results
506 of the time series "M" of UPPER dataset and of the highly noisy time series of
507 SPIRAL2 dataset. In particular, note that that Kwok and TSPRIMA methods
508 lead to the best results on SPIRAL2 data and seem less sensitive to noise than
509 Honeine and Bakir methods.

510
511 Lastly, for time series representation learning task, Table 5 indicates that
512 each studied method leads to the best sparse representations for at least some
513 datasets and that TSPRIMA performs better on almost all the datasets. Fig-
514 ure 8 shows the goodness of the sparse representations obtained. While all the
515 methods succeed to sparse represent some input time series, the time series of
516 "k" and "B" classes appear challenging for Honeine and Bakir methods. In Fig-
517 ure 9, we get a look on the quality of the learned atoms, that are involved into
518 the reconstruction of the input samples. The first row gives for some input sam-
519 ple "k" (on the left), the sparse representations learned by each method. The

520 three next rows, provide the three first atoms involved into the reconstructions.
 521 We can see that while the first atom learned by TsPRIMA is nearly sufficient
 522 to sparse represent the "k" input sample, the state of the art methods need
 523 obviously more than one atom to sparse represent the input sample. Finally,
 524 the analysis of Figure 7 indicates that Honeine method performs equivalently
 525 than Kwok and Bakir, whereas the Kwok performances are significantly better
 526 than those of Bakir method.

527
 528 Further comparisons (Table 6) are conducted in Section 5.4 to evaluate the
 529 efficiency of TsPRIMA related to the learned transformation R , regardless of
 530 the temporal embedding f_r . For averaging task, TsPRIMA, Honeine and Bakir
 531 lead equivalently to the best performances, followed by Kwok method (Figure
 532 10 (a)). From these results we can conjecture that, linear transformations seem
 533 sufficient to achieve good pre-image estimations for averaging task on these
 534 datasets, as both linear and non linear approaches (TsPRIMA, Honeine, Bakir)
 535 perform equivalently. Furthermore, while Honeine and Bakir involve the whole
 536 datasets for the centroid pre-image estimations, Kwok uses a subset of samples
 537 into the neighbourhood of φ , which may explain the slightly lower performances
 538 of Kwok method. Note that, although TsPRIMA involves, similarly to Kwok
 539 method, fewer samples into the neighbourhood of φ , it succeeds to reach the
 540 best performances thanks to the efficiency of the learned transformation R .
 541 For the remaining tasks reconstruction, denoising and representation learning,
 542 TsPRIMA achieves the highest performances, followed by far by Honeine, Kwok
 543 and Bakir (Figure 10 (b), (c) and (d)), which assesses the crucial contribution
 544 of the learned transformations R of TsPRIMA. Lastly, of particular note is that
 545 Honeine and Bakir that involve the whole training samples induce much com-
 546 putations, specifically for the time series embedding process, than Kwok and
 547 TsPRIMA that require fewer samples into the neighbourhood of φ .

548
 549 Finally, as all the studied methods propose closed-form solutions, they lead
 550 to comparable complexities. However, for large data, TsPRIMA and Kwok meth-
 551 ods are expected to perform faster as requiring fewer samples on the neighbor-
 552 hood of φ than Honeine and Bakir that involve the whole samples for pre-image
 553 estimation. Note that the complexity of the proposed solutions is mainly re-
 554 lated to the matrix inversion operator. In Kwok method, the inversion of ZZ^T
 555 required in Eq. (21), with Z of dimension $(q \times n)$ and n is the neighbourhood
 556 size, induces a complexity of $O(q^2n) + O(q^3)$; as q is in general small and fixed
 557 beforehand, the overall complexity is about $O(n)$. For Honeine method, Eq.
 558 (24) requires two inversions of XX^T and K , which induces, respectively, a com-
 559 plexity of $O(d^2N) + O(d^3)$ and $O(N^3)$, that leads to an overall complexity of
 560 $O(N^3)$. For Bakir method, Eq. (26), requires the inversion of the Gram matrix,
 561 which leads to a complexity of $O(N^3)$. For TsPRIMA, Eq. (32) involves the
 562 inversion of XX^T , with X is of dimension $(d \times n)$, d is the time series length
 563 and n is the neighbourhood size. The induced complexity is of $O(d^2n) + O(d^3)$.
 564 For the time series embedding part, the complexity is mainly related to the
 565 time warping function which is of order $O(d^2n)$. As d is in general higher than

566 the neighbourhood size n , the overall complexity for TSPrima is about $O(d^3)$.
567 To sum up, as the neighbourhood size $n \ll N$ and $d \ll N$ (for not extra
568 large time series), the complexity induced by both Kwok and TSPrima remains
569 lower than the one of Honeine and Bakir. Note that, the Honeine method can
570 be developed to consider only the neighbourhoods instead of all samples.

