Noise Level Estimation for Model Selection in Kernel PCA Denoising

Carolina Varon, *Member, IEEE*, Carlos Alzate, and Johan A. K. Suykens, *Fellow, IEEE*

***Abstract*— One of the main challenges in unsupervised learning is to find suitable values for the model parameters. In kernel principal component analysis (kPCA), for example, these are the number of components, the kernel, and its parameters. This paper presents a model selection criterion based on distance distributions (MDDs). This criterion can be used to find the number of components and the σ^2 parameter of radial basis function kernels by means of spectral comparison between information and noise. The noise content is estimated from the statistical moments of the distribution of distances in the original dataset. This allows for a type of randomization of the dataset, without actually having to permute the data points or generate artificial datasets. After comparing the eigenvalues computed from the estimated noise with the ones from the input dataset, information is retained and maximized by a set of model parameters. In addition to the model selection criterion, this paper proposes a modification to the fixed-size method and uses the incomplete Cholesky factorization, both of which are used to solve kPCA in large-scale applications. These two approaches, together with the model selection MDD, were tested in toy examples and real life applications, and it is shown that they outperform other known algorithms.**

***Index Terms*— Kernel principal component analysis (kPCA), least squares support vector machines (LS-SVMs), noise level estimation, unsupervised learning.**

I. Introduction

PRINCIPAL component analysis (PCA) [1] is an unsupervised learning technique used in many applications such as dimensionality reduction, data compression, denoising, and pattern recognition. It transforms an input dataset $\mathcal{D} = \{x_i\}_{i=1}^N$ of N zero-mean data points $x_i \in \mathbb{R}^d$ into a set of n_c uncorrelated variables, also known as *principal axes*. These variables correspond to the leading eigenvectors of the covariance matrix of $\mathcal{D}$, and they span a subspace that retains maximum variance of the data. A clear disadvantage of this technique is that only linear structures can be represented by these principal axes. However, these limitations are overcome by adaptations of PCA, such as those recently proposed in [2] and [3]. Another well-known adaptation is kernel PCA (kPCA) [4], which first applies a (possibly) nonlinear transformation φ to $\mathcal{D}$, to map the input space to a high-dimensional feature space $\mathcal{F}$. The mapping is performed by means of Mercer kernels K, which are the functions that generate positive semidefinite kernel matrices [5]. Classical PCA is then applied in this kernel induced space, and the resulting principal axes lie now in $\mathcal{F}$, rather than in the input space. To find the reverse mapping from $\mathcal{F}$ that transforms the denoised or reconstructed data back into the input space (the so-called *preimage* problem), Mika *et al.* [5] and Kwok *et al.* [6] find approximate solutions based on the distances between the input data and its (nonlinear mapped) projections in $\mathcal{F}$.

A well known challenge in unsupervised learning techniques, such as PCA and kPCA, is the selection of the model parameters. For example, the number of components n_c is required in both PCA and kPCA, while the kernel parameter is important only in kPCA. Different methodologies have been proposed to perform this selection. In [7], the problem of PCA was formulated using the Bayessian inference, which allows for automatic determination of the number of components that describe the structure of the data. Another method that uses the Bayessian inference to find the intrinsic dimensionality of the data was proposed in [8], and it allows to define the observation likelihood using not only the Gaussian functions but also exponential family distributions. Concerning the model selection in the kPCA, different heuristics [5], [9] and (semi) theoretical models have been proposed. For instance, in [10], a general loss function is included in the formulation of the problem and the model selection is carried out by means of a heuristic. A different approach is proposed in [11], where a noise level is estimated using parallel analysis. This method uses a null-dataset to estimate the level of noise. It does have some disadvantages, first, the null-dataset requires many permutations, and second, the cost function does not always indicate a clear maximum. Moreover, to have good noise level estimation in low dimensions, it is necessary to rotate the input data into a higher dimensional space. This last approach defines the first principal components as the meaningful information in the data, and the rest of components are associated with noise. This denoising concept is related to the minimum description length principle [12], which finds the minimum number of components needed to fully represent the intrinsic information in the data. The separation of the noise components is performed in the eigenvalue distribution,

Manuscript received October 1, 2013; revised December 23, 2014; accepted December 27, 2014. Date of publication January 16, 2015; date of current version October 16, 2015. This work was supported in part by the Fonds Wetenschappelijk Onderzoek under Grant G.0377.12, in part by the European Research Council under Grant AdG A-DATADRIVE-B, Grant CoE EF/05/006, and Grant GOA/10/09 through mobile ad hoc network, in part by the Interreg IVB NWE Programme through the European Union under Grant RECAP 209G, in part by Katholieke Universiteit Leuven, Leuven, Belgium, in part by the Flemish Government, and in part by the Belgian Federal Science Policy Office.

C. Varon and J. A. K. Suykens are with the STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Department of Electrical Engineering, and also with the iMinds Future Health Department, Katholieke Universiteit Leuven, Leuven B-3001, Belgium (e-mail: carolina.varon@esat.kuleuven.be; johan.suykens@esat.kuleuven.be).

C. Alzate is with the Smarter Cities Technology Centre, IBM Research—Ireland, Dublin 15, Ireland (e-mail: carlos.alzate@ie.ibm.com).

Digital Object Identifier 10.1109/TNNLS.2015.2388696

2162-237X © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

which is also the way in which the proposed approach works. Here, a new way to estimate the level of noise is presented. It requires neither rotations nor permutations, it can handle low dimensionality, and it is observed that the objective function has a global maximum for all the given examples. This new approach is called model selection based on distance distributions (MDDs), and it uses the statistical properties of the distribution of distances of the input data to generate a noise distribution, based on the assumption that uniformly distributed data generate unimodal (Euclidean) distance distributions. This property has been studied in [13]–[15] and used in [16] in the framework of visualization and exploration of high-dimensional data.

The selection of the model parameters is not the only challenge in unsupervised learning. A different problem arises when large-scale datasets need to be analyzed. When N is large, building a kernel matrix Ω becomes impractical. To overcome this problem, low-rank approximations of the kernel matrix, such as the Nyström method [17] and incomplete Cholesky factorization (ICF) [18], have been used [19]–[23]. They reduce the size of the eigenvalue problem and approximate the *full-size* eigenvectors. As a result, the model can be trained with a smaller set and the projections of the remaining data points can be predicted. In this paper, a slight modification of the fixed-size (FS) method [19], which is based on the Nyström approximation, is proposed.

The remainder of this paper is structured as follows. Section II describes the least squares support vector machine (LS-SVM) formulation of kernel PCA. In Section III, the problem of manifold learning by estimating the noise in different dimensionality is discussed. The criterion proposed in this paper to select the model parameters n_c and σ^2 is presented in Section IV. Section V explains two algorithms for large-scale kPCA and the adaptation to the FS method proposed in this paper. The experimental results using toy examples and real life problems are discussed in Section VI, and the conclusion is presented in Section VII.

II. LS-SVM Version of Kernel PCA

LS-SVM formulations to unsupervised problems, such as kernel PCA, were discussed in [24]. These formulations are based on primal-dual interpretations, where the dual problem is equivalent to the original formulation of kernel PCA [4]. The reason to use the LS-SVM formulation relies on the fact that it allows to perform estimations in the primal space, which represent an advantage for large-scale applications, where solving the dual problem for a very large dataset becomes infeasible [19]. This formulation also allows one to perform out-of-sample extensions. In Section V, two large-scale methodologies will be discussed.

Given a set of N data points $\{x_i\}_{i=1}^N$ with $x_i \in \mathbb{R}^d$, the projected variables $w^T\varphi(x_i)$, and $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d_h}$, the mapping to a high-dimensional feature space $\mathcal{F}$ of dimension d_h, the *primal problem* can be defined as

$$\begin{aligned} \max_{w,e} \quad & J_P(w,e) = \gamma \frac{1}{2}\sum_{i=1}^{N} e_i^2 - \frac{1}{2} w^T w \\ \text{s.t.} \quad & e_i = w^T(\varphi(x_i) - \hat{\mu}_\varphi), \quad i = 1,\ldots,N \end{aligned} \tag{1}$$

with γ a positive regularization constant and $\hat{\mu}_\varphi = (1/N)\sum_{i=1}^N \varphi(x_i)$. The Lagrangian of this primal problem is defined as

$$\begin{aligned} \mathcal{L}(w,e;\alpha) = \gamma \frac{1}{2}\sum_{i=1}^{N} e_i^2 - \frac{1}{2} w^T w \\ - \sum_{i=1}^{N} \alpha_i (e_i - w^T(\varphi(x_i) - \hat{\mu}_\varphi)) \end{aligned} \tag{2}$$

with Karush–Kuhn–Tucker (KKT) conditions for optimality

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow \; w = \displaystyle\sum_{i=1}^{N} \alpha_i (\varphi(x_i) - \hat{\mu}_\varphi) \\ \dfrac{\partial \mathcal{L}}{\partial e_i} = 0 \rightarrow \; \alpha_i = \gamma e_i, \quad i = 1,\ldots,N \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \rightarrow \; e_i - w^T(\varphi(x_i) - \hat{\mu}_\varphi) = 0, \\ \qquad\qquad\qquad\qquad i = 1,\ldots,N. \end{cases} \tag{3}$$

After eliminating the primal variables e and w and defining $\lambda = 1/\gamma$, the dual formulation becomes

$$\Omega_c \alpha = \lambda \alpha \tag{4}$$

where Ω_c is the centered kernel matrix with entries defined as

$$\Omega_{c,ij} = (\varphi(x_i) - \hat{\mu}_\varphi)^T(\varphi(x_j) - \hat{\mu}_\varphi), \quad i,j = 1,\ldots,N. \tag{5}$$

The mapping $\varphi(\cdot)$ might be unknown, but based on the Mercer theorem, it can implicitly be defined by means of a kernel function $K(x_i,x_j) = \varphi(x_i)^T\varphi(x_j)$. This kernel function is then used to compute a kernel matrix $\Omega_{ij} = K(x_i,x_j)$ and its centered version $\Omega_c = M_c\Omega M_c$ with the centering matrix

$$M_c = I - \frac{1}{N} 1_N 1_N^T \tag{6}$$

and $1_N = [1;1;\ldots;1]$. In this paper, the radial basis function (RBF) kernel function is used.

