Journal of Computer Science 10 (7): 1139-1150, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.1139.1150 Published Online 10 (7) 2014 (http://www.thescipub.com/jcs.toc)

HYPERPARAMETER SELECTION IN KERNEL PRINCIPAL COMPONENT ANALYSIS

[1]Md. Ashad Alam and [2]Kenji Fukumizu

[1]Department of Statistical Science, Graduate University for Advanced Studies, Tokyo, Japan
[2]The Institute of Statistical Mathematics, Tokyo, Japan

Received 2013-12-02; Revised 2014-01-16; Accepted 2014-02-15

ABSTRACT

In kernel methods, choosing a suitable kernel is indispensable for favorable results. No well-founded methods, however, have been established in general for unsupervised learning. We focus on kernel Principal Component Analysis (kernel PCA), which is a nonlinear extension of principal component analysis and has been used electively for extracting nonlinear features and reducing dimensionality. As a kernel method, kernel PCA also suffers from the problem of kernel choice. Although cross-validation is a popular method for choosing hyperparameters, it is not applicable straightforwardly to choose a kernel in kernel PCA because of the incomparable norms given by different kernels. It is important, thus, to develop a well-founded method for choosing a kernel in kernel PCA. This study proposes a method for choosing hyperparameters in kernel PCA (kernel and the number of components) based on cross-validation for the comparable reconstruction errors of pre-images in the original space. The experimental results on synthesized and real-world datasets demonstrate that the proposed method successfully selects an appropriate kernel and the number of components in kernel PCA in terms of visualization and classification errors on the principal components. The results imply that the proposed method enables automatic design of hyperparameters in kernel PCA.

Keywords: Kernel Principal Component Analysis, Pre-Image, Kernel Choice, Cross-Validation

1. INTRODUCTION

Dimension reduction is an essential part of modern data analysis, where we often need to handle large dimensional data. The purpose of dimension reduction may be visualization, noise reduction and pre-processing for further analysis. Among others, the Principal Component Analysis (PCA), (Pearson, 1901) is one of the most famous methods to reduce the dimensionality by projecting data onto a low-dimensional subspace with largest variance.

Kernel Principal Component Analysis (kernel PCA) (Scholkopf *et al*., 1998) has been proposed as a nonlinear extension of the standard PCA and has been applied to various purposes including feature extraction, denoising and pre-processing of regression. Kernel PCA is an example of the so-called kernel methods (Scholkopf and Smola, 2002), which aim to extract nonlinear features of the original data by mapping them into a high-dimensional feature space Reproducing Kernel Hilbert Space (RKHS). This mapping is called feature map. A number of methods have been proposed as kernel methods, which include Support Vector Machine (SVM), (Boser *et al*., 1992), kernel ridge regression (Saunders *et al*., 1998), kernel canonical correlation analysis (Akaho, 2001; Bach and Jordan, 2002; Alam *et al*., 2010), A novel multiclass SVM algorithm using mean reversion and coefficient of variance (Premanode *et al*., 2013) and so on.

It is well known that the performance of a kernel method is dependent highly on the choice of kernel. For supervised learning such as SVM and kernel ridge regression, cross-validation is popularly used for choosing the hyperparameters of a kernel algorithm, such as parameters in a kernel (e.g., bandwidth of Gaussian RBF kernel), with the objective function of learning. On

Corresponding Author: Md. Ashad Alam, Department of Statistical Science, Graduate University for Advanced Studies, Tokyo, Japan

the other hand, no well-founded methods have been proposed in general for unsupervised learning such as kernel PCA and kernel canonical correlation analysis.

This study focuses on kernel PCA and proposes a method for choosing hyperparameters: Parameters in a kernel and the number of kernel principal components. In the case of standard linear PCA, the algorithm can be formulated as minimization for self-regression with reduced rank and cross-validation approaches have been proposed for choosing the number of components (Krzanowski, 1987; Wold, 1978). In contrast, while a similar regression formulation is possible for kernel PCA, the crossvalidation approach is not applicable straightforwardly for choosing a kernel in kernel PCA: The error of the regression is given by the RKHS norm of the feature space associated with the kernel and thus the cross-validation errors are not comparable for different kernels.

As detailed in section 2, the proposed method for choosing the hyperparameters of kernel PCA uses crossvalidation for the reconstruction errors of pre-images in the original space. The pre-image of a feature vector is defined by an approximate inverse image of the feature map (Mika *et al*., 1999). Various methods have been already proposed to calculate the pre-image of a feature vector, as explained in section 2.1 (Mika *et al*., 1999; Kwok and Tsang, 2003; Bakir *et al*., 2004; Rathi *et al*., 2006; Arias *et al*., 2007; Zheng *et al*., 2010). In the proposed method, given an evaluation data in the cross-validation, we compute the pre-image of the corresponding feature vector projected onto the subspace given by kernel PCA and then evaluate the reconstruction error of the evaluation point. A kernel and the number of components corresponding to the minimum average reconstruction error are chosen as the optimum ones. We demonstrate the effectiveness of this method experimentally with various synthesized and real-world datasets.

1.1. Kernel PCA

In kernel methods, the nonlinear feature map is given by a positive definite kernel, which provides nonlinear methods for data analysis with efficient computation. A symmetric kernel k(.,.) defined on a space x is called positive definite if for arbitrary number of points $x_1,\ldots, x_n \in x$ the Gram matrix $(k(x_i, x_j))_{ij}$ is positive semi-definite. It is known (Aronszajn, 1950) that a positive definite kernel k is associated with a Hilbert space H, called Reproducing Kernel Hilbert Space (RKHS), consisting of functions on x so that the function value is reproduced by the kernel; namely, for any function $f \in H$ and point $x \in x$, the function value f (x) is given by:

$$f(x) = \langle f(.), k(.,x) \rangle_H \quad (1)$$

where, $\langle,\rangle_H$ in the inner product of H. Equation 1 is called the reproducing property. Replacing f with $k(.,\tilde{x})$ yields $k(x,\tilde{x}) = \langle k(.,x), k(.,\tilde{x}) \rangle_H$ for any $x,\tilde{x} \in x$.