571 6. Conclusion

572 This work proposes TsPRIMA, a new closed-form pre-image estimation method
573 for time series analytics under kernel machinery. The method consists of two
574 stages. In the first step, we define a time warp embedding function, driven by
575 distance constraints in the feature space, that allows to embed the time series in
576 a metric space. In the second step, the time series pre-image estimation is cast
577 as learning a linear (or a nonlinear) transformation to ensure a local isometry
578 between the time series embedding space and the feature space. Extensive ex-
579 periments show the efficiency and the benefits of TsPRIMA through three major
580 tasks that require pre-image estimation: 1) time series averaging, 2) time series
581 reconstruction and denoising and 3) time series representation and dictionary
582 learning. Future work will focus on using pre-image estimation methods to en-
583 hance the interpretability and the computation of deep learning tasks for time
584 series, sequence and graph analytics.

585 Acknowledgment

586 This work is supported by the French National Research Agency through
587 the projects LOCUST (ANR-15-CE23-0027-02) and API (ANR-18-CE23-0014).

588 References

- 589 [1] Aharon, M., Elad, M., Bruckstein, A., 2006. K-svd: An algorithm for
590 designing overcomplete dictionaries for sparse representation. *IEEE Trans-*
591 *actions on signal processing* 54, 4311–4322.
- 592 [2] Bahlmann, C., Haasdonk, B., Burkhardt, H., 2002. Online handwriting
593 recognition with support vector machines—a kernel approach, in: *Proceed-*
594 *ings Eighth International Workshop on Frontiers in Handwriting Recogni-*
595 *tion*, IEEE. pp. 49–54.
- 596 [3] Bakır, G.H., Weston, J., Schölkopf, B., 2004. Learning to find pre-images.
597 *Advances in neural information processing systems* 16, 449–456.
- 598 [4] Balasubramanian, K., Yu, K., Lebanon, G., 2013. Smooth sparse coding
599 via marginal regression for learning sparse representations, in: *International*
600 *Conference on Machine Learning*, pp. 289–297.
- 601 [5] Bengio, S., Pereira, F., Singer, Y., Strelow, D., 2009. Group sparse coding,
602 in: *Advances in neural information processing systems*, pp. 82–89.

- 603 [6] Bianchi, F.M., Livi, L., Mikalsen, K.Ø., Kampffmeyer, M., Jenssen,
604 R., 2018. Learning representations for multivariate time series with
605 missing data using temporal kernelized autoencoders. arXiv preprint
606 arXiv:1805.03473 .
- 607 [7] Chen, M., AlRegib, G., Juang, B.H., 2012. 6dmg: A new 6d motion ges-
608 ture database, in: Proceedings of the 3rd Multimedia Systems Conference,
609 ACM. pp. 83–88.
- 610 [8] Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista,
611 G., 2015. The ucr time series classification archive URL: [https://www.cs.
612 ucr.edu/~eamonn/time_series_data/](https://www.cs.ucr.edu/~eamonn/time_series_data/).
- 613 [9] Cloninger, A., Czaja, W., Doster, T., 2017. The pre-image problem for
614 laplacian eigenmaps utilizing l 1 regularization with applications to data
615 fusion. *Inverse Problems* 33, 074006.
- 616 [10] Cuturi, M., Vert, J.P., Birkenes, O., Matsui, T., 2007. A kernel for time
617 series based on global alignments, in: 2007 IEEE International Conference
618 on Acoustics, Speech and Signal Processing-ICASSP’07, IEEE. pp. II–413.
- 619 [11] Dau, H.A., Keogh, E., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S.,
620 Ratanamahatana, C.A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen,
621 A., Batista, G., Hexagon-ML, 2018. The ucr time series classification
622 archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.
- 623 [12] Demšar, J., 2006. Statistical comparisons of classifiers over multiple data
624 sets. *Journal of Machine learning research* 7, 1–30.
- 625 [13] Do, C.T., Douzal-Chouakria, A., Marié, S., Rombaut, M., Varasteh, S.,
626 2017. Multi-modal and multi-scale temporal metric learning for a robust
627 time series nearest neighbors classification. *Information Sciences* 418, 272–
628 285.
- 629 [14] Frambourg, C., Douzal-Chouakria, A., Gaussier, E., 2013. Learning multi-
630 temporal matching for time series classification, in: *International Sym-
631 posium on Intelligent Data Analysis*, Springer. pp. 198–209.
- 632 [15] Honeine, P., Richard, C., 2011a. A closed-form solution for the pre-image
633 problem in kernel-based machines. *Journal of Signal Processing Systems*
634 65, 289 – 299. URL: [https://link.springer.com/article/10.1007/
635 s11265-010-0482-9](https://link.springer.com/article/10.1007/s11265-010-0482-9), doi:10.1007/s11265-010-0482-9.
- 636 [16] Honeine, P., Richard, C., 2011b. Preimage problem in kernel-based
637 machine learning. *IEEE Signal Processing Magazine* 28, 77 – 88.
638 URL: <https://ieeexplore.ieee.org/document/5714388>, doi:10.1109/
639 MSP.2010.939747.
- 640 [17] Jenatton, R., Mairal, J., Obozinski, G., Bach, F.R., 2010. Proximal meth-
641 ods for sparse hierarchical dictionary learning., in: *ICML, Citeseer*. p. 2.

