Paul Honeine and Cédric Richard

Preimage Problem in Kernel-Based Machine Learning

An intimate connection with the dimensionality-reduction problem

© DIGITAL STOCK & LUSPHIX

Kernel machines have gained considerable popularity during the last 15 years, making a breakthrough in nonlinear signal processing and machine learning, thanks to extraordinary advances. This increased interest is undoubtedly driven by the practical goal of being able to easily develop efficient nonlinear algorithms. The key principle behind this, known as the kernel trick, exploits the fact that a great number of data-processing techniques do not explicitly depend on the data itself but rather on a similarity measure between them, i.e., an inner product. To provide a nonlinear extension of these techniques, one can apply a nonlinear transformation to the data, mapping them onto some feature space. According to the kernel trick, this can be achieved by simply replacing the inner product with a reproducing kernel (i.e., positive semidefinite symmetric function), the latter corresponds to an inner product in the feature space. One consequence is that the resulting nonlinear algorithms show significant performance improvements over their linear counterparts with essentially the same computational complexity.

While the nonlinear mapping from the input space to the feature space is central in kernel methods, the reverse mapping from the feature space back to the input space is also of primary interest. This is the case in many applications, including kernel principal component analysis (PCA) for signal and image denoising. Unfortunately, it turns out that the reverse mapping generally does not exist and only a few elements in the feature space have a valid preimage in the input space. The preimage problem consists of finding an approximate solution by identifying data in the input space based on their corresponding features in the high-dimensional feature space. It is essentially a dimensionality-reduction problem, and both have been intimately connected in their historical evolution, as studied in this article.

Digital Object Identifier 10.1109/MSP.2010.939747
Date of publication: 17 February 2011

1053-5888/11/$26.00©2011IEEE

AN INTRODUCTORY EXAMPLE: KERNEL PCA FOR DENOISING

LINEAR DENOISING WITH PCA

In general, some correlations exist among data, thus techniques for dimensionality reduction or the so-called feature extraction provide a way to confine the initial space to a subspace of lower dimensionality. The PCA, also known as Karhunen-Loève transformation, is one of the most widely used dimensionality-reduction techniques. Conventional PCA seeks principal directions that capture the highest variance in the data. Mutually orthonormal, these directions define the subspace, exhibiting information rather than noise, providing the optimal linear transformation. Here, the optimality is in the sense of least-mean-square reconstruction error. For instance, in data compression and manifold learning, much information is conserved by projecting onto the directions of highest variance, while in denoising, directions with small variance are dropped. These schemes are mathematically equivalent, and we use here a denoising schema without loss of generality.

Consider an input space $\mathcal{X}$ endowed by the inner product $\langle \cdot, \cdot \rangle$; for instance, a vectorial space with the Euclidean inner product $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = \boldsymbol{x}_i^\top \boldsymbol{x}_j$. Let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ denote a set of available data (observations) from $\mathcal{X}$. PCA techniques seek the axes that maximize the mean variance of the projected data under the unit-norm constraint, namely, $\psi_1, \psi_2, \ldots, \psi_k$ by maximizing $(1/n) \sum_{i=1}^{n} |\langle \boldsymbol{x}_i, \psi_\ell \rangle|^2$ subject to $\langle \psi_\ell, \psi_{\ell'} \rangle = \delta_{\ell\ell'}$ for all $\ell, \ell' = 1, 2, \ldots, k$. In this expression, the Kronecker delta is defined as $\delta_{\ell\ell'} = 1$ if $\ell = \ell'$, and $\delta_{\ell\ell'} = 0$ otherwise. Solving this constrained optimization problem using the Lagrangian provides the following problem:

$$\lambda_\ell \, \psi_\ell = \boldsymbol{C} \, \psi_\ell, \tag{1}$$

where λ_ℓ defines the amount of variance captured by ψ_ℓ, and C is the covariance matrix of the data. In other words, $(\lambda_\ell, \psi_\ell)$ is the eigenvalue–eigenvector of the covariance matrix, data assumed zero-mean. Furthermore, eigenvectors lie in the span of the data, since for every $\ell = 1, 2, \ldots, k$ we have

$$\psi_\ell = \frac{1}{\lambda_\ell} \boldsymbol{C} \, \psi_\ell = \frac{1}{\lambda_\ell n} \sum_{i=1}^{n} \langle \boldsymbol{x}_i, \psi_\ell \rangle \, \boldsymbol{x}_i.$$

The eigenvectors associated with the largest eigenvalues provide a relevant low-dimensional subspace. As a consequence, we are interested in elements from this relevant subspace. This is the case, for instance, in data denoising, where the projection of a given noisy data onto this subspace provides its noise-free counterpart. Therefore, the latter can be written as an expansion of the eigenvectors, namely, for a noisy data $\tilde{\boldsymbol{x}}$, we get the denoised $\psi = \sum_{i=1}^{k} \langle \tilde{\boldsymbol{x}}, \psi_i \rangle \, \psi_i$, and from the aforementioned expression, as a linear expansion in terms of the available data, by taking the form

$$\psi = \sum_{i=1}^{n} \alpha_i \, \boldsymbol{x}_i.$$

KERNEL PCA FOR NONLINEAR DENOISING

To provide a natural nonlinear extension of PCA, a nonlinear mapping is applied to the data as a preprocessing stage, prior to applying the PCA algorithm. Let $\phi(\,\cdot\,)$ be the nonlinear transformation mapping data from the input space $\mathcal{X}$ to some feature space $\mathcal{H}$. Then problem (1) essentially remains the same, with the covariance matrix associated to the transformed data. From the linear expansion with respect to the latter, the resulting principal axes take the form

$$\psi_\ell = \sum_{i=1}^{n} \langle \phi(\boldsymbol{x}_i), \psi_\ell \rangle_{\mathcal{H}} \, \phi(\boldsymbol{x}_i), \tag{2}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product in the feature space $\mathcal{H}$. In this space, each feature ψ_ℓ lies in the span of the mapped input data, with the coefficients given by the ℓth eigenvector of the eigenproblem

$$n \, \lambda_\ell \, \boldsymbol{\alpha}_\ell = \boldsymbol{K} \, \boldsymbol{\alpha}_\ell, \tag{3}$$

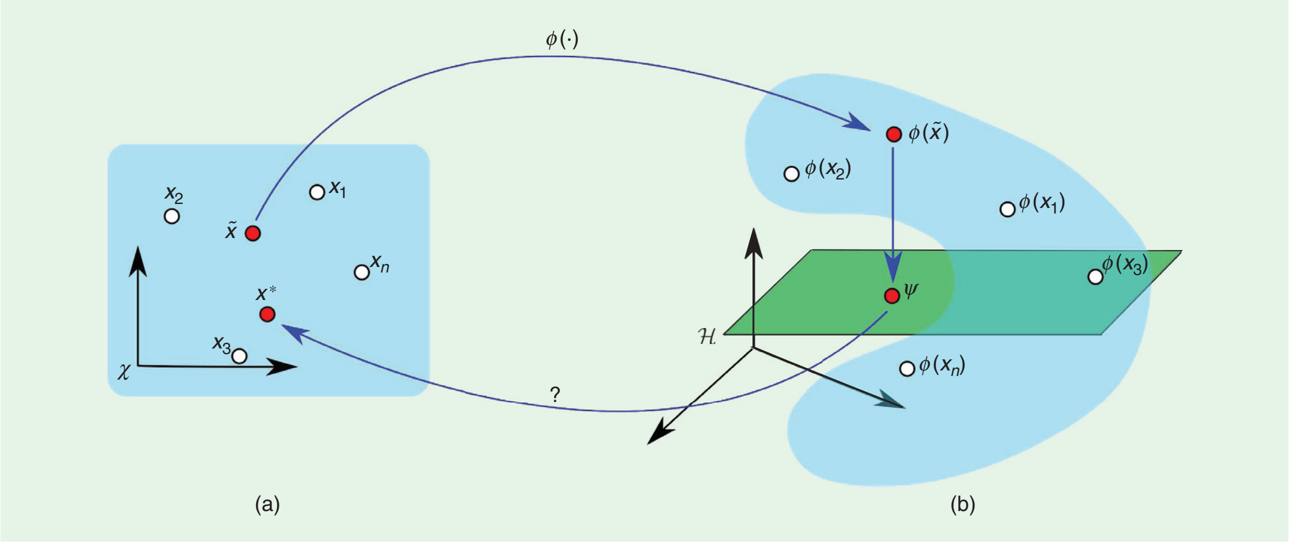
where $\boldsymbol{K}$ is the so-called Gram matrix with entries $\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle_{\mathcal{H}}$, for $i, j = 1, 2, \ldots, n$. As illustrated here, the expansion coefficients require only the evaluation of the inner products. Without the need to exhibit the mapping function, this information can be easily exploited for a large class of nonlinearities by substituting the inner product with a positive semidefinite kernel function. This argument is the kernel trick, which provides a nonlinear counterpart of the classical PCA algorithm, the so-called kernel PCA [1].