The eigenvalue problem (4) satisfies the KKT conditions (3), and every eigenvalue–eigenvector pair of the centered kernel matrix can be seen as a candidate solution of (1). The direction of maximal variance in the high-dimensional space, i.e., *the first principal component*, will be determined by the eigenvector corresponding to the largest eigenvalue of Ω_c. The total amount of components that can be extracted is determined by the size of the kernel matrix that is given by the number of training points.

The projection of an input data point x onto a subspace spanned by the most relevant principal axes in the feature space, also known as the score variable, can be obtained as

$$\begin{aligned} e(x) &= w^T(\varphi(x) - \hat{\mu}_\varphi) \\ &= \sum_{i=1}^{N} \alpha_i \Bigg(K(x_i,x) - \frac{1}{N}\sum_{j=1}^{N} K(x_j,x) - \frac{1}{N}\sum_{j=1}^{N} K(x_j,x_i) \\ &\qquad + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{k=1}^{N} K(x_j,x_k) \Bigg). \end{aligned} \tag{7}$$

For denoising applications, it is necessary to map the transformed points from the feature space back into the input space. This is called the *preimage* problem and is recognized as an ill-posed problem, where the solution does not always exist, and if it exists, it is not always unique [5], [6]. This problem becomes of primary interest in denoising applications, and several methodologies have been proposed to find approximate preimages. In this paper, an iterative fixed-point algorithm for RBF kernels [5] implemented in the least-squares support vector machines MATLAB toolbox[1] is used.

III. Noise and Dimensionality

It is well known that the main problem in PCA is to separate the components containing the information from those describing the noise in the dataset. One way of doing this is by comparing the eigenvalues computed from the data with the ones from a set of uncorrelated variables (i.e., null dataset), as proposed in parallel analysis [25] and later used in its kernel version [11]. This idea of spectral comparison is also used in this paper, but with a novel approach to estimate the noise eigenvalues. Details of this approach will be described later. For now, the focus will be put on the problems that are encountered when the null data is created from permutations of the input dataset, as in [11] and [25]. This will allow one to get more insights on the problem of noise estimation.

To find the noise spectrum, the input dataset can be first permuted several times along the dimensions, to destroy any manifold structure in the dataset [11]. After each permutation, a centered kernel matrix is computed with its corresponding eigenvalues (4). Finally, the noise spectrum is then characterized by the 95th percentile of the obtained eigenvalue distribution.

As an example, one can take a structured dataset [Fig. 1 (top)] $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{100}$, with $\mathbf{x}_i \in \mathbb{R}^2$, and another dataset with the same structure in 50 dimensions: $\mathcal{D}' = \{\mathbf{x}'_i\}_{i=1}^{100}$, with $\mathbf{x}'_i \in \mathbb{R}^{50}$. To create $\mathcal{D}'$, $\mathcal{D}$ is projected onto $\mathbb{R}^{50}$, where 48 dimensions equal zero. This new dataset is $\mathcal{D}^{50} = \{\mathbf{x}_i^{50}\}_{i=1}^{100}$, with $\mathbf{x}_i^{50} = [\mathbf{x}_i; 0; \ldots; 0]$, where $\mathbf{x}_i^{50} \in \mathbb{R}^{50}$. Then, a rotation matrix W$\in \mathbb{R}^{50\times 50}$ of the form

$$W = \begin{bmatrix} \mathbf{w} & 0 & \ldots & 0 \\ 0 & \mathbf{w} & \ldots & 0 \end{bmatrix} \tag{8}$$

with $\mathbf{w} = [w(1); \ldots; w(25)]$ the 25-point symmetric hamming window vector is used to generate the high-dimensional dataset $\mathrm{D}' = \mathrm{WD}^{50}$, with $\mathrm{D}' = [\mathbf{x}'_1, \ldots, \mathbf{x}'_{100}]$ and $\mathrm{D}^{50} = [\mathbf{x}_1^{50}, \ldots, \mathbf{x}_{100}^{50}]$.

The spectrum of $\mathrm{D}' \in \mathbb{R}^{50}$ for a given σ^2 is shown in Fig. 1 (bottom left). There, a clear difference between the eigenvalues of the structured data (solid line) and the ones of the permuted data (dashed line) can be observed. This implies that for 50 dimensions, the permutations successfully eliminate the existing structure contained in the set $\mathcal{D}$. This, however, is no longer the case in lower dimensions, as is shown in Fig. 1 (bottom right), where a dataset in $d = 2$ is used. This phenomenon, also known as *curse of dimensionality* [26], is due to the fact that the higher (lower) the dimensionality, the higher (lower) the entropy and the harder (easier) it is to find any shape or any correlation between the dimensions.

[1] Available at http://www.esat.kuleuven.be/stadius/lssvmlab/.

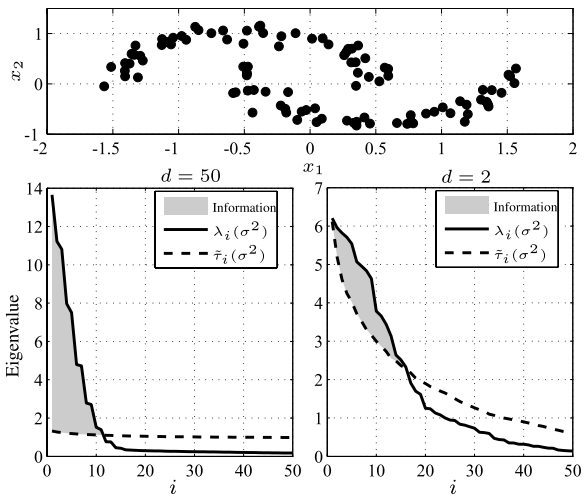


Fig. 1. Top: structured dataset $\mathcal{D}$, where $N = 100$, and $d = 2$. Bottom: data and noise spectra of the dataset in a 50-D, and in a 2-D space. The first 50 eigenvalues λ_i of a centered kernel matrix computed for a given σ^2 are displayed. The noise levels $\hat{\tau}_i$ are defined as the 95th percentiles of the eigenvalue distributions after 100 permutations of the input data. Note that after permuting the data in two dimensions, the noise level is highly similar to the eigenvalue spectrum of the data. This indicates the presence of structure in the permuted dataset. In 50 dimensions, this is no longer the case; instead, the noise level resembles the spectrum of a random dataset.

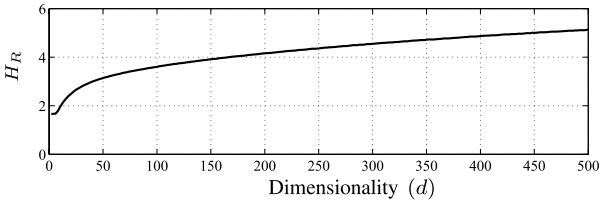


Fig. 2. Quadratic Renyi entropy H_R of a structured dataset of 100 points contained in different dimensions. Note that as the dimensionality increases, the harder it is to find any structure in the dataset.

Another way of observing this effect is by measuring the quadratic Renyi entropy of the data defined as

$$H_R = -\log \int p(x)^2 dx \tag{9}$$

which is related [27] to kernel PCA and density estimation through

$$\int \hat{p}(x)^2 dx = \frac{1}{N^2} 1_N^T \Omega 1_N \tag{10}$$

where Ω is the kernel matrix. This is shown in Fig. 2, where the quadratic Renyi entropy H_R for the given dataset increases (i.e., more uniformity) as a function of the dimensionality d.

The fact that any structure in a dataset cannot be easily destroyed in lower dimensions makes algorithms like kernel parallel analysis (kPA) [11] (or a method based on *random data points*) unsuccessful in estimating noise in low-dimensionality spaces. This limitation is overcome with the method proposed in this paper, which is able to get an

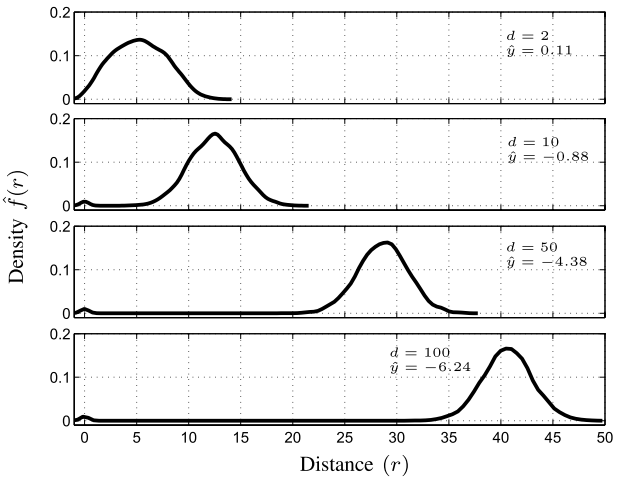


Fig. 3. Density functions of distances r for a dataset of 100 uniformly distributed points in spaces of 2, 10, 50, and 100 dimensions. Note that the skewness $\hat{y}$ of the distribution decreases as d increases.

estimation of the noise level, even in lower dimensions. This method is called the MDDs.

Before going into the details of MDD, however, it is important to understand the influence of dimensionality on the distribution of distances in a dataset. It has been shown in [13]–[16] that for uniformly distributed (i.e., unstructured) points, the probability distribution of the distances between the points is unimodal. Furthermore, the statistical moments of the distribution change as a function of the dimensionality. This is indicated in Fig. 3, where the distribution of $N = 100$ uniformly distributed data points in $[0, 1]^d$ clearly shows a unimodal structure, and its statistical moments change for increasing dimensionality.[2] Therefore, the main assumptions made in this paper are that the entropy for unstructured datasets (i.e., noise or uniformly distributed data points) is higher than the one for structured datasets and that noise is characterized by a unimodal distance distribution, as if it were generated by a uniformly distributed dataset.

IV. Model Selection Based on Distance Distributions

The assumption made at the end of the previous section is essential for the method presented here, which can be used to select the number of components n_c and the RBF kernel parameter σ^2. This method is called the MDDs.