To transform data for extracting nonlinear features, the mapping Φ: $x \rightarrow H$ is defined by: $\Phi(x) = k(.,x)$. Which is regarded as a function of the first argument. This map is called feature map and the vector Φ (x) in H is called feature vector. The inner product of two feature vectors is then given by $\langle \Phi(x), \Phi(\tilde{x}) \rangle_H = k(x,\tilde{x})$. This is known as the kernel trick, serving as a central equation in kernel methods. By this trick the kernel can evaluate the inner product of any two feature vectors efficiently without knowing an explicit form of either $\Phi(.)$ or H. With this computation of inner product, many linear methods of classical data analysis can be extended to nonlinear ones with efficient computation based on Gram matrices. Once Gram matrices are computed, the computational cost does not depend on the imensionality of the original space.

Kernel PCA (Scholkopf *et al*., 1998) conducts principal component analysis for the feature vectors. More precisely, given data points $X_i \in x$, i = 1, 2,..,n, kernel PCA outputs a set of principal functions by the following two-step procedure: (i) transform the data nonlinearly into the feature space H, i.e., $X_i \rightarrow \Phi(X_i)$, (ii) solve the linear PCA problem for the feature vectors, i.e., solve the directions in H for which the variance of $\{\Phi(X_i)\}$ along those directions is maximized. The algorithm of kernel PCA is described as follows (Scholkopf *et al*., 1998). Let $\tilde{\Phi}(X) := \Phi(X) - \frac{1}{n}\sum_{j=1}^{n}\Phi(X_j)$ be the centered feature vector. The estimated covariance matrix is given by $G = \frac{1}{n}\sum_{i=1}^{n}\tilde{\Phi}(X_i)\tilde{\Phi}(X_i)^T$ with the centered feature vectors. The principal directions $g \in H$ are given by the unit eigenvectors corresponding to the largest eigenvalues and thus the problem is converted to solving the eigenequation:

$$Gg = \tilde{\lambda} g$$

By using the kernel trick, this problem is reduced to the generalized eigenproblem that finds $g = \sum_{i=1}^{n} \alpha_i \tilde{\Phi}(X_i)$ such that Equation 2:

$$M\alpha = n\tilde{\lambda}\alpha, \quad \text{subject to } \alpha^T M\alpha = 1 \tag{2}$$

where, M is the n×n centered Gram matrix defined by M = CKC with $K_{ij} = k(X_i, X_j)$ and $C = I_n - \frac{1}{n}1_n 1_n^T$. Here In is the identity matrix of size n and 1_n is the vector with n ones. The constraint $\alpha^T M\alpha = 1$ corresponds to the condition $\langle g_i, g_h \rangle = \delta_{jh}$, where δ_{jh} is the Kronecker's delta.

Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ denote the ordered eigenvalues of M with associated eigenvectors $\alpha_1, .., \alpha_n$, where $\alpha_j = (\alpha_{1j} \ldots \alpha_{nj})^T$. The vectors are normalized so that $\alpha_j^T M_{\alpha h} = \delta_{jh}$. The j-th principal direction $g_j \in H$ is then given by:

$$g_j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{n} \alpha_{ij} \tilde{\Phi}(X_i)$$

and the j-th principal component of the data point X_i is given by:

$$\left\langle g_i, \tilde{\Phi}(X_i) \right\rangle = \frac{1}{\sqrt{\lambda_j}} (M_{\alpha j}) = \sqrt{\lambda_j}\alpha_{ij}$$

For a test point X out of the sample, the j-th principal component is similarly given by:

$$\left\langle g_j, \tilde{\Phi}(x) \right\rangle = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{n} \tilde{k}(x, x_i)\alpha_{ij}$$

where, $\tilde{k}(x,y) = k(x,y) - \frac{1}{n}\sum_{i=1}^{n} k(x, X_i) - \frac{1}{n}\sum_{i=1}^{n} k(X_i, y) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(X_i, X_i)$ is the centered kernel.

1.2. Choice of Kernel

The result of kernel PCA obviously depends on the choice of kernel. It is often the case that the kernel has some parameters like the popular examples shown in **Table 1**. In such a case, these parameters may have strong influence on the results. To depict the influence, using Wine data (see section 3) we show the plots of the first two kernel principal components with different values of inverse band width parameter s in Gaussian RBF kernel and degree d and constant c in the polynomial kernel (**Fig. 1**). From the figure, we see that in both the kernels the results of kernel PCA depend strongly on the parameters and an appropriate choice is indispensable for the method to give reasonable low-dimensional representation of data.

It is known that the standard PCA can be formulated as a self-regression or reconstruction problem; namely, the first r principal components of centered data $\{X_i\}_1^n \subset \mathbb{R}^d$ are equal to the projections BX_i given by the reduced rank regression:

$$\min_{A,B} \sum_{i=1}^{n} \left\| X_i - ABX_i \right\|^2 \text{ Subject to } BB^T = I_r$$

where, A and B are d×r and r×d matrices, respectively. Based on this regression formulation, the cross validation approach (Stone, 1974) has been used for the standard PCA to choose the number of components (Wold, 1978; Krzanowski, 1987) by minimizing the above self-regression errors.

In a similar manner, the kernel PCA can be also formulated as the self-regression of the centered feature vectors. In fact, it is easy to see that the first r principal directions are given by:

$$\min_{f_j, g_j \in H} \sum_{i=1}^{n} \left\| \tilde{\Phi}(X_i) - \sum_{j=1}^{r} f_j \left\langle g_j, \tilde{\Phi}(X_i) \right\rangle \right\|_H^2$$

where, $f_j, g_j \in H$ with $\langle g_i, g_l \rangle = \delta_{jl}$. One might expect that this self-regression formulation could be applied to the cross-validation method for choosing a kernel in kernel PCA. This is not possible, however, because the above regression error is measured by the RKHS norm given by the kernel and thus the errors are not comparable among different kernels.

The goal of this study is thus to propose a method of choosing a kernel (and the number of components) in kernel PCA by introducing a criterion comparable for different kernels.