Consider the denoising application using kernel PCA. For a given $\tilde{\boldsymbol{x}}$, its nonlinear transformation $\phi(\tilde{\boldsymbol{x}})$ is projected onto the subspace spanned by the most relevant principal axes, providing the denoised pattern. The latter can be written as a linear expansion of the k principal axes, $\psi_1, \psi_2, \ldots, \psi_k$, with

$$\psi = \sum_{i=1}^{k} \langle \boldsymbol{x}, \psi_i \rangle \, \psi_i. \tag{4}$$

Equivalently, the denoised pattern can also be written as a linear expansion of the n images of the training data, namely $\psi = \sum_{i=1}^{n} \alpha_i \, \phi(\boldsymbol{x}_i)$, where the expansion in (2) is used. In practice, one is interested in representing the denoised pattern in the input space, as illustrated in Figure 1. It turns out that most elements of the feature space, including the denoised patterns, are not valid images, i.e., the result of applying the map to some input data. To get the denoised counterpart in the original input space, one needs to operate an approximation scheme, i.e., estimate $\boldsymbol{x}^*$ such that its image $\phi(\boldsymbol{x}^*)$ is as close as possible to ψ.

Beyond this kernel-PCA example, the kernel trick is well known in the machine-learning community. It provides flexibility to derive nonlinear techniques based on linear ones, with the data being implicitly mapped into a feature space. This space is given by the span of the mapped data, i.e., all the linear expansions of mapped data. The price to pay is that, in general, not each element of the space is necessarily the image of some data.



[FIG1] Kernel machines map the input space [blue region in (a)] into a higher-dimensional space [blue region in (b)]. The reproducing kernel Hilbert space (rkHs) $\mathcal{H}$ is defined as the completion of the span of the mapped input data, with elements written as a linear expansion of mapped data. However, not each element of $\mathcal{H}$ is necessarily the image of some input data. The preimage problem consists of going back to the input space, e.g., to represent in the input space elements of the rkHs (e.g., the effect of projecting onto a subspace).

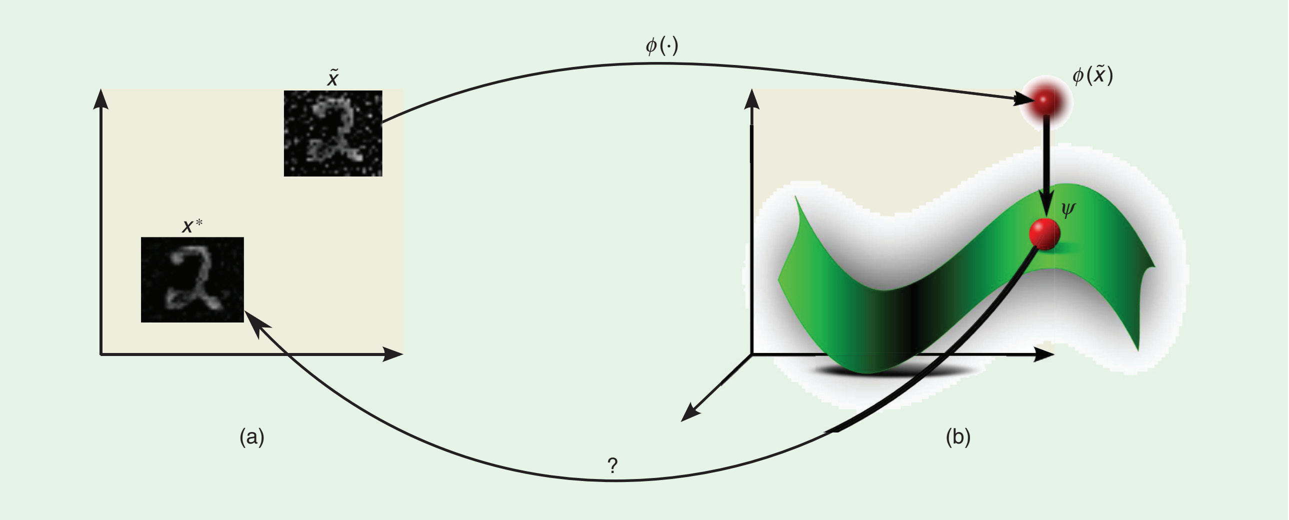
This is the case of most elements in the feature space, since they can be written as

$$\psi = \sum_{i=1}^{n} \alpha_i \, \phi(x_i),$$

as illustrated earlier with either a principal axe ψ_ℓ or a denoised feature ψ. To give proper interpretation for these components, one should define the way back from the feature into the input space. This is the preimage problem in kernel-based machine learning, as illustrated in Figure 2.

KERNEL-BASED MACHINE LEARNING

In the past 15 years or so, a novel breakthrough for artificial neural networks has been achieved in the field of pattern recognition and classification within the framework of kernel-based machine learning. They have gained wide popularity owing to the theoretical guarantees regarding performance and low computational complexity in nonlinear algorithms. Pioneered by Vapnik's support vector machines (SVMs) for classification and regression [2], kernel-based methods are nonlinear algorithms that can be adapted to an extensive class of nonlinearities. As a consequence, they have found numerous applications, including



[FIG2] Schematic illustration of the preimage problem for pattern denoising with kernel PCA. While dimensionality reduction through orthogonal projection is performed in the (b) feature space, a preimage technique is required to recover the denoised pattern in the (a) input space.

classification [3], regression [4], time-series prediction [5], novelty detection [6], image denoising [7], and bioengineering [8], to name just a few (see, e.g., [9] for a review).

REPRODUCING KERNELS AND rkHs

Originally proposed by Aizerman et al. in [10], the kernel trick provides an elegant mathematical means to derive powerful nonlinear variants of classical linear techniques. Most well-known statistical (linear) techniques can be formulated as an inner product between pairs of data. Thus, applying any nonlinear transformation to the data can only impact the values of the resulting inner products. Therefore, one does not need to compute such a transformation explicitly for a large class of nonlinearities. Instead, one only needs to replace the inner product operator with an appropriate kernel, i.e., a symmetric hermitian function. The only restriction is that the latter defines an inner product in some space. A sufficient condition for this is ensured by Mercer's theorem [11], which may be stated as follows: any positive semidefinite kernel can be expressed as an inner product in some space, where the positive semidefiniteness of a kernel $\kappa\colon \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ is determined by the property

$$\sum_{i,j} \alpha_i\,\alpha_j\,\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) \geq 0,$$

for all $\alpha_i, \alpha_j \in \mathbb{R}$ and $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$. Furthermore, the Moore-Aronszajn theorem [12] states that, to any positive semidefinite kernel κ corresponds a unique rkHs, whose inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$, usually called reproducing kernel, is κ itself.

The one-to-one correspondence between rkHs and positive semidefinite functions has proved to be quite useful in numerous fields (see [13] and references therein). Since the pioneering work of Aronszajn [12], reproducing kernels and rkHs formalism have been increasingly used, especially, after being selected for the resolution of interpolation problems by Parzen [14], Kailath [15], and Wahba [16]. An rkHs is a Hilbert space of functions for which point evaluations are bounded and where the existence and uniqueness of the reproducing kernel are guaranteed by the Riesz representation theorem. In fact, let $\mathcal{H}$ be a Hilbert space of functions defined on some compact $\mathcal{X}$, for which the evaluation $\psi(\boldsymbol{x})$ of the function $\psi\in\mathcal{H}$ is bounded for all $\boldsymbol{x}\in\mathcal{X}$. By this theorem, there exists a unique function $\phi(\boldsymbol{x})\in\mathcal{H}$ such as $\psi(\boldsymbol{x}) = \langle\psi,\phi(\boldsymbol{x})\rangle_{\mathcal{H}}$. Also denoted $\kappa(\,\cdot\,,\boldsymbol{x})$, this function has the following popular property:

$$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) = \langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)\rangle_{\mathcal{H}}, \tag{5}$$

for any $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$. Moreover, the distances can be easily evaluated using the kernel trick, since the distance between two elements can be given using only kernel values, with

$$\begin{aligned}\|\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{x}_j)\|_{\mathcal{H}}^2 &= \langle\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{x}_j),\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{x}_j)\rangle_{\mathcal{H}}\\ &= \kappa(\boldsymbol{x}_i,\boldsymbol{x}_i) - 2\,\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) + \kappa(\boldsymbol{x}_j,\boldsymbol{x}_j),\end{aligned} \tag{6}$$

where $\|\cdot\|_{\mathcal{H}}$ denotes the norm in the rkHs.