Given a training set $\mathcal{D}^t = \{x_n^t\}_{n=1}^{N_t}$, $x_n^t \in \mathbb{R}^d$, and a validation set $\mathcal{D}^v = \{x_m^v\}_{m=1}^{N_v}$, $x_m^v \in \mathbb{R}^d$, the Euclidean distances between $x_i \in \mathcal{D}^t$ and $x_j \in (\mathcal{D}^t \cup \mathcal{D}^v)$, $r_{ij} = \|x_i - x_j\|_2$, with $i = 1, \ldots, N_t$ and $j = 1, \ldots, N_t + N_v$, are computed. This results in a distribution of distances $\hat{f}_{\mathcal{D}}(r)$. From this distribution, the first four statistical moments, namely, the mean μ, standard deviation σ, skewness $\hat{y}$, and kurtosis β, are obtained and used to generate a second distribution[3] $\hat{f}_{\mathcal{N}}(\hat{r})$, with $0 \leq \hat{r} \leq r_{\max}$, where $r_{\max}$ is the maximum distance contained in the dataset. This second distribution is generated to mimic randomization of the input data points and to estimate the distance distribution of a noisy dataset. Values drawn from $\hat{f}_{\mathcal{N}}(\hat{r})$ are used as entries to a new distance matrix $\hat{R} \in \mathbb{R}^{N_t \times N_t}$, and they are organized in such a way that the conditions of a metric are satisfied. These conditions for arbitrary points x_i, x_j, and a are defined as

$$\begin{aligned} &1.\ \hat{r}(x_i, x_j) \geq 0 \\ &2.\ \hat{r}(x_i, x_j) = 0, \quad \text{iff } i = j \\ &3.\ \hat{r}(x_i, x_j) = \hat{r}(x_j, x_i) \\ &4.\ \hat{r}(x_i, a) \leq \hat{r}(x_i, x_j) + \hat{r}(x_j, a). \end{aligned} \tag{11}$$

The first condition guarantees nonnegativity of the metric, and it is satisfied by selecting points that are only included in $\hat{f}_{\mathcal{N}}(\hat{r})$. The second one is fulfilled by making $\hat{r}(x_i, x_i) = 0$ for $i = 1 \ldots, N$, and the symmetry of $\hat{R}$ will guarantee the third condition. The last condition is the so-called *triangle inequality* and needs to be satisfied in a recursive fashion, and again, only values selected from $\hat{f}_{\mathcal{N}}(\hat{r})$ will be used in the matrix. Details on how to generate this new distance matrix are presented in the Appendix. Once $\hat{R}$ is created, its entries will be used in an RBF kernel as

$$K_{\mathcal{N}}(x_i, x_j) = \exp\left(-\frac{\hat{r}(x_i, x_j)^2}{2\sigma^2}\right) \tag{12}$$

and the eigenvalue spectrum $\hat{\tau}$ for a given σ^2 is estimated from

$$\Omega_{\mathcal{N}_c} \nu = \hat{\tau} \nu \tag{13}$$

with $\Omega_{\mathcal{N}_c} = M_c \Omega_{\mathcal{N}} M_c$, where M_c is as defined in (6), and $\Omega_{\mathcal{N}_{ij}} = K_{\mathcal{N}}(x_i, x_j)$.

This artificial spectrum characterizes the noise level and can be compared with the one of the original data λ_i (4). It is important to keep in mind that the noise is estimated using a validation set that avoids overfitting of the training data. The number of components for a given σ^2 that contain information about the structure in the dataset will then be selected as

$$n_c(\sigma^2) = \max_{\lambda_i - \hat{\tau}_i > 0} i \tag{14}$$

and the kernel parameter σ^2 can be obtained after maximizing the structural information I_s expressed by n_c with respect to the noise, which is defined as

$$I_s(\sigma^2) = \sum_{i=1}^{n_c(\sigma^2)} \lambda_i - \hat{\tau}_i. \tag{15}$$

This is not the same as maximizing the variance explained by the first n_c components $\sum_{i=1}^{n_c} \lambda_i$ in the sense that here the estimated noise information (unstructured information) contained in each component is also considered. This criterion of maximizing the difference in variance between the data and the estimated noise is also used in [11] under the name of energy. The main difference between the estimation of I_s and energy relies on the definition of noise. In this paper, a novel procedure to estimate the noise is proposed, which works in

[2]The density function is computed using kernel smoothing density estimation, and the smoothing parameter is computed using the *Silverman rule* defined in [28].

[3]The Pearson system is used to generate the distribution. In MATLAB, this corresponds to the function *pearsrnd*, contained in the statistics toolbox.

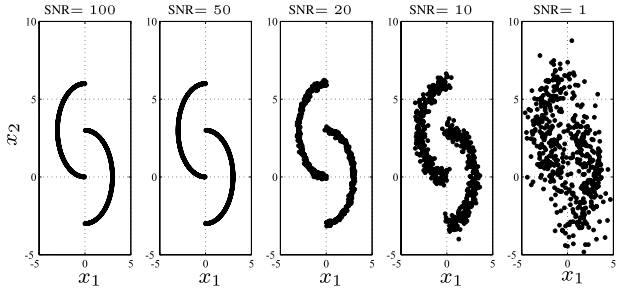


Fig. 4. Datasets with different SNRs.

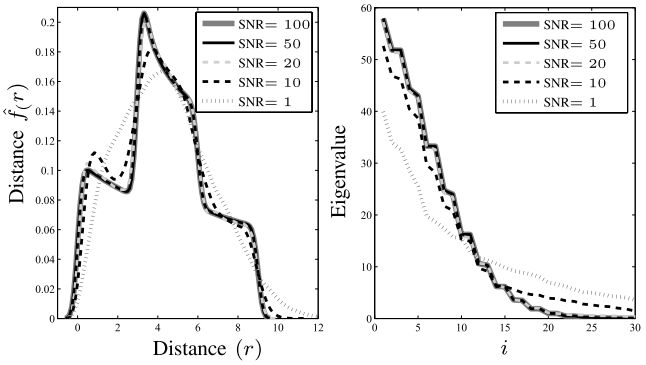


Fig. 5. Left: distance distributions of datasets with different SNRs. Right: eigenvalue spectra for each case with $\sigma^2 = 0.8$.

both high and low dimensionality. This is not the case for the noise estimation proposed in [11], which only works if the data is contained in a high-dimensional space (Section III). For low-dimensional spaces, there exists the possibility that the noise spectrum estimated with [11] will indicate a larger variance than the one in the input data. Besides, when the signal-to-noise ratio (SNR) is very small, the spectrum of the data still indicates the presence of structure. This will be clarified in the next example.

The datasets indicated in Fig. 4 represent the same structure but with different SNRs. If the distance distribution for each example is calculated, it is observed that the behavior tends to be unimodal as soon as more noise is added to the dataset. This is shown in Fig. 5. Note that the eigenvalue spectrum for the case with SNR $= 1$ still indicates the presence of information related to the original dataset. Now, if the distance distribution of the last case is compared against a unimodal one that is representing uniformly distributed points, the differences in noise spectra become more evident (Fig. 6).

In the last case, one can see that by creating a unimodal distance distribution from the statistical moments of the original one, and by limiting the values to the maximum distance in the real set, as done in MDD, a certain structure is imposed to the noise. This can be observed in the first eigenvalue of the spectrum of the noise. From the spectrum of the estimated noise, it is also possible to see that the power is now spread among all the other components, as expected for uniform noise. The main difference between this spectrum and the one produced by data permutation [11] is that for MDD the estimated noise does not follow the same pattern of the real

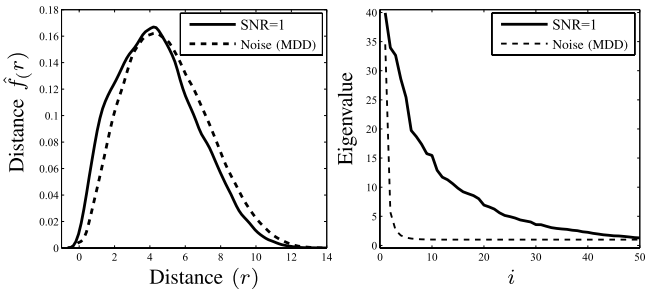


Fig. 6. Left: distance distributions of the dataset with SNR $= 1$ and the generated one using the statistical moments of the first one, as proposed in MDD. Right: eigenvalue spectra with $\sigma^2 = 0.8$.

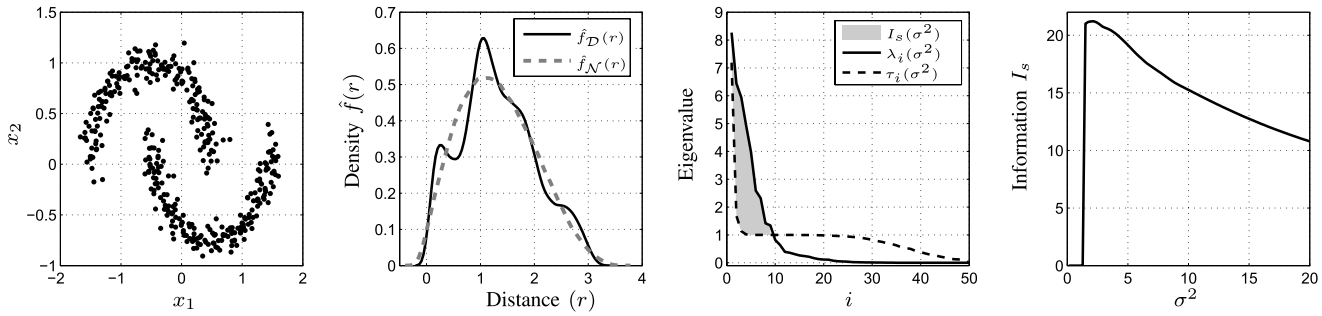
data. Therefore, only when the data is permuted to estimate the noise, those two spectra may encounter each other in more places. It is important to remember that the four statistical moments of the distance distribution are needed to estimate the noise. This was discussed in Section III, and it is related to the fact that important information about the dimensionality of the data is contained in the values of skewness $\hat{y}$ and kurtosis β. For low-dimensionality problems, using the first two statistical moments of the original distance distribution is enough, since with less dimensions, the distribution is closer to a normal one.

Now, concerning the first component as mentioned before, components are only selected when the noise is not overestimated. In other words, they are selected when the first component of the noise is lower than the one of the real data.

An important aspect to be considered here is the fact that the distance matrix is generated from a probability distribution, which requires the implementation of a Monte Carlo simulation. Note that the original distance values are not reused or resampled as in bootstrapping methods; on the contrary, new distance values are drawn from $\hat{f}_{\mathcal{N}}(\hat{r})$, which implies a data generation process characteristic of Monte Carlo methods. Here, the eigenvalue spectrum is defined as the 95th percentile confidence interval of each eigenvalue distribution generated after q initializations. In this paper, $q = 100$. Another aspect to keep in mind is that as the number of data points increases, the estimation of the statistical moments of the distance distribution becomes more accurate. Therefore, deviations in this estimation need to be considered when the size of the dataset is very small.