Table 1. Examples of popular kernels with parameters

Name of kernel	$k(x, \tilde{x})$	Parameter
Gaussian RBF kernel	$e^{-s\|x-\tilde{x}\|^2}$	s>0
Polynomial kernel	$(\langle x, \tilde{x} \rangle + c)^d$	$c \geq 0, d \in \mathbb{N}$

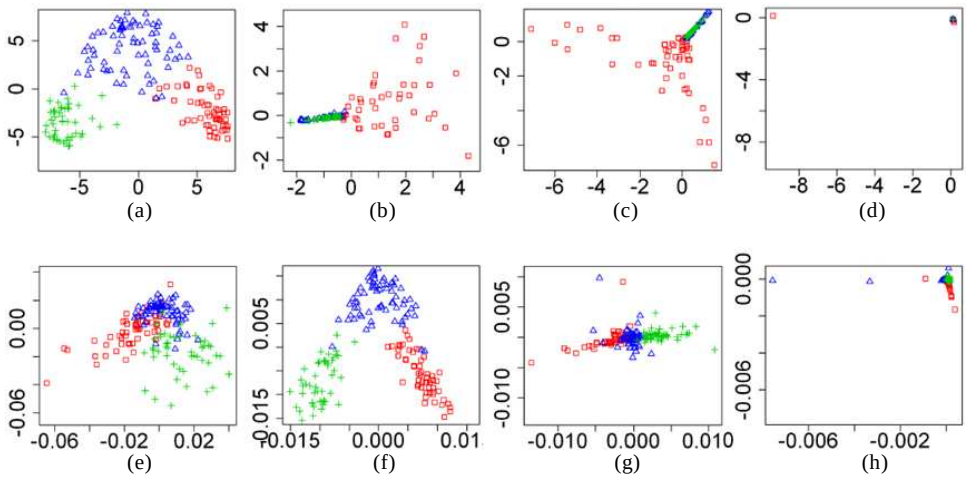

Fig. 1. Plots of the first two kernel principal components for wine data: Gaussian RBF kernel is used in the top panel (a) s = 0.05 (b) s = 0.75 (c) s = 1 (d) s = 10 and polynomial kernel in the bottom (e) c = 0.001, d = 2 (f) c = 10, d = 2 (g) c = 1, d = 3 (h) c = 1, d = 4

2. MATERIALS AND METHODS

The proposed method for choosing a kernel and the number of components uses cross-validation by the comparable reconstruction errors in the original space. To evaluate the errors, we need to solve the pre-image of the feature vectors projected on the subspace given by the principal directions. We first give a brief review of pre-image methods.

2.1. Pre-Image of Kernel PCA

While many kernel methods provide their output in the form of feature vectors in the RKHS, in some problems we want to find a point in the original space. Mika *et al*. (1999), kernel PCA is applied to a denoising task, in which an image corresponding to the RKHS vector obtained by kernel PCA is used as a denoised version of the original image.

Given a vector f in RKHS H, it is in general not possible to find a rigorous pre-image, that is a point X in the original space such that $\Phi(X) = f$ holds exactly. We thus define an (approximate) pre-image of f by the minimize of:

$$\min_{Z\in H}\left\|f-\Phi(Z)\right\|_H$$

In the original paper, Mika *et al*. (1999) have used the fixed-point iterative method. Many other approaches have been also proposed to solve the pre-image problem. A non-iterative approach of distance constraint has been proposed by Kwok and Tsang (2003), while it is dependent on the choice of neighborhood. An approach of learning a pre-image map was developed by Bakir *et al.* (2004). To apply this technique, we need an additional regularization parameter. Some authors have extended these approaches in different ways (Rathi *et al*., 2006; Arias *et al*., 2007; Zheng *et al*., 2010). More recently, a two-stage closed-form approach has been also proposed (Honeine and Richard, 2011). These advanced methods, however, usually require some tuning parameters. We use the fixed-point method in our proposed method, since it has a simple form for Gaussian RBF kernel.

We here explain the fixed-point method for solving the pre-image problem in the kernel PCA setting. Let: $X_1, X_2, \ldots\ldots, X_n \in \mathbb{R}^m$ be the training data for kernel PCA and $g_j = \sum_i \alpha_{ji}\tilde{\Phi}(X_i)(j=1,\ldots,l)$ be the unit principal directions. The projector onto the subspace spanned by $\{g_j\}_{j=1}^{l}$ is denoted by P_l, i.e., $P_l f = \sum_{j=1}^{l}\langle f, g_j\rangle g_j$. Given test point X in the original space, x the feature vector projected onto the principal subspace is given by $P_l\,\tilde{\Phi}(X)$. The pre-image of this vector in the RKHS is defined by the minimizer of Equation 3:

$$\rho(Z) = \left\|P_l\tilde{\Phi}(X) - \tilde{\Phi}(Z)\right\|_H^2 \tag{3}$$

It is easy to see that:

$$\rho(Z) = \left\|\tilde{\Phi}(Z)\right\|_H^2 - 2\left\langle\tilde{\Phi}(Z), P_l\tilde{\Phi}(X)\right\rangle_H + \left\|P_l\tilde{\Phi}(X)\right\|_H^2 = \tilde{k}(Z,Z) - 2\sum_{i=1}\gamma_i\tilde{k}(Z,X_i) + \Omega \tag{4}$$

where, $\gamma_i = \sum_{j,h} \alpha_{ji}\alpha_{jh}\tilde{k}(X_h, X)$ and Ω is a constant independent of Z.

For Gaussian RBF kernel, Equation 4 is equal to $\rho(Z) = 1 - 2\sum_{i=1}^{n} \gamma_i e^{-s\|X_i - Z\|^2} + \Omega$ and by setting the derivative zero we obtain the fixed-point algorithm:

$$Z_{t+1} = \frac{\sum_{i=1}^{n} \gamma_i e^{-s\|X_i - Z_t\|^2}}{\sum_{j=1}^{n} \gamma_i e^{-s\|X_i - Z_t\|^2}} X_i = \frac{\sum_{i=1}^{n} a_i X_i}{\sum_{j=1}^{n} a_j} \quad (5)$$

where, $\alpha_i = \gamma_i e^{-s\|X_i - Z_t\|^2}$.

In the case of polynomial kernels, the fixed point condition does not derive such an iterative form as the Gaussian RBF kernel. We thus use the steepest descent method for Equation 4 in our experiments on polynomial kernels in section 3.4.