The inherent modularity of reproducing kernels allows scaling-up linear algorithms into nonlinear ones, adapting kernel-based machines to tackle a large class of nonlinear tasks. Kernels are commonly defined on vectorial spaces, $\mathcal{X}$ endowed with the Euclidean inner product $\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle = \boldsymbol{x}_i^\top\boldsymbol{x}_j$ and the associated norm $\|\boldsymbol{x}_i\|$. They can be easily adapted to operate on images, e.g., in face recognition or image denoising. They are not restricted to vectorial inputs but can be naturally designed to measure similarities between sets, graphs, strings, and text documents [9]. As illustrated in Table 1, most of the kernels used in the machine-learning literature can be divided into two categories: projective kernels are functions of inner product, such as the polynomial kernel, and radial kernels (also known by isotropic kernels) are functions of distance, such as the Gaussian kernel. These kernels implicitly map the data into a high-dimensional space, even infinite dimensional for the latter.

[TABLE 1] COMMONLY USED KERNELS IN MACHINE LEARNING, WITH PARAMETERS $c>0$, $p\in\mathbb{N}_+$, AND $\sigma>0$.

KERNELS	EXPRESSIONS
PROJECTIVE	
MONOMIAL	$(\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle)^p$
POLYNOMIAL	$(c+\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle)^p$
EXPONENTIAL	$\exp(\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle/2\sigma^2)$
SIGMOID (PERCEPTRON)	$\tanh(\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle/\sigma+c)$
RADIAL	
GAUSSIAN	$\exp(-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2/2\sigma^2)$
LAPLACIAN	$\exp(-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|/2\sigma^2)$
MULTIQUADRATIC	$\sqrt{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2+c}$
INVERSE MULTIQUADRATIC	$1/\sqrt{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2+c}$

THE REPRESENTER THEOREM

In machine learning, inferences are focused on the estimation of the structure of some data, based on a set of available data. Given n observations, $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n$, and eventually the corresponding labels, $y_1,y_2,\ldots,y_n$, one seeks a function that minimizes a fitness error over the data, with some control of its complexity (i.e., functional norm). To this end, we consider the rkHs associated to the reproducing kernel as the hypothesis space from which the optimal is determined. The rkHs associated to κ can be identified, modulo certain details, with a space of functions defined by a linear combination of the functions $\phi(\boldsymbol{x}_1),\ \phi(\boldsymbol{x}_2),\ldots,\phi(\boldsymbol{x}_n)$. Its flexibility efficiently allows for solving optimization problems, owing to the (generalized) representer theorem. Originally derived by Kimeldorf and Wahba for splines in [17], it was recently generalized to kernel-based machine learning in [18], including SVM and kernel PCA, as follows in Theorem 1.

THEOREM 1 (REPRESENTER THEOREM)

For any function $\psi\in\mathcal{H}$ minimizing a regularized cost function of the form

$$\sum_{i=1}^{n} f\big(y_i,\psi(\boldsymbol{x}_i)\big) + \eta\, g(\|\psi\|_{\mathcal{H}}^2),$$

with $f(\cdot,\cdot)$ some loss function and $g(\cdot)$ a strictly monotonic increasing function on $\mathbb{R}_+$, can be written as an image expansion in terms of the available data, namely,

$$\psi = \sum_{i=1}^{n} \alpha_i \, \phi(\boldsymbol{x}_i). \tag{7}$$

This theorem shows that, even in an infinite-dimensional rkHs, one only needs to work in the subspace spanned by the n images of the training data.

Before we proceed further, we examine the effectiveness of this theorem on two machine-learning techniques: first, consider the kernel PCA where the projected variance is maximized, namely $\psi_1, \psi_2, \ldots, \psi_k = \arg\max_\psi (1/n) \sum_i |\langle \boldsymbol{x}_i, \psi \rangle|^2$, under the orthonormality constraint $\langle \psi_\ell, \psi_{\ell'} \rangle_{\mathcal{H}} = \delta_{\ell\ell'}$ for all $\ell, \ell' = 1, 2, \ldots, k$. As derived in the introductory example, one only needs to solve the eigenproblem (3), involving only n unknowns for each principal axes. These unknowns correspond to the weighting coefficients in the expansion (7). Second, we consider a regression problem known as ridge regression. In this case, the mean squared error is minimized, with

$$\min_{\psi} \frac{1}{n} \sum_{i=1}^{n} |y_i - \psi(\boldsymbol{x}_i)|^2 + \eta \|\psi\|^2_{\mathcal{H}}, \tag{8}$$

where the first term is the fitness error while the second one controls the complexity of the solution (known as Tikhonov regularization). By substituting (7) into (8), we get the optimization problem

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|^2 + \eta \, \boldsymbol{\alpha}^\top \boldsymbol{K} \boldsymbol{\alpha},$$

with $\boldsymbol{\alpha} = [\alpha_1 \, \alpha_2 \, \cdots \, \alpha_n]^\top$ and $\boldsymbol{y} = [y_1 \, y_2 \, \cdots \, y_n]^\top$. The optimal weighting coefficients are obtained by solving the linear system

$$(\boldsymbol{K} + \eta \boldsymbol{I}) \, \boldsymbol{\alpha} = \boldsymbol{y}, \tag{9}$$

where $\boldsymbol{I}$ is the identity matrix.

Such models as a sum of basis functions have been extensively studied in the literature, for instance, in interpolation problems [19], and more recently, in machine learning [20]. To illustrate this theorem, take for instance, the Gaussian kernel investigated in [21] for interpolation in two dimensions (2-D). For this kernel, we can think about the map $\phi(\boldsymbol{x}_i) \colon \boldsymbol{x}_i \mapsto \exp(-\|\cdot - \boldsymbol{x}_i\|^2 / 2\sigma^2)$ that transforms each input data into a Gaussian bump centered on that point. Clearly, the representer theorem (Theorem 1) states that the optimal solution is a linear combination of Gaussians centered on the available input data. However, it is well known that the sum of Gaussians centered at different points cannot be written as a single Gaussian. Thus, the solution ψ in (7) cannot be a Gaussian sitting on some arbitrary data; in other words, it is not a valid image of some $\boldsymbol{x} \in \mathcal{X}$, using the map $\phi(\cdot)$ associated to the Gaussian kernel. Finding an input $\boldsymbol{x}^*$ whose image can approximate the function ψ is the preimage problem.

SOLVING THE PREIMAGE PROBLEM

A problem is ill posed if at least one of the following three conditions that characterize the well-posed problems in the sense of Hadamard is violated: 1) a solution exists, 2) it is unique, and 3) it continuously depends on the data (also known as stability condition). Unfortunately, identifying the preimage is generally an ill-posed problem. This is an outcome of the higher dimensionality of the feature space compared with the input space. As a consequence, most elements of ψ in the rkHs might not have a preimage in the input space, i.e., there may not exist an $\boldsymbol{x}^*$ such that $\phi(\boldsymbol{x}^*) = \psi$. Moreover, even if $\boldsymbol{x}^*$ exists, it may not be unique. To circumvent this difficulty, one seeks an approximate solution, i.e., $\boldsymbol{x}^*$ whose map $\phi(\boldsymbol{x}^*)$ is as close as possible to ψ.

Consider a pattern ψ in the feature space $\mathcal{H}$, obtained by any kernel-based machine, e.g., a principal axe or a denoised pattern obtained from kernel PCA. By virtue of Theorem 1, let $\psi = \sum_{i=1}^{n} \alpha_i \, \phi(\boldsymbol{x}_i)$. The preimage problem consists of the following optimization problem

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \left\| \sum_{i=1}^{n} \alpha_i \, \phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}) \right\|^2_{\mathcal{H}}. \tag{10}$$

Equivalently, from the kernel trick, $\boldsymbol{x}^*$ minimizes the objective function

$$\Xi(\boldsymbol{x}) = \kappa(\boldsymbol{x}, \boldsymbol{x}) - 2 \sum_{i=1}^{n} \alpha_i \, \kappa(\boldsymbol{x}, \boldsymbol{x}_i), \tag{11}$$

where the term independent of $\boldsymbol{x}$ has been dropped.

As opposed to this functional formalism, one may also adopt a vectorwise representation, with elements in the rkHs given by their coordinates with respect to an orthogonal basis. Taking, for instance, the basis defined by the kernel PCA, as given in (4), each $\psi \in \mathcal{H}$ is represented vectorwise with $[\langle \psi, \psi_1 \rangle \, \langle \psi, \psi_2 \rangle \, \cdots \, \langle \psi, \psi_k \rangle]^\top$, thus defining a k-dimensional representation. In such a case, the Euclidean distance between the latter and the one obtained from the image of $\boldsymbol{x}^*$ is minimized. This is essentially a classical dimensionality-reduction problem, connecting the preimage problem to the historical evolution of dimensionality-reduction techniques. This is emphasized next, providing a survey on a large variety of methods.