The proposed methodology is illustrated in the following example. Given is the dataset shown in Fig. 7(a) with $N = 200$, where $N = N_t + N_v$. The distance distribution computed from the data and the one generated from its statistical moments are indicated in Fig. 7(b). After generating a distance matrix $\hat{R}$ and computing its eigenvalues $\hat{\tau}$ hundred times, the 95th percentile of the eigenvalue distribution is obtained and is represented by the dashed line in Fig. 7(c). The shaded area can be interpreted as the amount of information contained in the first nine components. Fig. 7(d) shows the information content I_s as a function of the kernel parameter σ^2.

It is also possible to maximize I_s for a known number of components. There are some applications in which n_c is already given [29], and one only needs to estimate the best σ^2.



(a) (b) (c) (d)

Fig. 7. Selection of the number of components n_c and σ^2. (a) Dataset consists of $N_t = 100$ and $N_v = 100$. (b) Density function of the distances in the data $\hat{f}_{\mathcal{D}}(r)$ (solid line) and the noise $\hat{f}_{\mathcal{N}}(r)$ (dashed line). (c) Eigenvalue spectrum of the data λ_i (solid line) and the noise $\hat{\tau}_i$ (dashed line). The shaded area corresponds to the structural information $I_s(\sigma^2)$ explained by the first nine components. (d) Information content as a function of σ^2.

V. Large Scale kPCA

From (7), it is clear that to compute the principal component projections of an input data point x, it is necessary to evaluate the kernel function $K(x, x_i)$, $i = 1, \ldots, N$ in all the training examples and to compute its eigendecomposition. This constitutes an issue since the computational cost of the algorithm directly depends on the size of the kernel matrix; hence in large scale applications, this becomes a limitation.

Different methodologies have been proposed to tackle this issue, for example, in [30] a probabilistic approach is used to reduce the number of training vectors. The size of the kernel matrix can also be reduced by means of low-rank approximation methods, such as Nyström [17], and ICF where a set of approximated eigenvectors are computed [20], [21]. These methodologies aim to find an approximated kernel matrix $\Omega \in \mathbb{R}^{M \times M}$, where $M \ll N$. Using the Nyström approximation method, the reduced set of support vectors is selected at random, but this does not guarantee a close representation of the underlying distribution of the data. A more suitable selection of support vectors is achieved using the FS method that actively selects the training examples using an iterative procedure. The FS and ICF are used in this paper and they will be discussed in the following sections.

1) Fixed-Size Method (FS): To select a subset of support vectors, the FS method was proposed in [19]. It actively selects the M support vectors that best represent the underlying distribution of the dataset by maximizing the quadratic Renyi entropy defined in (9).

In this paper, a slight modification to the original implementation presented in [19] is proposed under the name FS2. Here, the initial support vectors are not selected at random; instead, they correspond to the centroids of k clusters computed using k-means.

Remark 1: Note that the centroids of the clusters obtained with k-means already represent a certain structure in the data, which allows one to initialize the search of support vectors with a relatively high value of entropy.

The second modification to the original algorithm avoids the random replacement of candidate support vectors. In this paper, the candidate support vector that indicates the maximum similarity with the rest of the training points is replaced with another vector in the remaining set. In this way, the quadratic Renyi entropy converges faster.

2) Incomplete Cholesky Factorization (ICF): Any symmetric positive definite matrix $A \in \mathbb{R}^{N \times N}$ can be decomposed as $A = HH^T$, where H is a lower triangular matrix, called the Cholesky factor of A. This decomposition is not always unique or it may not even exist when A is positive semidefinite. However, it is still possible to compute a lower triangular matrix $G \in \mathbb{R}^{N \times Q}$ with $Q < N$ and $N-Q$ columns of G equal to zero such that $A \approx GG^T$. This ICF [18] allows one to compute a close approximation of the exact Cholesky factor H, where $\|A - GG^T\|_2^2 \leq \eta$, with η a user-defined error threshold. The matrix G is formed in an iterative way by performing symmetric permutations of the rows and columns of A and by keeping up to Q pivots for which the sum of the remaining $N-Q$ pivots is lower than η. Therefore, the larger η, the less pivots are selected, $Q \ll N$, and the worse the approximation is. The set of remaining $N-Q$ data points can be split into validation and test. This low-rank approximation of a positive semidefinite matrix has been used to find a reduced set of representative training points for sparse kernel spectral clustering [20], [21], to reduce the computational complexity of interior point methods in SVM training [31], and to compute approximated kernel ICA contrast functions [22], [32].

In this paper, the ICF is used to find an approximate Cholesky factor G of the kernel matrix Ω such that $\Omega \approx GG^T$. This can be done using the toolbox of kernel independent component analysis[4] described in [22]. From the factorization of the kernel matrix, it is possible to compute an approximated set of eigenvectors related to (4), as described in [20] and [21] in the framework of sparse kernel spectral clustering.

The singular value decomposition of the triangular matrix $G \in \mathbb{R}^{N \times Q}$ for $Q < N$ is

$$G = U \Lambda V^T \tag{16}$$

where the columns of $U \in \mathbb{R}^{N \times Q}$ correspond to the left singular vectors, $\Lambda \in \mathbb{R}^{Q \times Q}$ contains the singular values in the diagonal, and the columns of $V \in \mathbb{R}^{Q \times Q}$ are the right singular vectors of G. From this decomposition, the ICF of Ω can then be written as

$$\Omega \approx U \Lambda^2 U^T. \tag{17}$$

[4]Available at http://www.di.ens.fr/~fbach/kernel-ica/index.htm.

This decomposition can be used to rewrite (4) as

$$M_c U \Lambda^2 U^T \alpha^{(l)} = \lambda_l \alpha^{(l)}, \quad l = 1, \ldots, n_c \tag{18}$$

where M_c is the centering matrix defined in (6) and n_c is the number of principal components to compute.

Premultiplying (18) by U^T results in

$$\begin{aligned} U^T M_c U \Lambda^2 U^T \alpha^{(l)} &= \lambda_l U^T \alpha^{(l)} \\ U^T M_c U \Lambda^2 \rho^{(l)} &= \lambda_l \rho^{(l)} \end{aligned} \tag{19}$$

with $\rho^{(l)} = U^T \alpha^{(l)}$ and $l = 1, \ldots, n_c$.

Remark 2: Note that the size of the complexity of the eigendecomposition has been reduced because the training matrix is now of the size $Q \times Q$, and the eigenvectors $\alpha^{(l)} \in \mathbb{R}^N$ can be approximated by $U^T \rho$, with $\rho^{(l)} \in \mathbb{R}^Q$, where $Q < N$.

This methodology can be used when the σ^2 is given and one only wants to find the optimal number of components n_c. It could also be used to optimize the kernel parameter, but its high computation time will make it impractical. This is due to the fact that for different values of σ^2, a different set of pivots is selected and a new approximated Cholesky factor is computed. By doing so, a new distance matrix must be obtained, which requires higher computation times as the number of pivots increase. With this in mind, it is better to use ICF for cases when σ^2 is given.

These two algorithms can be used in combination with the model selection procedure presented in Section III. Their implementation is summarized in Fig. 8.

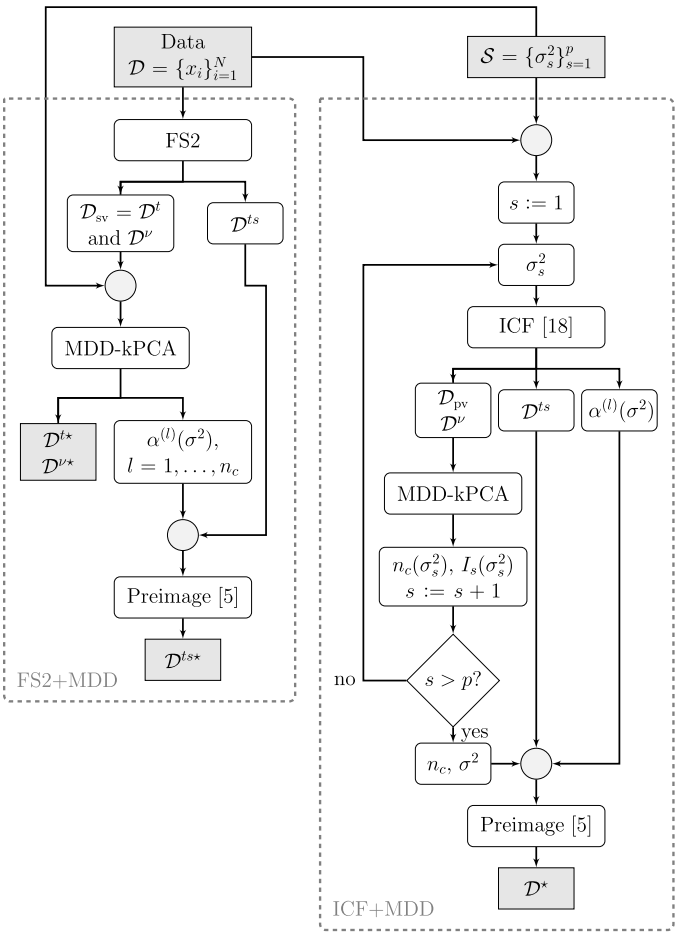


Fig. 8. Implementation of the large-scale kPCA with the proposed model selection algorithm MDD.

VI. Experimental Results

This section describes some experimental results, which were carried out in MATLAB R2012a on an Intel Core i7, 3.4 GHz, 7.8 GB RAM, running Ubuntu 12.04 Long Term Support. The RBF kernel function was used for all the experiments and a grid search for the bandwidth parameter $\mathcal{S} = \{\sigma_s^2\}_{s=1}^p$ was defined for each dataset separately, around the rule of thumb[5]

$$\hat{\sigma}^2 = 0.1 \times d \times \mathbb{E}[\text{Var}[\mathcal{D}']] \tag{20}$$

where $\mathcal{D}' = \{x'_j\}_{j=1}^d$, $x'_j \in \mathbb{R}^N$, and $\text{Var}[\mathcal{D}'] = [\text{Var}[x'_1], \text{Var}[x'_2], \ldots, \text{Var}[x'_d]]$, where each entry corresponds to the variance of one dimension in the dataset. The factor 0.1 was selected experimentally.