- 642 [18] Kwok, J.Y., Tsang, I.H., 2004. The pre-image problem in kernel methods.
643 IEEE transactions on neural networks 15, 1517–1525.
- 644 [19] Paparrizos, J., Franklin, M.J., 2019. Grail: efficient time-series representa-
645 tion learning. Proceedings of the VLDB Endowment 12, 1762–1777.
- 646 [20] Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S., 1993. Orthogonal matching
647 pursuit: Recursive function approximation with applications to wavelet
648 decomposition, in: Proceedings of 27th Asilomar conference on signals,
649 systems and computers, IEEE. pp. 40–44.
- 650 [21] Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization
651 for spoken word recognition. IEEE transactions on acoustics, speech, and
652 signal processing 26, 43–49.
- 653 [22] Schlegel, K., 2019. When is there a representer theorem? Journal of Global
654 Optimization 74, 401–415. doi:[10.1007/s10898-019-00767-0](https://doi.org/10.1007/s10898-019-00767-0).
- 655 [23] Schölkopf, B., Mika, S., Smola, A., Rätsch, G., Müller, K.R., 1998. Kernel
656 pca pattern reconstruction via approximate pre-images, in: International
657 Conference on Artificial Neural Networks, Springer. pp. 147–152.
- 658 [24] Scholkopf, B., Smola, A.J., 2001. Learning with kernels: support vector
659 machines, regularization, optimization, and beyond. MIT press.
- 660 [25] Shimodaira, H., Noma, K.i., Nakai, M., Sagayama, S., 2002. Dynamic
661 time-alignment kernel in support vector machine, in: Advances in neural
662 information processing systems, pp. 921–928.
- 663 [26] Soheily-Khah, S., Chouakria, A.D., Gaussier, E., 2015. Progressive and it-
664 erative approaches for time series averaging., in: AALTD@ PKDD/ECML.
- 665 [27] Soheily-Khah, S., Douzal-Chouakria, A., Gaussier, E., 2016. Generalized
666 k-means-based clustering for temporal data under weighted and kernel time
667 warp. Pattern Recognition Letters 75, 63–69.
- 668 [28] Van Nguyen, H., Patel, V.M., Nasrabadi, N.M., Chellappa, R., 2012. Kernel
669 dictionary learning, in: 2012 IEEE International Conference on Acoustics,
670 Speech and Signal Processing (ICASSP), IEEE. pp. 2021–2024.
- 671 [29] Wu, L., Yen, I.E.H., Yi, J., Xu, F., Lei, Q., Witbrock, M., 2018. Random
672 warping series: A random features method for time-series embedding. arXiv
673 preprint arXiv:1809.05259 .
- 674 [30] Yazdi, S.V., Douzal-Chouakria, A., 2018. Time warp invariant ksvd: Sparse
675 coding and dictionary learning for time series under time warp. Pattern
676 Recognition Letters 112, 1–8.

- 677 [31] Yazdi, S.V., Douzal-Chouakria, A., Gallinari, P., Moussallam, M., 2018.
678 Time warp invariant dictionary learning for time series clustering: appli-
679 cation to music data stream analysis, in: Joint European Conference on
680 Machine Learning and Knowledge Discovery in Databases, Springer. pp.
681 356–372.
- 682 [32] Zhu, F., Honeine, P., 2016. Bi-objective nonnegative matrix factorization:
683 Linear versus kernel-based models. IEEE Transactions on Geoscience and
684 Remote Sensing 54, 4012 – 4022. URL: [https://ieeexplore.ieee.org/
685 document/7448928](https://ieeexplore.ieee.org/document/7448928), doi:10.1109/TGRS.2016.2535298.