THE EXACT PREIMAGE WHEN IT EXISTS

Suppose that there exists an exact preimage of ψ, i.e., $\boldsymbol{x}^*$ such that $\phi(\boldsymbol{x}^*) = \psi$, then the optimization problem in (10) results into that preimage. Furthermore, the preimage can be easily computed when the kernel is an invertible function of $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$, such as some projective kernels including the polynomial kernel with odd degree and the sigmoid kernel (see Table 1). Let $h \colon \mathbb{R} \to \mathbb{R}$ define the inverse function such that $h(\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$. Then, given any orthonormal basis in the input space $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N\}$, every element $\boldsymbol{x} \in \mathcal{X}$ can be written as

$$\boldsymbol{x} = \sum_{j=1}^{N} \langle \boldsymbol{e}_j, \boldsymbol{x} \rangle \, \boldsymbol{e}_j = \sum_{j=1}^{N} h\big(\kappa(\boldsymbol{e}_j, \boldsymbol{x})\big) \, \boldsymbol{e}_j.$$

As a consequence, the exact preimage x^* of some pattern $\psi = \sum_{i=1}^{n} \alpha_i \, \phi(x_i)$, namely $\phi(x^*) = \psi$, can be expanded as

$$x^* = \sum_{j=1}^{N} h\Big(\sum_{i=1}^{n} \alpha_i \, \kappa(e_j, x_i) \Big) \, e_j.$$

THE STRUCTURE OF THE KERNEL FUNCTIONS PROVIDES USEFUL INSIGHTS TO DERIVE MORE APPROPRIATE OPTIMIZATION TECHNIQUES.

Likewise, when the kernel is an invertible function of the distance, such as radial kernels, a similar expression can be derived by using the polarization identity $4 \langle x^*, e_j \rangle = \|x^* + e_j\|^2 - \|x^* - e_j\|^2$ [22].

Clearly, such a simple derivation for the preimage is only valid under the crucial assumption that the preimage x^* exists. Unfortunately, for a large class of kernels, there are no exact preimages. Rather than seeking the exact preimage, we consider an approximate preimage by solving the optimization problem in (10). In what follows, we present several strategies for solving this problem. We first review the techniques based on classical optimization schemes and then present learning-based techniques by incorporating additional prior information.

GRADIENT DESCENT TECHNIQUES

Gradient descent is one of the simplest optimization techniques. It requires computing the gradient of the objective function (11), denoted as $\nabla_x \Xi(x^*)$. In its simplest form, the current guess x_t^* is updated into x_{t+1}^* by stepping into the direction opposite to the gradient, with

$$x_{t+1}^* = x_t^* - \eta_t \, \nabla_x \, \Xi(x_t^*),$$

where η_t is a step size parameter, often optimized using a line-search procedure. As an alternative to the gradient descent, one may use more sophisticated techniques, such as Newton's method. Unfortunately, the objective function is inherently nonlinear and clearly nonconvex. Thus, a gradient descent algorithm must be run many times with several starting values, hoping that a feasible solution will be among the local minima obtained over the runs.

FIXED-POINT ITERATION METHOD

The structure of the kernel functions provides useful insights to derive more appropriate optimization techniques beyond classical gradient descent. More precisely, the gradient of expression (11) has a closed-form expression for most kernels. By setting this expression to zero, this greatly simplifies the optimization scheme, resulting in a fixed-point iterative technique. Taking for instance the Gaussian kernel [7], the objective function in (11) becomes

$$-2 \sum_{i=1}^{n} \alpha_i \, \exp\left(- \|x - x_i\|^2 / 2\sigma^2 \right),$$

with its gradient

$$\nabla_x \Xi(x) = - \frac{2}{\sigma^2} \sum_{i=1}^{n} \alpha_i \, \exp\left(- \|x - x_i\|^2 / 2\sigma^2 \right) (x - x_i).$$

We get the preimage by setting this gradient to zero, which results in a fixed-point iterative expression

$$x_{t+1}^* = \frac{\sum_{i=1}^{n} \alpha_i \, \kappa(x_t^*, x_i) \, x_i}{\sum_{i=1}^{n} \alpha_i \, \kappa(x_t^*, x_i)},$$

with $\kappa(x_t^*, x_i) = \exp\left(- \|x_t^* - x_i\|^2 / 2\sigma^2 \right)$. Similar expressions can be derived for most kernels, such as the polynomial kernel of degree p [23] with

$$x_{t+1}^* = \sum_{i=1}^{n} \alpha_i \left(\frac{\langle x_t^*, \, x_i \rangle + c}{\langle x_t^*, \, x_t^* \rangle + c} \right)^{p-1} x_i.$$

Unfortunately, the fixed-point iterative technique still suffers from local minima and tends to be unstable. The numerical instability especially occurs when the value of the denominator decreases to zero. To prevent this situation, a regularized solution can be easily formulated, as studied in [24].

An interesting fact about the fixed-point iterative method is that the resulting preimage lies in the span of the available data, taking the form $x^* = \sum_i \beta_i x_i$ for some coefficients $\beta_1, \beta_2, \ldots, \beta_n$ to be determined. Thus, the search space is controlled, as opposed to gradient-descent techniques that explore the entire space. We further exploit information from available training data and their mapped counterparts, as discussed later.

LEARNING THE PREIMAGE MAP

To find the preimage map, a learning machine is constructed, with training elements from the feature space and estimated values in the input space, as follows: we seek to estimate a function Γ^* with the property that $\Gamma^*(\phi(x_i)) = x_i$, for $i = 1, 2, \ldots, n$. Then, ideally, $\Gamma^*(\psi)$ should give x^*, the preimage of ψ. To make the problem computationally tractable, two issues are considered in [25] and [26]. First, the function is defined on a vector space. This can be done by representing vectorwise any $\psi \in \mathcal{H}$ with $[\langle \psi, \psi_1 \rangle \; \langle \psi, \psi_2 \rangle \; \cdots \; \langle \psi, \psi_k \rangle]^\top$, using an orthogonal basis obtained from kernel PCA. Second, the preimage map Γ^* is decomposed into $\dim(\mathcal{X})$ functions to estimate each component of x^*. From these considerations, we seek functions $\Gamma_1^*, \Gamma_2^*, \ldots, \Gamma_{\dim(\mathcal{X})}^*$, with $\Gamma_m^* \colon \mathbb{R}^k \to \mathbb{R}$. Each of these functions is obtained by solving the optimization problem

$$\Gamma_m^* = \arg\min_{\Gamma} \sum_{i=1}^{n} f([x_i]_m, \Gamma(\psi)) + \eta \, g(\|\Gamma\|^2),$$

where $f(\cdot\,, \cdot)$ is some loss function and $[\cdot]_m$ denotes the mth component operator. By taking for instance the distance as a loss function, we get

$$\Gamma_m^* = \arg\min_{\Gamma} \frac{1}{n} \sum_{i=1}^{n} \big| [x_i]_m - \Gamma(\psi) \big|^2 + \eta \, \|\Gamma\|^2.$$

This optimization problem can be easily solved by a matrix-inversion scheme in analogy to the ridge-regression problem (8) and its linear system (9). This learning approach is further investigated in the literature, incorporating neighborhood information [27] and regularization with a penalized learning [28]. All these methods are based on a set of available data in the input space and the associated images in the rkHs. The method discussed next carries this concept further by exploring pairwise distances in both spaces.

AN INTERESTING FACT ABOUT FIXED-POINT ITERATIVE METHOD IS THAT THE RESULTING PREIMAGE LIES IN THE SPAN OF THE AVAILABLE DATA.

MULTIDIMENSIONAL SCALING-BASED TECHNIQUES

As illustrated in the earlier preimage-learning approach, the preimage map seeks data in the input space based on their associated images in the rkHs. Essentially, this is a low-dimensional embedding of objects from a high-dimensional space. This problem has received a lot of attention in multivariate statistics under the framework of multidimensional scaling (MDS) [29]. MDS techniques mainly embed data in a low-dimensional space by preserving pairwise distances. This approach has been applied with success to solve the preimage problem [23]. Consider each distance in the rkHs $\delta_i = \|\psi - \phi(\boldsymbol{x}_i)\|_{\mathcal{H}}$ and its counterpart in the input space $\|\boldsymbol{x}^* - \boldsymbol{x}_i\|$. Ideally, these distances are preserved, namely

$$\|\boldsymbol{x}^* - \boldsymbol{x}_i\|^2 = \|\psi - \phi(\boldsymbol{x}_i)\|_{\mathcal{H}}^2, \tag{12}$$

for every $i = 1, 2, \ldots, n$. It is easy to verify that if there exists an i such that $\psi = \phi(\boldsymbol{x}_i)$, then we get the preimage $\boldsymbol{x}^* = \boldsymbol{x}_i$ (Figure 3).