Two methods were used for comparison, the maximization of the Shannon entropy of the kernel matrix [29] (Shannon), and kPA [11]. The first method only allows one to find the kernel parameter σ^2. Therefore, the number of components n_c was selected using the methodology proposed in this paper (MDD) based on the maximization of the structural information content I_s. With the second method, however, it is possible to find both model parameters simultaneously, and the results were obtained by means of a publicly available code[6] proposed in [11]. A comparison between the two algorithms for large scale kPCA is also presented in this paper.

[5]Used in the tutorial of http://www.esat.kuleuven.be/stadius/lssvmlab/.

[6]Available at http://www2.imm.dtu.dk/~kwjo/index.php?page=code.

A. Toy Examples

Three artificial datasets were created after adding noise (SNR = 10) to predefined clean patterns. These patterns were then used to evaluate the performance of the algorithms after denoising. The first comparison was made between the denoised results obtained using all the points, and using the two large scale methodologies, namely, the modified FS (FS2) and ICF. Each model selection algorithm was tested using these three methodologies.

The first dataset consists of two semicircles of $N = 500$ points contained in a space of two dimensions. For the first large scale approach proposed in this paper, namely, FS2 + MDD, the training set was selected by maximizing the quadratic Renyi entropy with a subset of 50 support vectors. The remaining 450 points were split into validation set $N_v = 250$ and test set $N_{\text{ts}} = 200$. The selected model corresponds to $\sigma^2 = 2.19$ and $n_c = 9$. For the second approach (ICF + MDD), a set of pivots was determined using ICF with $\eta = 10^{-2}$ and $\sigma^2 = 1.13$ computed using (20). The resultant number of components was $n_c = 24$ for 70 pivots and $N_v = 250$. The definition of the parameter η can help to limit the number of pivots that are selected in cases where the σ^2 is small. However, in this paper, its value was defined experimentally and no optimization was performed.

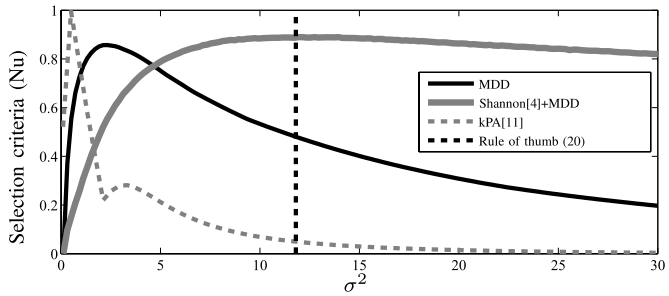


Fig. 9. Model selection criteria for different values of σ^2. Note that the kPA criteria have two local maxima, while the methodology proposed in this paper reveals a global one. The rule of thumb of (20) gets a value close to the maximum of the curve of the Shannon entropy. This rule of thumb can be taken as an upper bound for the definition of the search grid of σ^2.

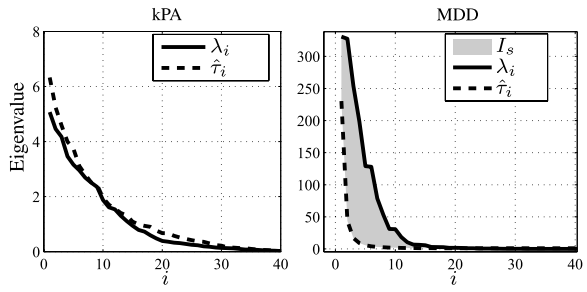


Fig. 10. Eigenvalue spectra for the square example using kPA and MDD. Note that no component is selected using kPA, because the estimated noise is always larger than the information contained in the data.

The model selection criteria are indicated in Fig. 9, and the denoised datasets for all the criteria are shown in Fig. 11. Table I presents the mean square error (MSE) measured between the original clean test patterns and the denoised test data. The results obtained using the methodologies of [11] and [29] are also indicated. The algorithm based on the maximization of the Shannon entropy was not tested together with ICF, because the kernel parameter was already defined using (20).

The second toy example corresponds to a square of 2000 points in a 2-D space, and the second one consists of two rings of 3000 points contained in a 3-D space. As in the previous experiment, 50 support vectors were selected from each dataset by means of FS2 and $N_v = 1000$. For ICF + MDD, the kernel parameter was computed using (20), and 98 pivots were selected for the square dataset, and 493 for the 3-D rings example, with $N_v = 1000$. The model parameters and the MSE are shown in Table I. The estimated noise from kPA tends to be overestimated in low dimensions (Section III); in other words, the level of noise can be larger than the level of variance explained by the components. For this reason, this method failed to select any components for the square and 3-D rings datasets. Fig. 10 shows the eigenvalues obtained for the square example using kPA and MDD. Note that for kPA, the noise estimation is always larger than the information contained in the data, which confirms what was discussed in Section III.

All toy examples with their corresponding denoised sets are indicated in Fig. 11. From the figure and Table I, it is clear that the best denoised results are obtained using MDD and all the points of the dataset. However, the performance is not compromised when FS2 is used for large-scale applications.

The computation times for the different model selection methodologies are indicated in Fig. 12. For this comparison, the kernel parameter was set to the optimal value, the size of the training set was changed and a mean computation time was obtained after running the algorithms 50 times on each dataset. Note that the computation times of ICF + MDD are lower than for FS2 + MDD; this is due to the fact that the set of approximated eigenvectors is computed from a set of pivots, which is normally much smaller than the training set. The simulations with FS2 + Shannon + MDD are not indicated since the σ^2 is fixed and the algorithm that is used in this paper to compute n_c is the same as for FS2 + MDD. In addition, the publicly available code used to implement kPA is unstable when the dataset is larger than 500 points. For this reason, a new implementation was used for these particular cases, and it was shown to be faster and more efficient than the public code (Fig. 12). The disadvantage of the public implementation is that all the 100 permutations of the input data are stored *all* in the memory, and after this, the kernel matrices and eigendecompositions are computed. To make the code more efficient, each permutation was performed separately, and once its kernel matrix and eigenvalue decomposition were computed, the input (permuted) matrix was replaced by a new permutation. The fact that the 100 input matrices are not stored in the memory makes the algorithm more efficient and able to handle more than 500 points.

In these toy examples, a certain ratio between the number of training and validation points was used. For instance, for the square example, this ratio was set to 50/1000. To evaluate how sensitive the proposed algorithm is to this ratio, the training set was selected using FS2 with SV going from 50 to 1450 points in steps of 100. The validation set was selected at random and it changed from 1450 to 50 data points in steps of 100. For each pair training-validation, 50 runs were performed where the MSE was computed on the test set. Fig. 13 shows the median and the 25th and 75th percentiles of the MSE values and the selected model parameters for different training sizes. It can be observed that when the model is trained using 350 points or more, the same kernel parameter ($\sigma^2 \approx 0.28$) can be selected, and a similar MSE can be obtained. However, the number of components n_c increases as the size of the kernel matrix increases. Therefore, the ratio between the size of training and validation sets influences the sparsity of the solution. Furthermore, it is remarkable that after selecting the training set and randomizing the validation set several times, no high variations were found. Hence, the sensitivity of the proposed approach to changes in training sizes and changes in validation data is very low.

B. Denoising of Digits

The second type of application handles the denoising of two datasets, namely, the University of California at Irvine (UCI) *multiple features*, and the U.S. Postal Service (USPS) handwritten datasets. The UCI dataset consists of 218 digit images

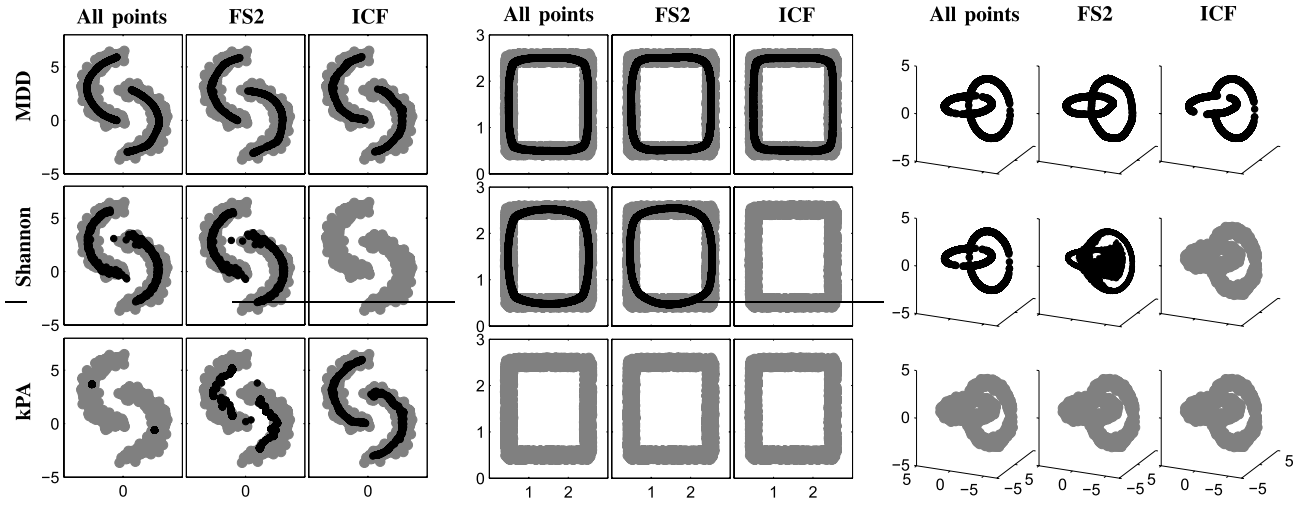


Fig. 11. Denoised datasets (black dots) after model selection. The gray dots represent the input dataset. The first three columns correspond to the semicircles, the next three to the square example, and the last three to the rings. For visualization purposes, the original rings are not indicated in MDD and Shannon. Note that there are no results for the Shannon entropy with ICF, because the kernel parameter is already defined, and the number of components is determined using MDD. For the square and the rings, the kPA was not able to find any component.