2.2. Method for Hyperparameter Choice

For the objective function of cross-validation, we use reconstruction errors between a test point X and the corresponding pre-image Z of the projected feature vector $P_l \tilde{\Phi}(X)$ given by kernel PCA. The reconstruction errors are measured by the distance of the original space $x = \mathbb{R}^m$. By this approach, unlike the regression error in the RKHS, we can consider comparable errors for different kernels. The architecture of the proposed method is given in **Fig. 2**. The algorithm of the kernel choice in kernel PCA is given in **Fig. 3**. We describe the Leave-One-Out Cross Validation (LOOCV) for simplicity, but the extension to the general K-fold cross-validation is straightforward. By a similar algorithm we are able to select the number of principal components or any other hyperparameters.

In solving approximate pre-images, the fixed-point or the steepest descent method may be trapped by local minima. To avoid this problem, we use five initial points for the optimization algorithm and choose the best one. As shown in the next section, the obtained pre-images give appropriate results.

Note also that the fixed-point method may not work well for a very large inverse-bandwidth s, since the term of the nearest X_i is dominant in the right hand side of Equation 5 so that Z_t may stay at X_i. In the experiments, we set a reasonable parameter range of s by checking the kernel PCA results with two components.

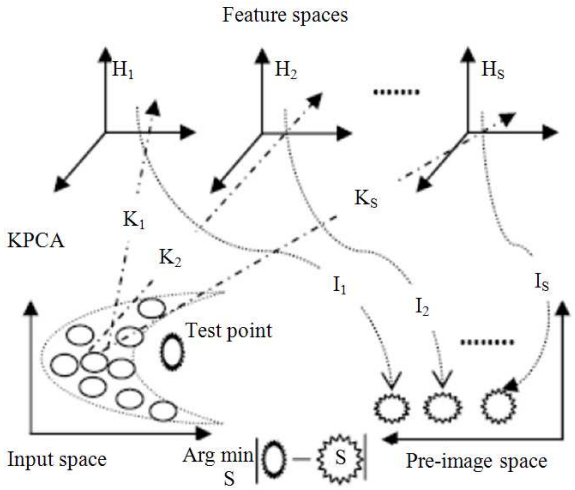

Fig. 2. Architecture of kernel choice in kernel PCA

Input: $D = \{\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_n\}$ in $\mathbb{R}^m$. Parameters $\{s_1, \ldots, s_T\}$ for kernel k_s. Threshold TH.

1. Set $h = 1$.
2. Do the following steps:
   (1) Set $i = 1$.
   (2) Solve kernel PCA for $D - \{\mathbf{X}_i\}$ with kernel k_{s_h} (Eq. 2).
   (3) Compute the approximate pre-image $\mathbf{Z}_i^h$ for $\mathbf{X}_i$ using the fixed-point method Eq.(4). The iteration stops if $\|\mathbf{Z}_{t+1} - \mathbf{Z}_t\| < TH$.
   (4) Compute the reconstruction error $E_i^h = \|\mathbf{X}_i - \mathbf{Z}_i^h\|^2$.
   (5) $i := i + 1$.
   (6) If $i > n$, BREAK; otherwise go to (2).
3. Compute the LOOCV error $E^h = \frac{1}{n}\sum_{i=1}^{n} E_i^h$.
4. $h := h + 1$.
5. If $h > T$, END. Otherwise, go to 2.
6. $h_{opt} := \arg\min_h E^h$.

Output: $s_{opt} := s_{h_{opt}}$.

Fig. 3. Algorithm of kernel choice in kernel PCA with Gaussian RBF kernel

3. RESULTS

We apply the proposed method for choosing the parameters in a kernel and the number of principal

components in kernel PCA for various datasets. Gaussian RBF kernel is used except section 3.4, where polynomial kernel is discussed. We use two synthesized and seven real-world datasets, which are summarized in **Table 2**. For the real world datasets, we standardize each variable of data before applying kernel PCA. In solving pre-images, we take initial values from the uniform distribution on the interval [-1; 1]. The detailed discussions on the results will be shown in section 4.

3.1. Synthesized Data

We use two synthesized datasets to illustrate the effectiveness of the proposed method. Each dataset is of two dimension and have three clusters.

Synthesized data-1. About 175 data are generated along three circles of different radii with small noise:

$$X_i = r_i \begin{pmatrix} \cos(Z_i) \\ \sin(Z_i) \end{pmatrix} + \in_i \qquad (6)$$

where, r_i = 1, 0.5 and 0.25, for i = 1,…,100, i = 101,….,150 and i = 151,….,175, respectively, $Z_i \sim U[-\pi, \pi]$ and $\in_i \sim N(0, 0.01\ I_2)$ independently.

Synthesized data-2. This is an example taken from (Scholkopf and Smola, 2002). The dataset has 150 points, which consists of 50 points from each of three Gaussian distributions with means (-0.5, -0.1), (0, 0.7) and (0.5, 0.1) and variance 0.1.

We prepare the inverse bandwidth parameters s∈{0.05, 1, 5, 10, 25, 50} and s∈{1, 5, 10, 20, 50, 100, 200} for Synthesized data-1 and Synthesized data-2, respectively and calculate the LOOCV reconstruction errors by pre-images. To see the variations over sampling, we generate 100 samples for each case of data 1 and 2 and make boxplots. **Figure 4** shows (a): Scatter plots of a sample of the original datasets, (b): The boxplots and (c,d): The scatter plots of first two kernel principal components with the best kernel bandwidths (c) and with other ones (d). We can see by comparing (c) and (d) that the proposed method chooses a hyper parameter that can separate three clusters clearly, which suggests the effectiveness of the method. Note that kernel PCA does not use the explicit information of the three clusters, while they are displayed with different colors and markers for visualization purpose.