One way to solve this problem is to minimize the mean-square error between these distances, with

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \sum_{i=1}^{n} \left| \|\boldsymbol{x} - \boldsymbol{x}_i\|^2 - \|\psi - \phi(\boldsymbol{x}_i)\|_{\mathcal{H}}^2 \right|^2.$$

To solve this optimization problem, a fixed-point iteration method is proposed by setting the gradient of the aforementioned expression to zero, resulting in the expression

$$\boldsymbol{x}^* = \frac{\sum_{i=1}^{n} (\|\boldsymbol{x}^* - \boldsymbol{x}_i\|^2 - \delta_i^2)\,\boldsymbol{x}_i}{\sum_{i=1}^{n} (\|\boldsymbol{x}^* - \boldsymbol{x}_i\|^2 - \delta_i^2)}.$$

Another approach to solve this problem is to separately consider the identities (12), resulting in n equations

$$2\langle \boldsymbol{x}^*, \boldsymbol{x}_i \rangle = \langle \boldsymbol{x}^*, \boldsymbol{x}^* \rangle + \langle \boldsymbol{x}_i, \boldsymbol{x}_i \rangle - \delta_i^2,$$

for $i = 1, 2, \ldots, n$. In these expressions, the unknown also appears on the right-hand side, with $\langle \boldsymbol{x}^*, \boldsymbol{x}^* \rangle$. This unknown quantity can be easily identified in the case of centered data, since taking the average of both sides results in

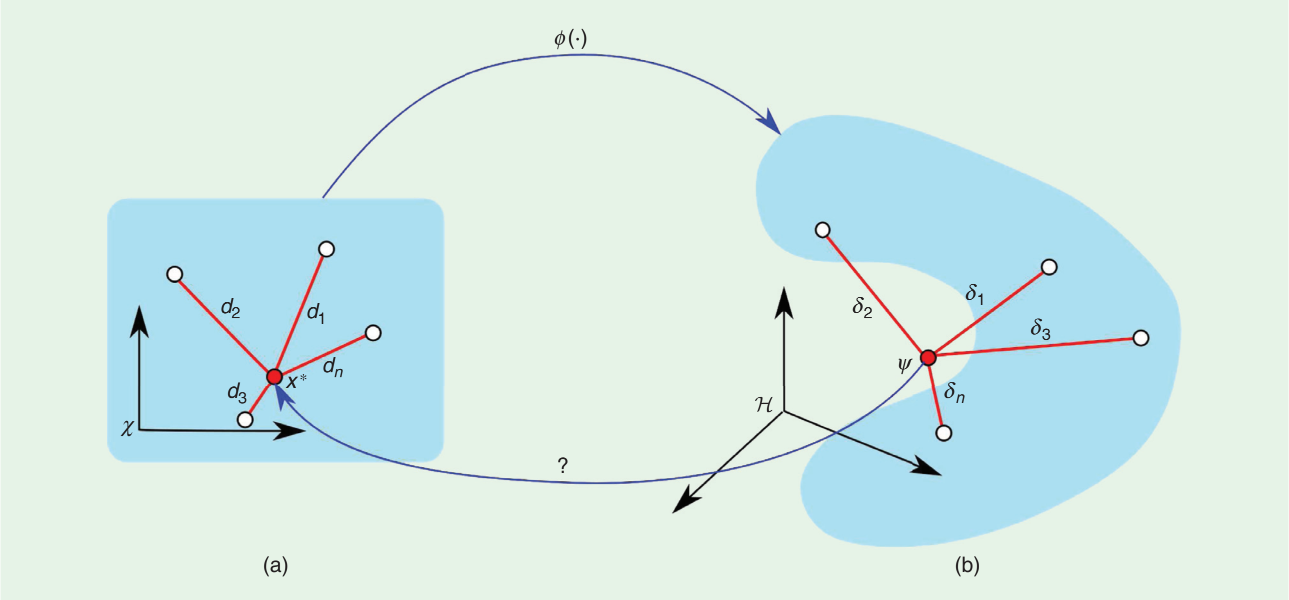
$$\langle \boldsymbol{x}^*, \boldsymbol{x}^* \rangle = \frac{1}{n} \sum_{i=1}^{n} \left(\delta_i^2 - \langle \boldsymbol{x}_i, \boldsymbol{x}_i \rangle \right).$$

Let ϵ be the vector having all its entries equal to $(1/n)\sum_{i=1}^{n} (\delta_i^2 - \langle \boldsymbol{x}_i, \boldsymbol{x}_i \rangle)$ then, in matrix form, we have

$$2X^\top \boldsymbol{x}^* = \operatorname{diag}(X^\top X) - \left[\delta_1^2\,\delta_2^2\,\cdots\,\delta_n^2\right]^\top + \epsilon,$$

where $X = [\boldsymbol{x}_1\,\boldsymbol{x}_2\,\cdots\,\boldsymbol{x}_n]$ and $\operatorname{diag}(\,\cdot\,)$ is the diagonal operator with $\operatorname{diag}(X^\top X)$ the column vector with entries $\langle \boldsymbol{x}_i, \boldsymbol{x}_i \rangle$. The unknown preimage is obtained using the least-squares solution, namely

$$\boldsymbol{x}^* = \frac{1}{2}(XX^\top)^{-1} X \left(\operatorname{diag}(X^\top X) - [\delta_1^2\,\delta_2^2\,\cdots\,\delta_n^2]^\top \right),$$



[FIG3] Schematic illustration of the MDS-based technique where the preimage is identified from pairwise distances in both (a) input and (b) feature spaces.

where the term $(XX^\top)^{-1}X\epsilon$ becomes zero, thanks to the assumption of centered data.

To keep this technique tractable in practice, only a certain neighborhood is considered in the preimage estimation, in the same spirit as the locally linear embedding scheme in dimensionality reduction [30]. This approach opened the door to a range of other techniques, borrowed from dimensionality reduction and manifold learning literature [31].

CONFORMAL MAP APPROACH

In addition to the distance-preserving method of MDS, one may also propose a preimage method by preserving inner product measures. Using such a strategy, the angular measure is also preserved, since $x_i^\top x_j / \|x_i\| \|x_j\|$ defines the cosine of the angle between x_i and x_j in the Euclidean input space. For this reason it is called the conformal map approach. A recent technique to solve the preimage problem based on the conformal map has been presented in [32]. To this end, a coordinate system in the rkHs is constructed with an isometry with respect to the input space. We emphasize the fact that the model is not coupled with any constraint on the coordinate functions, as opposed to the orthogonality between the functions resulting from the kernel PCA.

By virtue of Theorem 1, each of the n coordinate functions can be written as a linear expansion of the available images, namely $\Psi_\ell = \sum_{i=1}^n \theta_{\ell,i}\, \phi(x_i)$, for $\ell = 1, 2, \ldots, n$, with unknown weights to be determined, rearranged in a matrix Θ. Therefore, the coordinates of any element of the rkHs can be obtained by a projection onto these coordinate functions, thus any $\phi(x_i)$ can be represented with the n coordinates in $\boldsymbol{\Psi}_{x_i} = [\langle \Psi_1, \phi(x_i)\rangle \; \langle \Psi_2, \phi(x_i)\rangle \; \cdots \; \langle \Psi_k, \phi(x_i)\rangle]^\top$. Ideally, the inner products are preserved in both coordinate system and Euclidean input space, specifically

$$\boldsymbol{\Psi}_{x_i}^\top \boldsymbol{\Psi}_{x_j} = x_i^\top x_j, \tag{13}$$

for all $i, j = 1, 2, \ldots, n$. This can be solved by minimizing the fitness error over all pairs,

$$\min_{\Psi_1, \ldots, \Psi_n} \sum_{i,j=1}^n \left| x_i^\top x_j - \boldsymbol{\Psi}_{x_i}^\top \boldsymbol{\Psi}_{x_j} \right|^2 + \eta \sum_{\ell=1}^n \|\Psi_\ell\|_{\mathcal{H}}^2,$$

where the second term incorporates regularization. This can be written in matrix form as

$$\min_{\boldsymbol{\Theta}} \frac{1}{2} \|X^\top X - K\, \boldsymbol{\Theta}^\top \boldsymbol{\Theta} K\|_F^2 + \eta\, \mathrm{tr}(\boldsymbol{\Theta}^\top \boldsymbol{\Theta} K),$$

where $\mathrm{tr}(\,\cdot\,)$ denotes the trace of a matrix and $\|\cdot\|_F$ the Frobenius norm, i.e., the root of sum of squared (absolute) values of all its elements, or equivalently $\|M\|_F^2 = \mathrm{tr}(M^\top M)$. By taking the derivative of this expression with respect to $\boldsymbol{\Theta}^\top \boldsymbol{\Theta}$, one obtains

$$\boldsymbol{\Theta}^\top \boldsymbol{\Theta} = K^{-1}\left(X^\top X - \eta K^{-1}\right) K^{-1}. \tag{14}$$

Now we are in a position to determine the preimage of some $\psi = \sum_{i=1}^n \alpha_i\, \phi(x_i)$. Its coordinates associated to the system of coordinate functions $\Psi_1, \Psi_2, \ldots, \Psi_n$ are given by

$$\langle \psi, \Psi_\ell \rangle_{\mathcal{H}} = \sum_{i,j=1}^n \theta_{\ell,i}\, \alpha_j\, \kappa(x_i, x_j),$$

for $\ell = 1, 2, \ldots, n$. By preserving the inner products in both spaces, ideally the model in (13) can be extended to ψ, resulting in

$$X^\top x^* = K \boldsymbol{\Theta}^\top \boldsymbol{\Theta} K \boldsymbol{\alpha}.$$

By combining this expression with (14), we get the simplified expression $X^\top x^* = (X^\top X - \eta K^{-1})\,\boldsymbol{\alpha}$, whose least square solution is

$$x^* = (XX^\top)^{-1} X (X^\top X - \eta K^{-1})\,\boldsymbol{\alpha}.$$

It is worth noting that this expression is independent of the kernel type under investigation.