TABLE I

COMPARISON OF DIFFERENT MODEL SELECTION CRITERIA AND TWO LARGE-SCALE TECHNIQUES ON TOY EXAMPLES. THE TRAINING SETS FOR FS2 CONSISTED OF 50 DATA POINTS. RESULTS ARE INDICATED AS MSE $\pm$ STANDARD DEVIATION AND (σ^2, n_c)

		All points	FS2	ICF [16]
Semicircles ($N = 500$, $d = 2$)	**MDD**	0.063 ± 0.07 $(0.8, 33)$	0.04 ± 0.05 $(2.19, 9)$	0.09 ± 0.17 $(1.13, 24)$
	Shannon[29]+MDD	0.155 ± 0.18 $(11.6, 7)$	0.12 ± 0.14 $(11.8, 4)$	$--$ $--$
	kPA[11]	3.04 ± 2.64 $(3.11, 1)$	2.75 ± 2.16 $(12.91, 4)$	0.23 ± 0.28 $(1.13, 6)$
Square ($N = 2000$, $d = 2$)	**MDD**	0.0033 ± 0.003 $(0.28, 22)$	0.007 ± 0.009 $(0.18, 10)$	0.0025 ± 0.0022 $(0.13, 34)$
	Shannon[29]+MDD	0.0059 ± 0.0052 $(0.66, 12)$	0.011 ± 0.011 $(0.66, 6)$	$--$ $--$
	kPA[11]	$- \pm -$ $(-, 0)$	$- \pm -$ $(-, 0)$	$- \pm -$ $(0.13, 0)$
Rings ($N = 3000$, $d = 3$)	**MDD**	0.021 ± 0.030 $(1.25, 71)$	0.037 ± 0.041 $(2.51, 13)$	0.051 ± 0.10 $(1.25, 71)$
	Shannon[29]+MDD	0.040 ± 0.055 $(2.51, 45)$	0.037 ± 0.041 $(2.51, 13)$	$--$ $--$
	kPA[11]	$- \pm -$ $(-, 0)$	$- \pm -$ $(-, 0)$	$- \pm -$ $(1.25, 0)$

of 15×16 pixels, and the USPS dataset of $N = 9298$ images of 16×16 pixels. In these examples, the Gaussian noise was added to the original digits and a subset of 180 images from the UCI set and 2000 from the USPS were selected as training set using FS2. The validation sets contained 20 and 1000 images, respectively. The SNR for the UCI dataset was 8.22 and 5.51 for USPS. The MSE was measured between the original digits without noise and the denoised images. For the UCI dataset, a comparison between the different model selections and large-scale methodologies was performed. These results are presented in Table II, while Fig. 15 shows the search grid for the model parameters. Note that the solution found by MDD using all the points is indicated by the gray circle, and it is close to the minimum MSE. The performances for the USPS dataset are presented in Table III. In this case, only the results for the large-scale methods are presented due to the large size of the dataset. Some denoised test images of the UCI dataset are presented in Fig. 16, which also shows the denoised test images using the state-of-the-art denoising algorithm proposed in [33]. Note that MDD does not lead to a sparse solution compared with FS2 + kPA. In this particular case, permuting the data points as in [11] will result in a good approximation of the noise. Another way of approximating the noise is by generating several random datasets of uniformly distributed

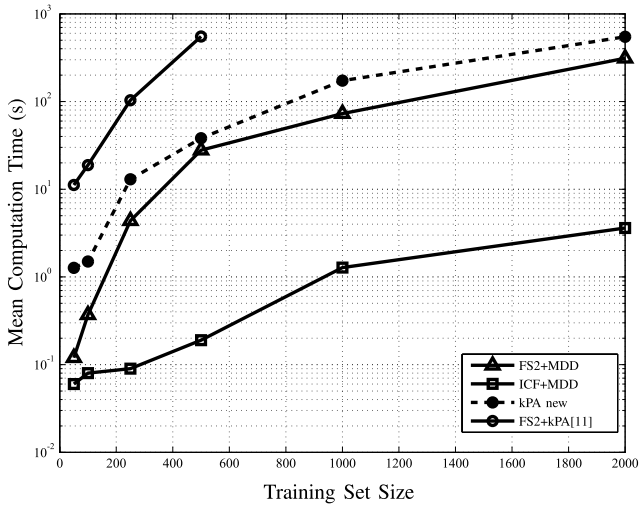


Fig. 12. Mean computation time on the toy examples. The dashed line indicates the time for the algorithm of kPA implemented in this paper, which performs better than the one in [11] (circles). The main reason why FS2 + MDD is much slower than ICF + MDD is the Monte Carlo simulations that are required to compute the noise level.

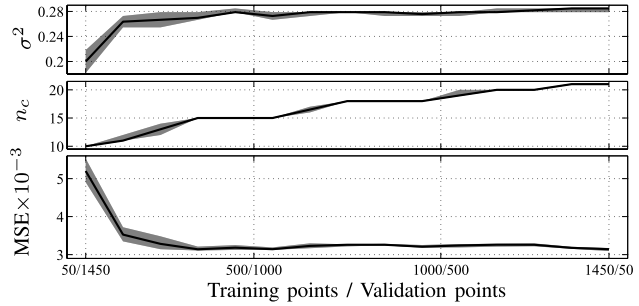


Fig. 13. Model parameters and MSE (on test set) for different ratios of training and validation points. The solid black line indicates the median for 50 independent runs, and the shaded area indicates the 25th and 75th percentiles. Note that from 350 training points on, the generalization remains almost constant. The value of n_c closely depends on the size of the kernel matrix.

points within the limits of the input dimensions. By doing this, a slightly better performance compared with kPA can be achieved, and it corresponds to $n_c = 15$ and $\sigma^2 = 14.78$ for the UCI dataset with MSE 0.09 ± 0.01, and to $n_c = 33$ and $\sigma^2 = 11.72$ for the USPS dataset with MSE 0.03 ± 0.01.

As mentioned in Section IV, it is important to use the four statistical moments of the original distance distribution, because they all contain valuable information about the structure and dimensionality of the data. For example, if in this particular application, only the first two moments of the original distance distribution are used to generate the noise distance matrix, the corresponding eigenvalue distribution for the optimal MDD σ^2 does not allow for selecting any component. This can be observed in Fig. 14(a) (solid gray line), where it is also clear that after the first component, the noise is always below the data. If now, instead of looking at the optimal MDD σ^2, the information content I_s is computed for each kernel parameter, it is observed that for some cases ($I_s = 0$), the noise may contain more information than the original data [see the solid gray line in Fig. 14(b)]. Here, the advantage of

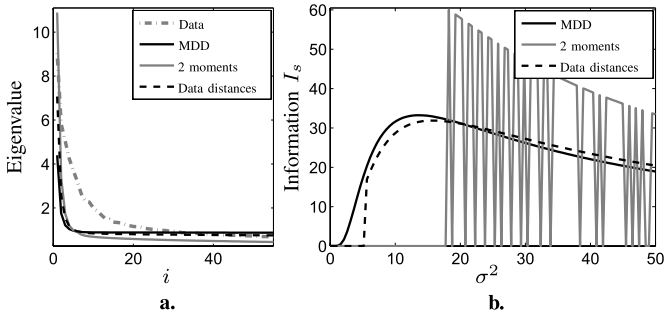


Fig. 14. (a) Eigenvalue distributions and (b) information for different noise estimation approaches. Two moments generate the noise matrix using only the first two statistical moments of the real distance distribution. This method does not provide a valid approximation for different values of σ^2. Data distances indicate the results when the entries of the noise matrix are drawn from the observed distance distribution. The last case underestimates the noise contained in the data.

TABLE II
COMPARISON OF DIFFERENT MODEL SELECTION CRITERIA AND TWO LARGE-SCALE TECHNIQUES ON THE UCI DATASET WITH $N = 218$ AND $d = 240$. THE TRAINING SETS FOR FS2 CONSISTED OF 180 DATA POINTS. THE RESULTS ARE INDICATED AS MSE $\pm$ STANDARD DEVIATION AND (σ^2, n_c)

	All points	**FS2**	**ICF [16]**
MDD	0.08 ± 0.01 $(13.48, 35)$	0.09 ± 0.01 $(13.35, 29)$	0.08 ± 0.05 $(1.8, 77)$
Shannon[23]+MDD	0.10 ± 0.01 $(17.82, 33)$	0.11 ± 0.01 $(19.07, 27)$	$--$ $--$
kPA[7]	0.10 ± 0.01 $(14.93, 16)$	0.11 ± 0.01 $(9.06, 13)$	0.08 ± 0.01 $(1.8, 82)$

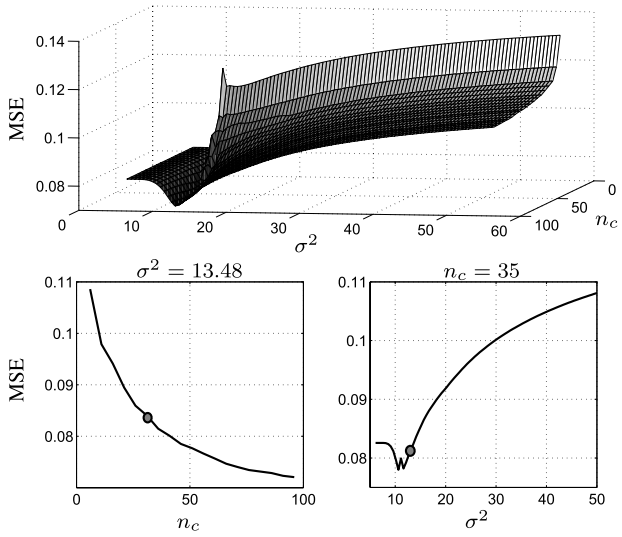


Fig. 15. Top: grid search for the model parameters. Bottom: MSE for the parameters identified by MDD using all the points in the UCI dataset. The gray circles indicate the selected parameters. Note that the selected σ^2 produces results with a performance close to the minimum of the curve. However, when looking at the number of components for the given kernel parameter, the best model tends toward nonsparse solutions. According to the curve, the best solution corresponds to $\sigma^2 = 12.65$ and $n_c = 91$.

using kurtosis and skewness becomes clear. It is important to mention that for low-dimensionality problems, using the first two moments will produce similar results than when using all

TABLE III
SIZE OF THE TRAINING SET, MODEL PARAMETERS INDICATED AS (σ^2, n_c), AND MSE $\pm$ STANDARD DEVIATION ON DENOISED TEST DATA AND ORIGINAL USPS PATTERNS

	USPS ($N = 9298$, $d = 256$)			
	Training	σ^2	n_c	MSE
FS2+MDD	2000	9.88	313	0.03 ± 0.01
ICF[21]+MDD	1361	117.35	272	0.09 ± 0.06
Shannon[29]+MDD	2000	13.26	30	0.04 ± 0.04
FS2+kPA[11]	2000	13.26	28	0.04 ± 0.04
FS2+MDD [a]	2000	11.72	33	0.03 ± 0.01

[a] Results of noise estimation after generating, several times, a dataset of uniformly distributed points within the limits of the input dimensions.

the four. This is because the lower the dimensionality of the data, the closer is the artificial distribution to a normal one.