Table 2. The configuration of datasets

Dataset	# data	Dimension	# classes
Synthesized-1	175	2	(3)
Synthesized-2	150	2	(3)
Wine	178	13	3
Diabetes	145	3	3
BUPA	345	6	2
Fertility	100	9	2
Zoo	101	16	7
USPSG-500	500	256	5
Food	961	6	-

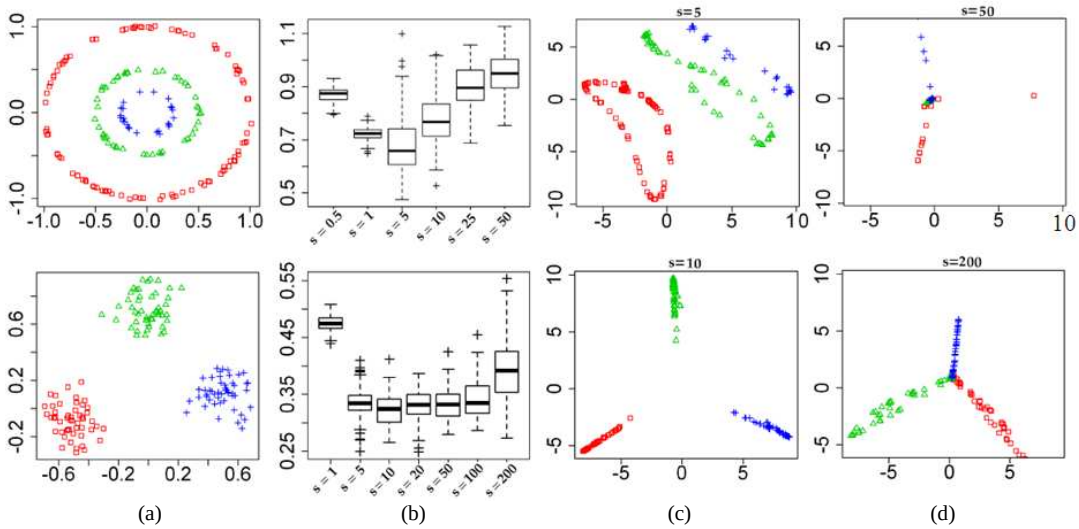

Fig. 4. Kernel PCA for Synthesized data-1 (top) and Synthesized data-2 (bottom), (a) scatter plot for the two variables of a sample. (b) boxplots of the LOOCV reconstruction errors for 100 samples, (c,d) scatter plot of the first two kernel principal components using (c) the best inverse kernel widths (s = 5, 10) and (d) larger bandwidths s = 50, 200

3.2. Computational Cost

To illustrate the computational cost of the proposed method, the CPU time (in second) for six different sizes of data (n) and five numbers of components (l) using synthesized data-2 are shown in **Table 3**. The CPU time increases as the sample size is larger, since the computation of LOOCV and the optimization of pre-images is heavier for larger samples. The configuration of the computer is Intel (R) Core (TM) i7 CPU 920@ 2.67 GHz., memory 12.00 GB and 64-bit operating system. We have used 'kernlab' package in R program for implementation of the kernel PCA. Gaussian RBF kernel is used inverse band width s = 50.

3.3. Real World Problems

We first apply the proposed method to five datasets: Wine, Diabetes, BUPA liver disorders, Fertility and Zoo, the former three of which are taken from Izenman (2008) and available at the website of the book and the latter two are taken from UCI Machine Learning Repository (Bache and Lichman, 2013).

As the kernel PCA is an unsupervised method, the evaluation of results is not straightforward. Since kernel PCA is often used as a pre-processing technique for regression and classification, we evaluate the LOOCV classification errors with the k-NN classifier (k = 5) to see the appropriateness of the hyper parameters chosen by the proposed method. Note that we do not use the class labels for kernel PCA, but use them only for evaluating the classification errors.

We consider a set of inverse bandwidths $s \in \{0.05, 0.10, 0.25, 0.50, 0.75, 1.00, 10.00\}$ and six numbers of kernel principal components $l \in \{2, 3, 4, 5, 8, 10\}$ for each dataset. The LOOCV reconstruction errors used in the proposed method and the LOOCV classification errors for all the hyperparameters are shown in **Table 4**, from which we see that the selected hyperparameters attain the minimum or close to the minimum classification error for all the datasets. This suggests that the proposed method provides appropriate hyperparameters that maintain the cluster structure effective for the classification tasks.

We next apply the proposed method for larger datasets in dimensionality and sample size. USPS data (Song *et al*., 2008) consists of 16×16 gray scale images of handwritten digits and thus the dimensionality is 256. The original dataset has 2007 images, but we draw 100 images from each of five digits 1, 2, 3, 4, 5 and add Gaussian noise with mean 0 and standard deviation 0.01. The dataset is referred to as USPSG-500. We take seven inverse bandwidths $s \in \{0.0001, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.025\}$ and eight numbers of kernel principal components $l \in \{2, 4, 8, 16, 32, 64, 128, 256\}$. The LOOCV reconstruction errors in the proposed method are shown in **Table 5**, in which the minimum is attained at s = 0.01 and l = 64. The kNN (k = 5) misclassification rates estimated with LOOCV are also listed in the table.

We next apply the proposed method to the nutritional value of food, which is not for classification. The dataset has 961 food items with six nutritional components as attributes (Izenman, 2008). We consider seven values of inverse bandwidths $s \in \{0.001, 0.1, 0.5, 0.75, 1, 5, 10, 100, 200\}$ and five numbers of components $l \in \{1, 2, 3, 4, 5, 6\}$. The results are displayed in **Table 6**. The smallest LOOCV reconstruction error is attained at s = 0.5 and l = 2. Since, unlike classification tasks, it is not straightforward to evaluate the performance of the proposed method, we show the scatter plots of the first two kernel principal components using three values of inverse bandwidths $s \in \{0.001, 0.5, 200\}$ in **Fig. 5**.

Table 3. Computational cost (in second) of the proposed method for synthesized data-2 with di-erent data sizes (n) and the numbers of components (l)

N/l	2	4	6	8	10
100	20.3	20.3	22.3	28.5	29.0
200	86.8	87.0	102	110	152
400	512	549	610	684	896
600	1.54×10^3	1.55×10^3	1.56×10^3	1.63×10^3	2.2310^3
800	$3.39\times10_3$	3.40×10^3	3.67×10^3	3.51×10^3	4.77×10^3
1000	6.06×10^3	6.51×10^3	6.50×10^3	6.52×10^3	1.07×10^4

Table 4. Five real-world data sets. LOOCV reconstruction errors and LOOCV classification errors for inverse bandwidths (s) and the number of components (l), the minimum values are written in bold fonts and the classification errors with the hyperparameters chosen by the proposed method are underlined