Furthermore, this technique can be easily extended to identify the preimages of a set of elements in rkHs, since the term between parentheses needs to be computed only once. In fact, this is a matrix-completion scheme like the one studied in [33]. This corresponds to completing an inner-product matrix based on another Gram matrix, the matrix of kernel values.

SCOPE OF APPLICATION OF THE PREIMAGE PROBLEM

In this section, we present some application examples that involve solving the preimage problem. Our first experiments are with kernel PCA on toy data and are mainly intended to illustrate the preimage problem. Then we provide a comparative study of the several methods presented in this article on image denoising problem. Finally, we show how the preimage can be required in other applications beyond kernel PCA. To this end, we consider a problem of autolocalization of sensors in wireless sensor networks.

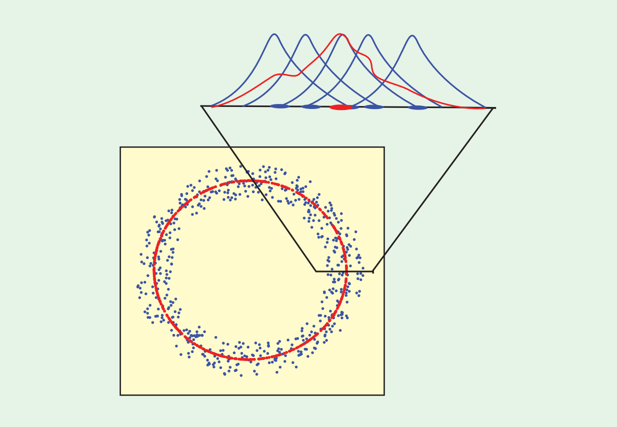
SOME APPLICATIONS OF KERNEL PCA WITH PREIMAGE

FEATURE EXTRACTION

The first illustration considered here is the use of kernel PCA on synthetic data to provide a visual illustration of PCA versus kernel PCA for feature extraction. The data distribution takes the form of a ring in 2-D, with an inner diameter of two and an outer diameter of three. Within this region, $n = 600$ training data were generated, as illustrated in Figure 4 with blue dots. To extract the most relevant feature, two methods were used: on the one hand, the conventional PCA and on the other hand kernel PCA with a preimage step. The PCA technique provided linear axes by solving the eigenvector problem and thus did not capture the circular shape of the data. This is illustrated by projecting the data onto the first principal axis, given by red curve in Figure 4(a). The kernel PCA was applied using a Gaussian kernel with bandwidth $\sigma = 2$, the principal axes being defined by a sum of n Gaussian functions in an infinite-dimensional feature space. A preimage method was required to derive the axes, or representations of these axes, within the input space. As shown in Figure 4(b), this technique captured the nonlinear feature in the original space.

As described at the beginning of this article, when we introduced the preimage problem with the Gaussian kernel, each data is mapped into a Gaussian bump centered around it. By taking the sum of these Gaussians with some optimized weighting coefficients, we get the principal distribution whose mean, if it exists, provides the preimage. It is worth noting that the definition of a mean only exists and makes sense for Gaussian like curves and not for a sum of Gaussians centered at different points. A schematic illustration of the preimage problem is given in Figure 5, taking only a (unidirectional) radial cut in the ring-distributed data. The data obtained by solving the preimage problem can be interpreted as the center of the distribution Gaussian that best approximates the sum of Gaussians.

In this application, a fixed-point iterative technique was used. Next, we give a comparative study of several techniques given in this article by considering the image-denoising problem.



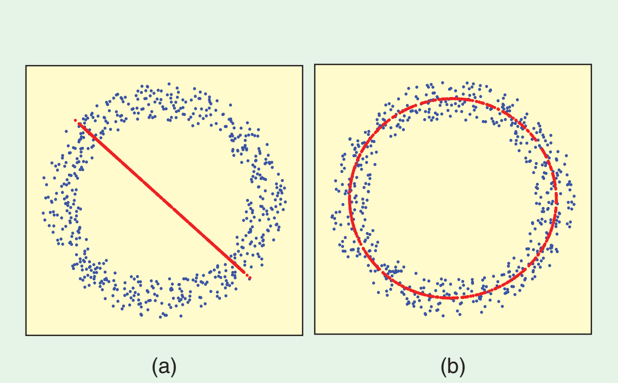
[FIG5] Schematic illustration of the preimage problem with the Gaussian kernel, where the profile corresponds to a radial cut in the ring-distributed data. From the sum of Gaussians (red curve), the preimage corresponds to the mean value of the distribution (blue dots).

IMAGE DENOISING

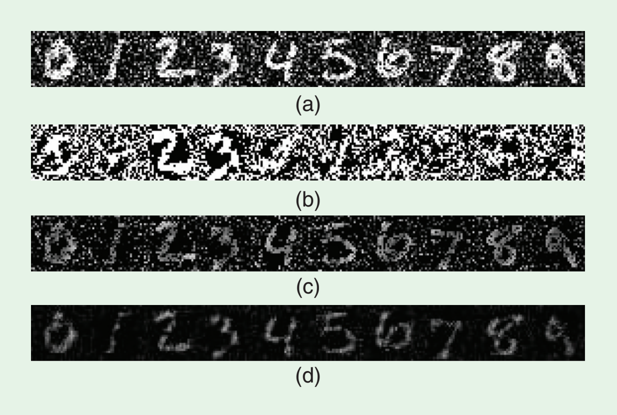
In this section, we illustrate the results obtained in a problem of real-image denoising, using three techniques: the fixed-point iterative method, MDS-based technique, and conformal map approach. The images consist of the modified National Institute of Standards and Technology (NIST) database of handwritten digits [34], corresponding to handwritten digits, from 0 to 9, in (almost) binary 28-by-28 pixels. From a machine-learning point of view, each image can be represented as a point in a 28×28 dimensional space. The original images were corrupted by adding a zero-mean white Gaussian noise with variance 0.2. In the training stage, a set of 1,000 images, 100 of each digit, were used to train the kernel PCA, retaining only 100 leading principal axes. We used the Gaussian kernel for the three algorithms, with the bandwidth set to $\sigma = 10^5$.

To illustrate the ability of this method for image denoising, another set of ten images, one for each digit, was considered under the same noise conditions. These images are illustrated in Figure 6(a), with the results obtained with the (b) fixed-point iterative, (c) MDS-based, and (d) conformal methods. For such applications, the fixed-point iterative algorithm was found to be inappropriate, even with a large number of iterations (here 10,000 iterations were used). To take advantage of prior knowledge, the same training data set was used for learning the reverse map. Realistic results were obtained using the MDS-based method. It is obvious that the conformal algorithm achieved better denoised results. For this simulation, the regularization parameter was set to $\eta = 10^{-9}$.

In an attempt to provide a measure of computational requirements, we considered the (average) total CPU time of each algorithm. These algorithms were implemented on a MATLAB running on a MacBook Pro Duo Core to offer a comparative study. With 10,000 iterations, the fixed-point iterative algorithm required a total CPU time of up to 1 h. The MDS-based and conformal algorithms required 5 min and 1.5 s, respectively.



[FIG4] Denoising data distributed on a ring, using (a) classical PCA and (b) kernel PCA with preimage. The extracted feature is (a) linear and (b) circular.



[FIG6] Application to handwritten digit denoising with kernel PCA, using several preimage methods presented in this article. (a) Ten digits corrupted by noise. (b) Fixed-point iterative method. (c) MDS-based method. (d) Conformal method.

AUTOLOCALIZATION IN WIRELESS SENSOR NETWORKS

With recent technological advances in both electronics and wireless communications, low-power and low-cost tiny sensors have been developed for monitoring physical phenomena and tracking applications. Densely deployed in the inspected environment with efficiently designed distributed algorithms, wireless ad hoc networks seem to offer several opportunities. They were successfully employed in many situations, ranging from military applications such as battleground supervision to civilian applications such as habitat monitoring and healthcare surveillance (see [35], [36] and references therein). While these sensors are often randomly deployed, e.g., for monitoring inhospitable habitats and disaster areas, information captured by each sensor remains obsolete as long as it stays unaware of its location. Implementing a self-localization device, such as a global positioning system receiver, at each sensor device may be too expensive and too power hungry for the desired application with battery-powered devices. As a consequence, only a small fraction of the sensors may be location aware, the so-called anchors or beacons. The other sensors have to estimate their locations by exchanging some information with its neighbors.