A different approach to generate the artificial distance matrix is using the observed distance distribution. Fig. 14(a) (dashed black line) indicates the corresponding eigenvalues for the optimal MDD σ^2. Note that the first eigenvalues are larger, and this is due to the fact that more data structure is still contained in the distribution of distances. In addition, the noise tends to be underestimated. In this example, eight more components are needed for data denoising, which causes a drop in performance: 0.11 ± 0.03. If I_s is maximized for this case [see the dashed black line in Fig. 14(b)], the resulting MSE is 0.1 ± 0.01 with $n_c = 32$ and 13.37. These differences in performance are not critical, which indicates that creating the noise matrix using the observed distance distribution can be seen as an alternative, keeping in mind that the noise will tend to be underestimated.

Going back to the results obtained with the proposed approach, for the USPS dataset, it is observed that the MSE values obtained with ICF + MDD are larger than for all the other algorithms. This can be due to the fact that the kernel parameter σ^2 is computed using the rule of thumb in 20 and that the user defined error threshold η is not optimized. The reasons for this are discussed at the end of Section V.

The performance of the algorithms was evaluated for different SNRs. Fig. 17 shows the MSE and the standard deviations for different SNR applied to the UCI and USPS datasets. The results for this application were compared with the algorithm in [33]. From Fig. 17, it is clear that MDD and the large-scale methods described in this paper outperform the other algorithms, and lower errors are obtained when the SNR increases. It is also clear that the kPA and Shannon methods indicate lower performances even for high-dimensional data. In addition, for low SNR, the method in [33] performs similar to ICF + MDD, which rapidly decreases the performance as the SNR gets worse. This can also be observed in the denoised test images in Fig. 16.

C. ECG-Derived Respiration

The last application considered in this paper is the reconstruction of the respiratory signal from the electrocardiographic signal (ECG). This reconstructed signal is called



Fig. 16. (a) Noisy test UCI dataset. (b) Denoised after FS2 + MDD with $\sigma^2 = 13.35$ and $n_c = 29$. (c) ICF + MDD with $\sigma^2 = 1.8$ and $n_c = 77$. (d) FS2 + Shannon + MDD with $\sigma^2 = 19.07$ and $n_c = 27$. (e) FS2 + kPA with $\sigma^2 = 9.06$ and $n_c = 13$. (f) Kong *et al.* [33].

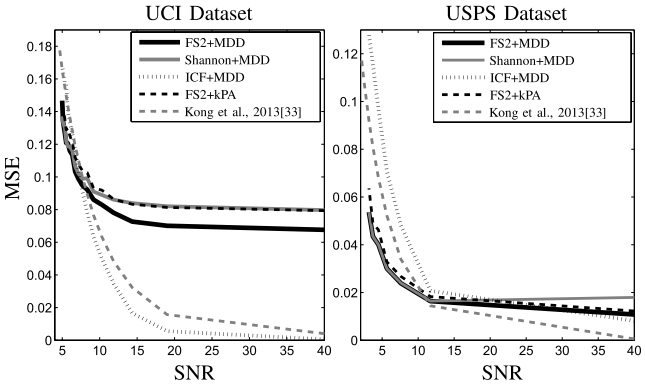


Fig. 17. MSE for different SNRs of the UCI and USPS datasets. Lowest errors are obtained with ICF + MDD and the method in [33]. However, when SNR is lower, these two methods tend to perform worse than the other algorithms. Note that the kPA and Shannon indicate low performances even for high-dimensional datasets.

the ECG-derived respiration and it has already been studied in the framework of kPCA [29], where the maximization of the Shannon entropy of the kernel matrix was proposed as a criterion to select σ^2. This particular application is included in this paper, not only for the sake of comparison between the model selection criteria, but also because the number of components is already known, namely, $n_c = 1$. In this application, it does not make sense to measure performance using the MSE, because the estimated signal may have some delays due to the nature of its computation. Therefore, different measurements like the mean magnitude squared coherence (MSC) and the correlation are used to evaluate the resemblance of the estimated respiratory signal to the real one [29].

The dataset consists of 28 arrays of 31 time series, which are modulated by the respiratory signal with d up to 349. The goal of this application is to reconstruct the respiration based on these modulations. Details of this dataset can be found in [29]. For this implementation, only the selection of the kernel parameter is evaluated because as mentioned before, $n_c = 1$. The reason for this is that the information described by the first component is related to the fundamental modulation of the time series, which corresponds to the respiratory signal.

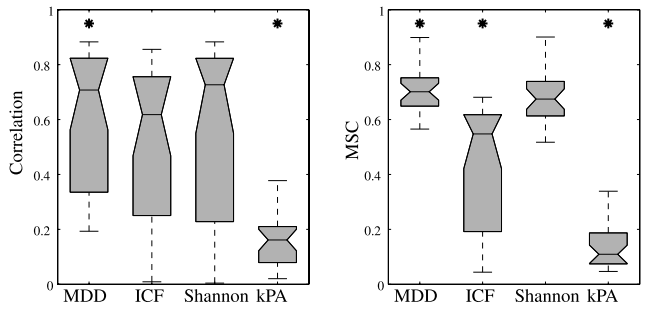


Fig. 18. Boxplots of correlation and mean MSC between the reconstructed and the original respiratory signals for all the 28 ECG segments. Note that kPA performs significantly worse than the other methods, while MDD performs slightly better than the Shannon criteria, especially when looking at the coherence (MSC) at the respiratory frequency. However, only the difference between MDD and kPA is significant (indicated by *). The values of σ^2 selected by MDD and Shannon were very similar ($p > 0.05$), which can explain the comparable values of correlation and MSC for both criteria.

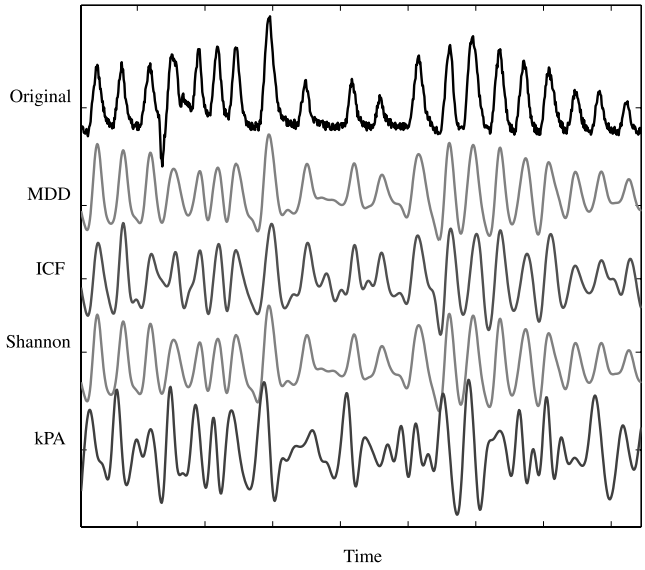


Fig. 19. Original and ECG-derived respiratory signals. Note that the signal obtained with MDD closely resembles the original signal, and this is also the case in the one computed after maximizing the Shannon entropy of the kernel matrix. For this example, the value of σ^2 selected by MDD was 188.7 and by Shannon was 139.4.

The median MSC equals $0.70(0.65, 0.74)$ for FS2 + MDD, $0.67(0.61, 0.73)$ for Shannon, and $0.1(0.07, 0.18)$ for kPA, with the results expressed as median (25th, 75th) amongst the 28 different arrays. The kernel parameter was tuned for each signal separately. The most interesting performance measure is the coherence, since it is only measured at the desired frequency band, namely, the one defined by the respiration. The box plots in Fig. 18 show the different values of MSC and correlation for the whole dataset, and a comparison between the original and the reconstructed signals is shown in Fig. 19. The differences between the performances of the different methods were evaluated with a confidence interval of 95% using the Kruskal–Wallis test and the multicomparison test with the Bonferroni correction. From Fig. 18, it is clear that kPA performs significantly worse than the other methods, and the ICF shows a much lower MSC than Shannon and MDD. Note that the signals obtained with MDD and Shannon closely resemble the original respiratory signal; this is because the values of σ^2 selected with both criteria were not significantly different ($p > 0.05$). However, the MSC values were larger for MDD than for Shannon, in 23 out of 28 cases. This indicates that a slight improvement in performance can be obtained with MDD. A reason to use Shannon instead of MDD in this particular application could be the computation time. Since the number of components is already known, one only needs to maximize the Shannon entropy of the kernel matrix.

VII. Conclusion

This paper presented a criterion to select the model parameters σ^2 for RBF kernels and n_c in kernel PCA. The main contribution of this paper relies on the estimation of the noise by generating an artificial distance distribution based on the original dataset. This new distribution allows the noise to be estimated without generating any artificial dataset. It was shown that this method outperforms other criteria for model selection, especially in lower dimensional spaces where other methods deem to fail. Based on the findings presented in this paper, MDD can be used both in low- and high-dimensional spaces, but its advantages are better exploited in low dimensions.

In addition to the model selection criterion, this paper proposed a modification to the FS method, which can be used to reduce the size of the training set. This method, together with the ICF, was used to approximate the kernel matrix in large-scale applications.

The strength of the methodology proposed in this paper is that it is able to retrieve the model parameters in any dimensional space and that it can be useful in many applications. One clear drawback is the implementation of a Monte Carlo simulation, which can be seen as a bottleneck when online applications are considered. Future studies may consider avoiding this implementation.

Further research might explore the application of MDD in different unsupervised methods like clustering and in supervised techniques like regression.

Appendix
Computation of the Distance Matrix $\hat{R}$

When a set of values is drawn from the distribution $\hat{f}_{\mathcal{N}}(\hat{r})$, these values can be organized in the matrix $\hat{R}$ in such a way that the entry $\hat{R}_{ij} = \hat{r}(x_i, x_j)$ is always lower or equal than the minimum value of the pairwise sum of the columns i and j. This is shown as

$$\hat{R}_{ij} \leq \min(\hat{R}_i + \hat{R}_j) \tag{21}$$

where $\hat{R}_i$ and $\hat{R}_j$ correspond to the ith and jth column of the matrix $\hat{R}$. Once this condition is satisfied, one can say that the matrix $\hat{R}$ satisfies the triangle inequality $\forall ij$. The next example shows how to generate this matrix.