	Reconstruction errors						Classification errors					
S/l	2	3	4	5	8	10	2	3	4	5	8	10
Wine												
0.05	3.749	3.846	3.952	3.713	3.893	4.040	5.056	3.933	3.371	2.809	3.371	2.809
0.10	3.418	3.495	3.582	3.560	3.556	3.845	2.247	2.809	3.371	2.247	2.809	3.371
0.25	3.422	3.596	3.531	3.885	3.584	3.733	2.247	2.809	4.494	3.933	5.618	5.618
0.50	3.518	3.603	3.651	3.719	3.790	3.723	3.933	5.057	6.180	5.618	7.303	8.427
0.75	3.789	3.703	3.751	3.858	3.882	3.939	25.281	7.303	6.180	6.180	7.865	9.551
1.00	3.788	3.923	3.883	3.919	3.807	3.825	33.708	30.337	9.551	8.427	9.551	10.674
10.00	4.131	4.070	4.005	4.073	4.119	4.134	39.888	41.573	42.697	41.011	38.764	42.135
Diabetes												
0.05	2.343	2.398	2.183	7.591	16.027	20.605	20.690	21.310	21.310	20.000	20.000	20.000
0.10	1.761	1.872	1.795	1.713	4.879	6.913	23.448	21.310	21.310	24.828	24.138	25.517
0.25	1.598	1.467	1.660	1.636	2.066	2.751	20.690	20.000	21.310	20.690	21.379	20.000
0.50	1.505	1.318	1.492	1.476	1.597	1.712	22.069	20.690	20.000	21.379	21.379	21.379
0.75	1.555	1.494	1.560	1.519	1.575	1.716	21.379	20.000	22.069	22.069	22.069	21.379
1.00	1.626	1.617	1.609	1.530	1.647	1.512	20.690	23.448	24.138	20.690	20.000	20.690
10.00	2.362	2.167	2.152	2.098	2.292	2.269	37.241	37.241	40.000	34.483	36.552	37.241
BUPA												
0.05	2.964	2.751	2.758	2.190	5.462	5.300	42.319	43.479	40.580	42.029	42.029	42.029
0.10	2.439	2.232	2.325	2.042	4.266	4.838	48.116	46.667	41.739	47.826	48.116	47.826
0.25	2.064	2.042	2.012	2.123	2.175	2.269	50.145	48.696	42.029	50.145	50.145	50.145
0.50	2.138	2.148	2.077	2.238	2.166	2.071	50.145	46.957	44.638	49.855	49.855	49.855
0.75	2.253	2.196	2.147	2.364	2.138	2.241	53.333	42.609	49.855	53.333	53.333	53.623
1.00	2.128	2.177	2.154	2.282	2.256	2.123	50.145	43.189	47.826	50.145	50.145	50.145
10.00	2.464	2.447	2.467	2.427	2.392	2.481	44.058	44.928	44.928	44.058	44.058	44.058
Fertility												
0.05	3.955	4.132	4.100	3.911	3.876	3.811	13.000	14.000	16.000	13.000	13.000	13.000
0.10	3.570	3.568	3.560	8.067	3.490	3.428	15.000	11.000	12.000	15.000	15.000	15.000
0.25	3.325	3.330	3.349	3.279	3.442	3.407	11.000	14.000	15.000	11.000	11.000	11.000
0.50	3.601	3.592	3.630	3.713	3.764	3.559	13.000	11.000	11.000	13.000	13.000	13.000
0.75	3.896	3.848	3.911	4.031	3.624	3.673	10.000	10.000	12.000	10.000	10.000	10.000
1.00	3.989	3.936	3.892	3.919	3.774	3.819	12.000	15.000	13.000	12.000	12.000	12.000
10.00	3.678	3.663	3.568	3.714	3.489	3.500	12.000	15.000	13.000	12.000	12.000	12.000
Zoo												
0.05	4.581	4.644	5.460	6.051	7.434	5.957	12.871	12.871	13.861	12.871	11.881	11.881
0.10	3.861	3.858	3.816	3.820	4.886	5.369	14.851	11.881	12.871	15.842	16.832	14.851
0.25	3.607	3.615	3.632	3.748	3.863	3.871	19.802	15.842	10.891	17.822	19.802	19.802
0.50	3.572	4.078	3.460	3.637	3.935	3.667	22.772	12.871	11.881	22.772	22.772	21.782
0.75	3.523	3.591	3.801	3.750	3.893	4.140	27.723	27.723	26.733	26.733	24.752	24.752
1.00	3.738	3.853	3.866	3.999	3.896	4.013	24.752	25.743	30.693	22.772	24.752	25.743
10.00	4.013	4.049	3.992	4.024	4.037	4.006	56.436	53.465	48.514	55.446	55.446	56.436

3.4. Polynomial Kernel

We use the proposed method for choosing the hyperparameters in the polynomial kernel. Using Wine dataset, we consider seven values of offset parameters $c \in \{0.1, 0.5, 1, 5, 10, 25, 50\}$, two values of degree $d \in \{2, 3\}$ and four numbers of kernel principal components $l \in \{2, 3, 4, 5\}$. The results are given in **Table 7**. We observe that the smallest LOOCV reconstruction error is attained in the area close to the minimum classification error.

Table 5. USPSG-500. LOOCV reconstruction errors and LOOCV classification errors (bold numbers indicate the minimum value)

s/l	2	4	8	16	32	64	128	256
Reconstruction errors in the proposed method								
0.0001	1139.810	1203.316	1159.020	752.494	134.936	130.678	143.080	534.779
0.0010	129.168	129.627	124.333	110.106	143.470	82.734	69.729	203.068
0.0025	42.422	40.708	44.493	38.588	51.707	26.516	92.497	107.448
0.0050	18.967	21.120	22.642	20.010	18.957	20.592	26.828	33.991
0.0075	18.989	15.903	16.963	14.804	14.369	13.909	14.879	17.523
0.0100	16.648	15.081	14.161	12.785	12.485	12.444	15.787	14.270
0.0250	13.339	13.498	13.149	13.085	13.915	14.173	14.086	14.595
Classification errors (%)								
0.0001	32.000	11.200	4.600	2.000	3.000	3.000	3.400	4.000
0.0010	31.000	12.200	4.400	2.200	2.800	3.000	3.000	4.200
0.0025	31.400	11.600	4.400	2.600	2.200	3.000	3.200	3.400
0.0050	31.600	11.000	4.600	3.00	1.800	2.400	3.600	4.400
0.0075	28.200	11.400	4.800	3.400	1.800	2.800	3.200	5.200
0.0100	31.000	15.200	4.400	3.800	3.000	2.200	2.600	5.000
0.0250	45.800	25.400	7.600	5.200	5.600	6.800	5.200	15.600