For this purpose, each sensor determines a ranging (distance) with other sensors, from intersensor measurements such as the received signal strength indication (RSSI), the connectivity, the hop count, and the time difference of arrival. Most methods used for autolocalization in sensor networks are based on either MDS techniques or semidefinite programming (for a survey, see [37] and [38]), identifying a function that links the ranging between sensors to their locations. However, if the data are not intersensor distances or are linked to coordinates by an unknown nonlinear function, e.g., using the RSSI measurements or the estimated covariance sensor data [39], linear techniques such as MDS and PCA fail to accurately estimate the locations. Once again, the kernel machines provide an elegant way to overcome this drawback.

Here, we describe the method proposed in [40]. The main idea can be described in three stages. In the first stage, we construct the reproducing kernel and its associated rkHs that best describes the anchor pairwise similarities. In the second stage, a nonlinear manifold is designed from similarities between anchor–sensor measurements by applying the kernel PCA technique. The final stage consists of estimating the coordinates of nonanchor sensors by applying a preimage technique on their projections onto the manifold. Next, we describe these three stages before presenting the experimental results.

Consider a network of N sensor nodes, with n location-aware anchors and $N-n$ sensors of unknown location, living in a p-dimensional space, e.g., $p=2$ for localization in a plane. Let $\boldsymbol{x}_i \in \mathbb{R}^p$ be the coordinates of the ith sensor, rearranged such that indices $i = 1, 2, \ldots, n$ correspond to anchors. Let $\tilde{K}(i,j)$ be the intersensor similarity between sensors i and j, such as RSSI.

KERNEL SELECTION FROM INTERANCHOR SIMILARITIES

As a model of similarity measurements, the appropriate reproducing kernel should be chosen and tuned up, which allows a physical meaning of the results obtained from the kernel PCA (next stage). The alignment criterion [41] provides a measure of similarity between the reproducing kernel and target function, e.g., between a Gaussian kernel and RSSI measurements. Maximizing the alignment $\mathcal{A}(K, \tilde{K})$ provides the optimal-reproducing kernel, faithful to the interanchor measurements, where

$$\mathcal{A}(K, \tilde{K}) = \frac{\langle K, \tilde{K} \rangle_F}{\sqrt{\langle K, K \rangle_F \langle \tilde{K}, \tilde{K} \rangle_F}},$$

with $\langle \cdot, \cdot \rangle_F$ as the Frobenius inner product between two matrices. Taking, for instance, the Gaussian kernel, the optimization problem is reduced to finding the optimal bandwidth. In practice, this optimization problem is solved at each anchor, using only the information from its neighborhood.

KERNEL PCA UPON ANCHORS

After identifying the reproducing kernel adapted to the measurements, a kernel-PCA approach is applied to provide the most relevant subspace of the associated rkHs. Classical kernel PCA is computed by a diagonalization scheme, which may be computationally expensive for in-network processing. An alternative approach can be done using an iterative scheme, such as the kernel-Hebbian algorithm [42] (we refer the reader to [40] for its implementation in wireless sensor networks).

PREIMAGE FOR LOCATION ESTIMATION

For each sensor, we represent its image in the rkHs associated to the kernel, maximizing the alignment criterion. The image is projected onto the manifold, obtained using kernel PCA with anchor pairwise similarities. The problem of estimating the coordinates from that representation is the preimage problem.

EXPERIMENTAL RESULTS

The first batch of experiments was carried out on simulated measurements. For this purpose, we considered a network of sensors measuring some physical phenomena, e.g., temperature, atmospheric pressure, or luminance. In a static field, we assumed that measurements were jointly generated from a normal distribution, with decreasing correlations between measurements as a function of the distance between sensors. This information was used as a local similarity measure between sensors [39]. More precisely, we considered the spherical model, commonly used in environmental and geological sciences [43], defined by a covariance of the form $\zeta(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$ with

$$\zeta(u) = \begin{cases} 1 - \frac{3}{2d}u + \frac{1}{2d^3}u^3 & \text{for } 0 \le u \le d; \\ 0 & \text{for } d < u, \end{cases}$$

where d denotes the cutoff distance, and fixed to $d = 60$ in our experiments. The profile of the spherical model is illustrated in Figure 7. The experiments consisted of 100 sensors, from which 20 were anchors with known locations, randomly spread over a 100-by-100 square region. For each sensor, 200 measurements were collected, and the Gaussian kernel was considered. Figure 8 illustrates the localization results obtained with this method.

In a second experiment, real measurements of RSSI were collected from an indoor experiment at the Motorola facility in Plantation, Florida. The environment is a 14-by-13 m office area, partitioned by cubicle walls (height = 1.8 m). The network consisted of 40 unknown-location sensors and four anchors near the corners. The experimental settings are described more in detail in [44] (see also http://www.eecs.umich.edu/~hero/localize/). For each sensor i, we collected the RSSI associated to it in a 44-dimensional vector, denoted by $\boldsymbol{u}_i$. The intersensor similarity between sensors is given by the matrix $\tilde{K}$, defined between sensors i and j by

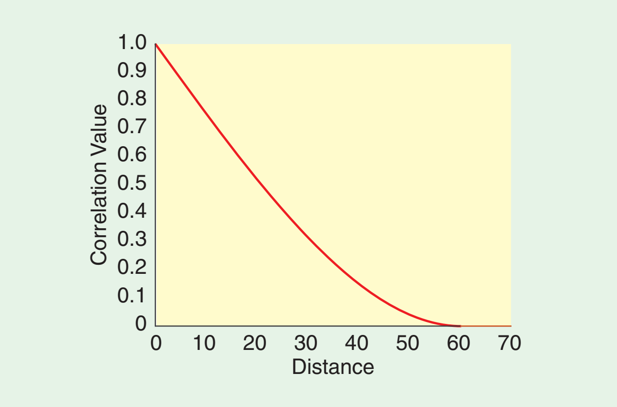
$$\tilde{K}(i,j) = \exp\left(-\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2/200\right).$$

The Gaussian kernel was considered, with its bandwidth optimized by maximizing the alignment. The proposed method gives a root-mean-square location error over the 40 sensors of 2.13 m each. This should be compared to the maximum-likelihood estimator studied in [44] (that turned out to be biased), having a root-mean-square location error of 2.18 m.

FINAL REMARKS

This article presented the preimage problem in machine learning, providing an overview of the state-of-the-art methods and approaches for solving such a problem. Our aim was to show how this problem is intimately related to dimensionality reduction issues, borrowing and enhancing ideas derived from dimensionality reduction and manifold learning. Throughout this article, we studied this problem for kernel PCA and provided a comparative study of several methods for image denoising. We extended the range of application of the preimage problem to another context, sensor autolocalization in wireless sensor networks.

By interpreting the processing in the feature space to the original input space, this strategy opens the way to a range of diverse signal-processing problems. These problems are nonlinear kernel-based formulations of classical signal processing methods, including the independent component analysis [45] and the Kalman filter [46]. Another area of application is the preimage problem on structured spaces, including biological sequence analysis in bioinformatics [47] and string analysis in natural language [48]. In the latter, the authors derived a preimage solution for a string kernel, using a graph-theoretical formulation. All these promising areas of application of the preimage problem open an avenue for future work.

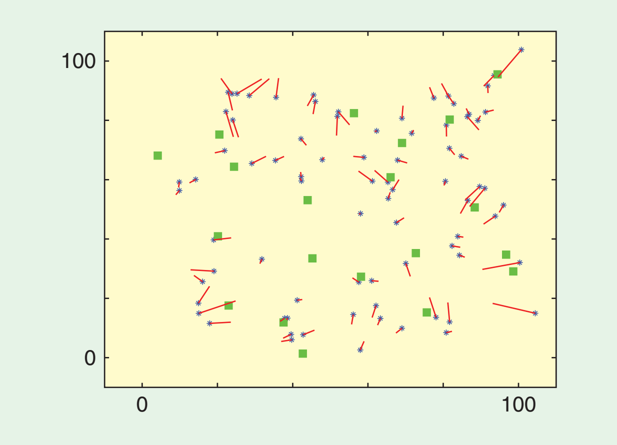


[FIG7] Profile of the spherical model as a function of the distance. The cutoff distance is set to $d = 60$.

AUTHORS

Paul Honeine (paul.honeine@utt.fr) received his Dipl.-Ing. degree in mechanical engineering in 2002 and his M.Sc. degree in industrial control in 2003, both from Lebanese University in Lebanon. In 2007, he received the Ph.D. degree from the University of Technology of Troyes, France. Since September 2008, he has been an assistant professor at the Institut Charles Delaunay (UMR CNRS 6279) at the University of Technology of Troyes, France. He is the coauthor of the 2009 Best Paper Award at the IEEE Workshop on Machine Learning for Signal Processing (MLSP). His research interests include nonstationary signal analysis, nonlinear adaptive identification, and machine learning.