For a dataset of N data points, both distance matrices R (original) and $\hat{R}$ (generated) have dimensionality $N \times N$. It is clear that the diagonal consists of only zeros and that they are both symmetric matrices. Therefore, it is sufficient to generate $N_r = (N^2 - N)/2$ data points $\{\hat{r}_i\}_{i=1}^{N_r}$ from $\hat{f}_{\mathcal{N}}(\hat{r})$, where $\hat{r}_1 \geq \hat{r}_2 \geq \cdots \geq \hat{r}_{N_r}$. These values can then be

organized in the off-diagonal entries of the lower triangular matrix of $\hat{R}$

$$\hat{R}_L = \begin{bmatrix} 0 & & & & & \\ \hat{r}_1 & 0 & & & & \\ \hat{r}_2 & \hat{r}_N & 0 & & & \\ \hat{r}_3 & \hat{r}_{N+1} & \hat{r}_{2N-2} & \cdots & & \\ \vdots & \vdots & \vdots & \vdots & 0 & \\ \hat{r}_{N-1} & \hat{r}_{(2N-3)} & \hat{r}_{(3N-4)} & \cdots & \hat{r}_{N_r} & 0 \end{bmatrix}.$$

Once this matrix has been created, the condition in (21) is satisfied.

Keep in mind that a distance matrix usually does not satisfy the order imposed in the approach described above. This order is introduced here because it allows for speeding up the creation of $\hat{R}$. However, to guarantee that the eigenvalue distribution of a kernel matrix computed using $\hat{R}$ is not affected by this ordering, a distance matrix generated in an iterative way from $\hat{f}_{\mathcal{N}}(\hat{r})$ was computed. During the generation of this matrix, each entry of $\hat{R}$ was checked in each iteration. The eigenvalue distribution of the kernel using this matrix was then compared with the one obtained from the ordered distance matrix. This was done for different datasets and different number of points, and it was found that both distributions do not differ significantly ($p > 0.05$). Hence, using the fast and efficient approach described in this section, it is possible to keep the eigenvalue distribution and, at the same time, satisfy the different conditions in (11). It is necessary to emphasize that both matrices were created using the same unimodal distribution.

REFERENCES

[1] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 1986.

[2] Z. Fan *et al.*, "Modified principal component analysis: An integration of multiple similarity subspace models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1538–1552, Aug. 2014.

[3] A. Papaioannou and S. Zafeiriou, "Principal component analysis with complex kernel: The widely linear model," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1719–1726, Sep. 2014.

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

[5] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," in *Advances in Neural Information Processing Systems*, M. J. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge, MA, USA: MIT Press, 1999.

[6] J. T.-Y. Kwok and I. W. Tsang, "The pre-image problem in kernel methods," *IEEE Trans. Neural Netw.*, vol. 15, no. 6, pp. 1517–1525, Nov. 2004.

[7] C. M. Bishop, "Bayesian PCA," in *Advances in Neural Information Processing Systems*, M. J. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge, MA, USA: MIT Press, 1999.

[8] J. Li and D. Tao, "Simple exponential family PCA," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 485–497, Mar. 2013.

[9] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[10] C. Alzate and J. A. K. Suykens, "Kernel component analysis using an epsilon-insensitive robust loss function," *IEEE Trans. Neural Netw.*, vol. 19, no. 9, pp. 1583–1598, Sep. 2008.

[11] K. W. Jørgensen and L. K. Hansen, "Model selection for Gaussian kernel PCA denoising," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 163–168, Jan. 2012.

[12] J. Rissanen, "MDL denoising," *IEEE Trans. Inf. Theory*, vol. 46, no. 7, pp. 2537–2543, Nov. 2000.

[13] A. Hinneburg, C. C. Aggarwal, and D. A. Keim, "What is the nearest neighbor in high dimensional spaces?" in *Proc. 26th Int. Conf. Very Large Data Bases*, Cairo, Egypt, Sep. 2000, pp. 506–515.

[14] M. Köppen, "The curse of dimensionality," in *Proc. 5th Online World Conf. Soft Comput. Ind. Appl. (WSC5)*, Sep. 2000, pp. 4–8.

[15] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'nearest neighbor' meaningful?" in *Database Theory*, C. Beeri and P. Buneman, Eds. Berlin, Germany: Springer-Verlag, 1999.

[16] S. Lespinats, M. Verleysen, A. Giron, and B. Fertil, "DD-HDS: A method for visualization and exploration of high-dimensional data," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1265–1279, May 2007.

[17] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. Cambridge, MA, USA: MIT Press, 2001.

[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.

[19] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. Singapore: World Scientific, 2002.

[20] C. Alzate and J. A. K. Suykens, "Sparse kernel models for spectral clustering using the incomplete Cholesky decomposition," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Hong Kong, Jun. 2008, pp. 3556–3563.

[21] C. Alzate and J. A. K. Suykens, "Sparse kernel spectral clustering models for large-scale data analysis," *Neurocomputing*, vol. 74, no. 9, pp. 1382–1390, 2011.

[22] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *J. Mach. Learn. Res.*, vol. 3, pp. 1–48, Jul. 2003.

[23] M. Li, W. Bi, J. T. Kwok, and B.-L. Lu, "Large-scale Nyström kernel matrix approximation using randomized SVD," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 152–164, Jan. 2015.

[24] J. A. K. Suykens, T. Van Gestel, J. Vandewalle, and B. De Moor, "A support vector machine formulation to PCA analysis and its kernel version," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 447–450, Mar. 2003.

[25] J. L. Horn, "A rationale and test for the number of factors in factor analysis," *Psychometrika*, vol. 30, no. 2, pp. 179–185, 1965.

[26] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton, NJ, USA: Princeton Univ. Press, 1961.

[27] M. Girolami, "Orthogonal series density estimation and the kernel eigenvalue problem," *Neural Comput.*, vol. 14, no. 3, pp. 669–688, 2002.

[28] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.

[29] D. Widjaja, C. Varon, A. C. Dorado, J. A. K. Suykens, and S. Van Huffel, "Application of kernel principal component analysis for single-lead-ECG-derived respiration," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 4, pp. 1169–1176, Apr. 2012.

[30] M. E. Tipping, "Sparse kernel principal component analysis," in *Advances in Neural Information Processing Systems*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. Cambridge, MA, USA: MIT Press, 2001.

[31] S. Fine and K. Scheinberg, "Efficient SVM training using low-rank kernel representations," *J. Mach. Learn. Res.*, vol. 2, pp. 243–264, Mar. 2002.

[32] C. Alzate and J. A. K. Suykens, "A regularized kernel CCA contrast function for ICA," *Neural Netw.*, vol. 21, nos. 2–3, pp. 170–181, 2008.

[33] X. Kong, K. Li, Q. Yang, L. Wenyin, and M.-H. Yang, "A new image quality metric for image auto-denoising," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, USA, Dec. 2013, pp. 2888–2895.



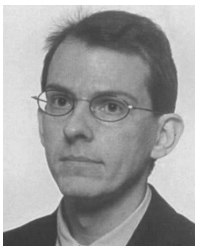
Carolina Varon (M'11) received the Professional degree in electronics engineering from the Universidad de Ibagué, Ibagué, Colombia, in 2005, and the M.Sc. degree in astronomy and astrophysics and the M.Sc. degree in artificial intelligence from Katholieke Universiteit Leuven, Leuven, Belgium, in 2009 and 2010, respectively, where she is currently pursuing the Ph.D. degree with the Department of Electrical Engineering.

She was a Site Engineer with Agroindustrial del Tolima, Ibagué, from 2002 to 2003. In 2005, she joined Security Solutions, Bogotá, Colombia, where she was the Technical Support Specialist for Latin America until 2007. Her current research interests include machine learning, kernel methods, and signal processing.



Carlos Alzate received the B.Sc. degree in electronics engineering from the Universidad Nacional de Colombia, Bogotá, Colombia, in 2002, and the master's degree in artificial intelligence and the Ph.D. degree in engineering from Katholieke Universiteit Leuven (KU Leuven), Leuven, Belgium, in 2004 and 2009, respectively.

He held a Research Foundation–Flanders Post-Doctoral Fellowship with the Department of Electrical Engineering, KU Leuven, from 2010 to 2012. He is currently a Research Scientist with the Smarter Cities Technology Centre, IBM Research—Ireland, Dublin, Ireland. His current research interests include machine learning, kernel methods, unsupervised and semisupervised learning, optimization, and natural language processing.



Johan A. K. Suykens (M'02–SM'04–F'14) was born in Willebroek, Belgium, in 1966. He received the M.Sc. degree in electromechanical engineering and the Ph.D. degree in applied sciences from Katholieke Universiteit Leuven (KU Leuven), Leuven, Belgium, in 1989 and 1995, respectively.

He was a Visiting Post-Doctoral Researcher with the University of California at Berkeley, Berkeley, CA, USA, in 1996. He has been a Post-Doctoral Researcher with Fund for Scientific Research, Flanders, Belgium, and is currently a Professor with KU Leuven. He has authored the books entitled *Artificial Neural Networks for Modeling and Control of Nonlinear Systems* (Kluwer Academic Publishers) and *Least Squares Support Vector Machines* (World Scientific), and co-authored a book entitled *Cellular Neural Networks, Multi-Scroll Chaos and Synchronization* (World Scientific).

Dr. Suykens was a recipient of the IEEE Signal Processing Society Best Paper (Senior) Award in 1999, and several best paper awards at international conferences. He was also a recipient of the International Neural Networks Society Young Investigator Award for significant contributions in the field of neural networks in 2000 and the ERC Advanced Grant in 2011. He served as the Director and an Organizer of the NATO Advanced Study Institute on Learning Theory and Practice, Leuven, in 2002, the Program Co-Chair of the International Joint Conference on Neural Networks in 2004 and the International Symposium on Nonlinear Theory and its Applications in 2005, an Organizer of the International Symposium on Synchronization in Complex Networks in 2007, and a Co-Organizer of the Neural Information Processing Systems Workshop on Tensors, Kernels, and Machine Learning in 2010. He served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1997 to 1999 and 2004 to 2007, and the IEEE TRANSACTIONS ON NEURAL NETWORKS from 1998 to 2009.