Table 6. LOOCV reconstruction errors for food data

S/l	1	2	3	4	5	6
0.001	20.226	18.741	18.334	18.361	18.462	13.901
0.1	2.215	2.024	1.977	1.840	1.849	1.956
0.5	1.923	1.738	2.143	2.097	2.034	1.922
0.75	1.817	1.908	1.883	1.891	1.850	1.930
1	1.854	1.844	1.813	1.798	2.050	1.927
5	2.306	2.214	2.128	2.229	2.203	2.238
10	2.380	2.286	2.200	2.239	2.808	2.259
100	1.987	1.982	1.943	2.014	2.088	2.234
200	2.070	2.066	2.097	2.123	2.234	2.192

Table 7. Polynomial kernel for Wine data: LOOCV reconstruction errors and the LOOCV classification errors (bold numbers indicate the minimum value)

	l = 2		l = 3		l = 4		l = 5	
c/d	2	3	2	3	2	3	2	3
Reconstruction errors in the proposed method								
0.1	4.165	3.807	4.059	3.818	4.108	3.821	4.153	3.805
0.5	4.051	3.781	3.978	3.758	4.003	3.768	3.952	3.805
1.0	3.976	3.837	3.888	3.869	3.966	3.709	4.023	3.819
5.0	3.752	3.859	3.759	3.813	4.108	3.803	4.153	3.739
10.0	3.784	3.780	3.740	3.810	4.003	3.762	3.952	3.792
25.0	3.755	3.820	3.709	3.730	3.966	3.768	4.023	3.775
50.0	3.761	3.782	3.735	3.724	3.750	3.777	3.736	3.743
Classification errors (%)								
0.1	18.539	17.978	16.292	3.933	15.730	5.056	15.730	4.494
0.5	14.045	17.978	11.798	3.933	11.798	3.933	14.045	4.494
1.0	15.730	16.292	12.360	3.371	11.798	3.933	8.989	3.933
5.0	2.247	3.371	3.933	3.371	3.933	3.371	1.685	3.933
10.0	2.809	1.685	3.371	2.809	1.685	3.371	2.247	2.809
25.0	3.933	3.371	2.809	2.247	4.494	2.247	2.247	2.247
50.0	4.494	3.933	2.809	2.809	4.494	2.247	2.247	2.247

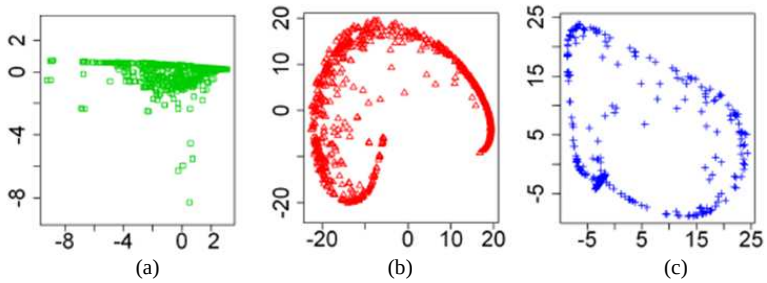

Fig. 5. Visualization of the first two kernel principal components of food data (a) s = 0.001, (b) s = 0.5 and (c) s = 200

4. DISCUSSION

While kernel PCA has been applied in various areas of machine learning, such as dimensionality reduction, feature extraction, de-noising and so on (Scholkopf and Smola, 2002; Rathi *et al*., 2006; Hofmann, 2007; Zheng *et al*., 2010; Feng and Liu, 2013), in most cases the kernel and the number of features are chosen in a heuristic way. Recently, multi-kernel PCA (Ren *et al*., 2013) has been also proposed, which applies combination of multiple kernels instead of choosing one. It is well known, however, that the multi-kernel approach results in a computationally heavy algorithm, which may need advanced optimization technique. The method proposed in this study, in contrast, is based on the reconstruction errors in the original space, which can be regarded as a natural extension of the aim of the standard linear PCA. The required computation is simply cross-validation with a basic optimization algorithm such as the fixed-point or gradient method.

We provide detailed discussions on the experimental results for real-world data sets in section 3. For classification data sets, we can see from **Table 4 and 5** that the hyperparameter (bandwidth parameter in Gaussian RBF kernel and the number of principal components) gives the best or close to best LOOCV classification error: The best for Wine data and the second or third best for the other 5 data sets. In all cases, we observe that the chosen hyperparameters are close to the best parameters for the classification error. These experimental observations imply that the proposed method gives appropriate hyperparameters, with which the low dimensional features obtained by kernel PCA represent effective information of data.

From **Table 6 and Fig. 5**, we can see that the hyperparameter chosen by the proposed method provides the features with clearer structure than the other two hyperparameters used in (a) and (c). For this data set, Izenman (2008) provides detailed analysis on the results of kernel PCA with a hand-tuned bandwidth parameter: A meaningful "curve" structure is observed in the result of two-dimensional kernel PCA. As shown in **Fig. 5**, our method automatically chooses such a hyperparameter that accords with the observation in Izenman (2008).

We can also observe from **Table 7** that the proposed method chooses the hyperparameters for kernel PCA with polynomial kernel so that the corresponding LOOCV for classification error attains the third best. This accords with the observation on the other cases with Gaussian RBF kernel and demonstrates the appropriateness of the proposed method.

Regarding the computational cost of the proposed method, the proposed method needs to solve the preimage problem for each of the data, which may cause a computational issue for large data set. **Table 3** shows that the computational time increases roughly quadratically with respect to the sample size. To reduce the computational cost, it may be possible to use only a part of data for evaluating reconstruction errors in choosing hyperparameters.

5. CONCLUSION

We have discussed the kernel PCA and proposed a method for choosing hyperparameters, optimal kernel (parameters in a kernel) and the number of kernel principal components, through the LOOCV for the reconstruction errors of pre-images. We have made empirical studies using synthesized examples and real-world datasets. For evaluation of the proposed method, in addition to visualization, we used classification errors for the projected data onto the subspace chosen by the method, if the data set is provided for a classification task. We have observed that for all the datasets classification performance of the kernel PCA chosen by the proposed method is the best or close to the best

among the candidates of hyperparameters. The experimental results imply that the proposed method successfully provides an automatic way of finding such hyperparameters that give appropriate low-dimensional representation of data by kernel PCA.