Cédric Richard (cedric.richard@unice.fr) is a full professor at Observatoire de la Côte d'Azur, University of Nice, Sophia-Antipolis, France. He is a junior member of the Institut Universitaire de France. He was an associate editor for *IEEE Transactions on Signal Processing* (2006–2010). He is a member of the Signal Processing Theory and Methods Technical Committee of the IEEE Signal Processing Society and is the author of more than 100 papers. He received the 2009 Best Paper Award at the IEEE Workshop on MLSP. His current research interests include statistical signal processing and machine learning.



[FIG8] Estimated locations of 80 sensors (asterisk) based on 20 anchors of known positions (green squares), with error to real position represented by a line (—).

REFERENCES

[1] B. Schölkopf, A. Smola, and K. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[2] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[3] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. Müller, "Fisher discriminant analysis with kernels," in *Advances in Neural Networks for Signal Processing*, Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds. San Mateo, CA: Morgan Kaufmann, 1999, pp. 41–48.

[4] R. Rosipal and L. Trejo, "Kernel partial least squares regression in reproducing kernel hilbert space," *J. Mach. Learn. Res.*, vol. 2, pp. 97–123, Dec. 2002.

[5] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Signal Processing*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.

[6] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds. Cambridge, MA, MIT Press, vol. 12, 2000.

[7] S. Mika, B. Schölkopf, A. Smola, K. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," in *Proc. 1998 Conf. Advances in Neural Information Processing Systems II*. Cambridge, MA: MIT Press, 1999, pp. 536–542.

[8] G. Camps-Valls, J. L. Rojo-Alvarez, and M. Martinez-Ramon Manuel, Eds., *Kernel Methods in Bioengineering, Signal and Image Processing*. Hershey, PA: IGI Publishing, 2007.

[9] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge: U.K.: Cambridge Univ. Press, 2004.

[10] M. Aizerman, E. Braverman, and L. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," *Automat. Remote Contr.*, vol. 25, pp. 821–837, 1964.

[11] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," *Roy. Soc. Lond. Philos. Trans. A*, vol. 209, pp. 415–446, Jan. 1909.

[12] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.

[13] D. Alpay, Ed., *Reproducing Kernel Spaces and Applications*, (ser. Operator Theory: Advances and Applications). Cambridge, MA: Birkhäuser, 2003, vol. 143.

[14] E. Parzen, "Statistical inference on time series by RKHS methods," in *Proc. 12th Biennial Seminar*, R. Pyke, Ed. Montreal, Canada: Canadian Mathematical Congress, 1970, pp. 1–37.

[15] T. Kailath, "RKHS approach to detection and estimation problems—I: Deterministic signals in gaussian noise," *IEEE Trans. Inform. Theory*, vol. 17, no. 5, pp. 530–549, Sept. 1971.

[16] G. Wahba, *Spline Models for Observational Data*. Philadelphia: Society for Industrial and Applied Math (SIAM), 1990.

[17] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *J. Math. Anal. Applicat.*, vol. 33, no. 1, pp. 82–95, 1971.

[18] B. Schölkopf, R. Herbrich, and R. Williamson, "A generalized representer theorem," Royal Holloway College, Univ. London, U.K., Tech. Rep. NC2-TR-2000-81, 2000.

[19] C. A. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," *Construct. Approx.*, vol. 2, no. 1, pp. 11–22, Dec. 1986.

[20] F. Girosi, M. Jones, and T. Poggio, "Priors stabilizers and basis functions: From regularization to radial, tensor and additive splines," CBCL, MIT, Cambridge, MA, Tech. Rep. AIM-1430, CBCL-075, 1993.

[21] I. P. Schagen, "Interpolation in two dimensions—A new technique," *J. Inst. Math. Applicat.*, vol. 23, no. 1, pp. 53–59, 1979.

[22] B. Schölkopf, "Support vector learning," Ph.D. dissertation, Technischen Universität, Berlin, Germany, 1997.

[23] J. T. Kwok and I. W. Tsang, "The pre-image problem in kernel methods," in *Proc. 20th Int. Conf. Machine Learning (ICML)*. Washington, DC: AAAI Press, Aug. 2003, pp. 408–415.

[24] T. J. Abrahamsen and L. K. Hansen, "Input space regularization stabilizes pre-images for kernel PCA de-noising," in *Proc. IEEE Workshop Machine Learning for Signal Processing*, Grenoble, France, 2009, pp. 1–6.

[25] G. Bakir, J. Weston, and B. Schölkopf, "Learning to find pre-images," in *Neural Information Processing Systems 2003*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004, vol. 16, pp. 449–456.

[26] G. Bakir, "Extension to kernel dependency estimation with applications to robotics," Ph.D. dissertation, Tech. Univ., Berlin, Germany, Nov. 2005.

[27] W.-S. Zheng and J.-H. Lai, "Regularized locality preserving learning of pre-image problem in kernel principal component analysis," in *Proc. 18th Int. Conf. Pattern Recognition (ICPR)*. Washington, DC: IEEE Computer Society, 2006, pp. 456–459.

[28] W.-S. Zheng, J. H. Lai, and P. C. Yuen, "Penalized preimage learning in kernel principal component analysis," *IEEE Trans. Neural Networks*, vol. 21, no. 4, pp. 551–570, 2010.

[29] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, 2nd ed. (ser. Monographs on Statistics and Applied Probability). London, Chapman and Hall, Sept. 2000.

[30] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, Dec. 22, 2000.

[31] P. Etyngier, F. Ségonne, and R. Keriven, "Shape priors using manifold learning techniques," in *Proc. 11th IEEE Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[32] P. Honeine and C. Richard, "Solving the pre-image problem in kernel machines: A direct method," in *Proc. IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, Grenoble, France, Sept. 2009, pp. 1–6.

[33] Y. Yamanishi and J.-P. Vert. (2007). Kernel matrix regression. Tech. Rep. arXiv:q-bio/0702054v1 [Online] Available: http://arxiv.org/abs/q-bio/0702054v1

[34] Y. Lecun and C. Cortes. (1998). The MNIST database of handwritten digits [Online]. Available: http://yann.lecun.com/exdb/mnist

[35] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Los Alamitos, CA: IEEE Computer Society, 2001, vol. 4, pp. 2033–2036.

[36] C. S. Raghavendra and K. Sivalingam, Eds., *Proc. 2nd ACM Int. Workshop Wireless Sensor Networks and Applications*. San Diego, CA: ACM, Sept. 2003.

[37] J. Bachrach and C. Taylor, "Localization in sensor networks," *Handbook of Sensor Networks*, I. Stojmenovic, Ed. New Jersey, Wiley, pp. 277–310, 2005.

[38] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Comput. Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[39] N. Patwari and A. O. Hero, "Manifold learning algorithms for localization in wireless sensor networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, May 2004, vol. 3. pp. 857–860.

[40] M. Essoloh, C. Richard, H. Snoussi, and P. Honeine, "Distributed localization inwireless sensor networks as a pre-image problem in a reproducing kernel hilbert space," in *Proc. European Conf. Signal Processing (EUSIPCO)*, Lausanne, Switzerland, Aug. 2008, pp. 1–5.

[41] N. Cristianini, A. Elisseeff, J. Shawe-Taylor, and J. Kandola, "On kernel target alignment," in *Proc. Neural Information Processing Systems (NIPS)*, 2002, pp. 367–373.

[42] K. Kim, M. Franz, and B. Schölkopf, "Iterative kernel principal component analysis for image modeling," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 9, pp. 1351–1366, 2005.

[43] T. Gneiting, "Compactly supported correlation functions," Environmental Protection Agency, NRCSE, Seatle, WA, Tech. Rep. NRCSE-TRS No. 045, May 2000.

[44] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[45] J. Yang, X. Gao, D. Zhang, and J.-Y. Yang, "Kernel ICA: An alternative formulation and its application to face recognition," *Pattern Recognit.*, vol. 38, no. 10, pp. 1784–1787, Oct. 2005.

[46] L. Ralaivola and F. D'Alché-Buc, "Time series filtering, smoothing and learning using the kernel Kalman filter," in *Proc. Int. Joint Conf. Neural Networks*, 2005, vol. 3. pp. 1449–1454.

[47] S. Sonnenburg, A. Zien, P. Philips, and G. Ratsch, "Poims: positional oligomer importance matrices—Understanding support vector machine-based signal detectors," *Bioinformatics*, vol. 24, no. 13, pp. i6–14, July 2008.

[48] C. Cortes, M. Mohri, and J. Weston, "A general regression technique for learning transductions," in *Proc. 22nd Int. Conf. Machine Learning (ICML)*. New York, ACM, 2005, pp. 153–160.