There are also limitations of the proposed method. First, the optimization such as fixed-point and steepest descent method for computing the pre-image has possibility of being trapped by local optimum. Applying other pre-image methods to alleviate this problem will be an important future research. Second, since our method uses the cross-validation with pre-image optimization, it may be time-consuming for large datasets. One possible approach is to use a part of data for evaluating reconstruction errors and it is also an interesting future direction to develop a more efficient way of hyperparameter choice for kernel PCA. Third, the reconstruction errors in the proposed method assume that the original space admits a metric, while kernel PCA can be applied to more general data spaces including non-metric spaces. It is also among our future studies to consider hyperparameter choice applicable to kernel PCA for non-metric spaces.

6. ACKNOWLEDGEMENT

This study has been supported by JSPS Grant-in-Aid for Scientific Research 22300098 and MEXT Grant-in-Aid for Scientific Research on Innovative Areas 25120012.

7. REFERENCES

Akaho, S., 2001. A kernel method for canonical correlation analysis. Int. Meet. Psychometr. Society, 35: 321-377.

Alam, M.A., M. Nasser and K. Fukumizu, 2010. A Comparative study of kernel and robust canonical correlation analysis. J. Multimidia, 5: 3-11. DOI: 10.4304/jmm.5.1.3-11

Arias, P., G. Arias and G. Sapiro, 2007. Connecting the out of sample and pre-image problems in kernel methods. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun. 17-22, IEEE Xplore Press, Minneapolis, MN, pp: 1-8. DOI: 10.1109/CVPR.2007.383038

Aronszajn, N., 1950. Theory of reproducing kernels. Trans. Am. Math. Society, 68: 337-404. DOI: 10.2307/1990404

Bach, F.R. and M. I. Jordan, 2002. Kernel independent component analysis. J. Machine Learn. Res., 3: 1-48.

Bache, K. and M. Lichman, 2013. UCI machine learning repository.

Bakir, G., J. Weston and B. Scholkopf, 2004. Learning to Find Pre-Images. In: Advances in Neural Information Processing Systems, Thrun, S., L. Saul and B. Scholkopf (Eds.), MIT Press, Cambridge, ISBN-10: 0262201526, pp: 449-456.

Boser, B.E., I.M. Guyon and V.N. Vapnik, 1992. A training algorithm for optimal margin classifiers. Proceedings of the 5th Annual Workshop on Computational Learning Theory, Jul. 27-29, Pittsburgh, PA, USA., pp: 144-152. DOI: 10.1145/130385.130401

Feng, Y. and Y. Liu, 2013. A cellular automata model based on nonlinear kernel principal component analysis for urban growth simulation. Environ. Plann. B, 40: 116-134. DOI: 10.1068/b37142

Hofmann, H., 2007. Kernel PCA for novelty detection. Patt. Recogn., 40: 863-874. DOI: 10.1016/j.patcog.2006.07.009

Honeine, P. and C. Richard, 2011. A closed-form solution for the pre-image problem in kernel-based machines. J. Signal Process. Syst., 63: 289-299. DOI: 10.1007/s11265-010-0482-9

Izenman, A.J., 2008. Modern Multivariate Statistical Techniques: Regression, Classification and Manifold Learning. 1st Edn., Springer, New York, ISBN-10: 0387781889, pp: 731.

Krzanowski, W.J., 1987. Cross-validation in principal component analysis. Biometrics, 43: 575-584.

Kwok, J.T.Y. and I.W. Tsang, 2003. The pre-image problem in kernel methods. Proceedings of the 20th International Conference on Machine Learning, (CML' 03), Washington DC, pp: 408-415.

Mika, S., B. Scholkopf, A. Smola, K.B. Muller and G. Ratsch, 1999. Kernel PCA and de-noising in feature spaces. Proceedings of the Conference on Advances in Neural Information Processing Systems II, (IPS' 99), MIT Press Cambridge, MA, USA., pp: 536-542.

Pearson, K., 1901. On lines and planes of closest fit to systems of points in space. Philosophical Magaz., 2: 559- 572. DOI: 10.1080/14786440109462720

Premanode, B., J. Vongprasert, N. Sopipan and C. Toumazou, 2013. A novel multiclass support vector machine algorithm using mean reversion and coefficient of variance. J. Math. Stat., 9: 208-218. DOI: 10.3844/jmssp.2013.208.218

Rathi, Y., S. Dambreville and A. Tannenbaum, 2006. Statistical shape analysis using kernel PCA. Proc. SPIE, 6064: 425-432. DOI: 10.1117/12.641417

Ren, S., P. Ling, M. Yang, Y. Ni and Z. Song, 2013. Multi-kernel pca with discriminant manifold for hoist monitoring. J. Applied Sci., 13: 4195-4200. DOI: 10.3923/jas.2013.4195.4200

Saunders, C., A. Gammerman and V. Vovk, 1998. Ridge regression learning algorithm in dual variables. Proceedings of the 15th International Conference on Machine Learning, (CML' 98), Morgan Kaufmann, San Francisco, CA., pp: 515-521.

Scholkopf, B. and A.J. Smola, 2002. Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. 1st Edn., MIT Press, Cambridge MA., ISBN-10: 0262194759, pp: 626.

Scholkopf, B., A. J. Smola and K.R. Muller, 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput., 10: 1299-1319. DOI: 10.1162/089976698300017467

Song, L., A. Smola, K. Borgwardt and A. Gretton, 2008. Colored maximum variance unfolding. Adv. Neural Inform. Process. Syst., 20: 1385-1392.

Stone, M., 1974. Cross-validatory choice and assessment of statistical predictions (with discussion). J. Royal Stat. Society, Series B, 36: 111-147.

Wold, S., 1978. Cross-validation estimation of the number of components in factor and principal components models. Technometrics, 20: 397-405. DOI: 10.1080/00401706.1978.10489693

Zheng, W.S., J. H. Lai and P.C. Yuen, 2010. Penalized preimage learning in kernel principal component analysis. Neural Netw., 21: 551-570. DOI: 10.1109/TNN.2009.2039647