We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us? Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

4

Subset Basis Approximation of Kernel Principal Component Analysis

Yoshikazu Washizawa
The University of Electro-Communications
Japan

1. Introduction

Principal component analysis (PCA) has been extended to various ways because of its simple definition. Especially, non-linear generalizations of PCA have been proposed and used in various areas. Non-linear generalizations of PCA, such as principal curves (Hastie & Stuetzle, 1989) and manifolds (Gorban et al., 2008), have intuitive explanations and formulations comparing to the other non-linear dimensional techniques such as ISOMAP (Tenenbaum et al., 2000) and Locally-linear embedding (LLE) (Roweis & Saul, 2000).

Kernel PCA (KPCA) is one of the non-linear generalizations of PCA by using the kernel trick (Schölkopf et al., 1998). The kernel trick nonlinearly maps input samples to higher dimensional space so-called the feature space $\mathcal{F}$. The mapping is denoted by Φ, and let $\boldsymbol{x}$ be a d-dimensional input vector,

$$\Phi : \mathbb{R}^d \to \mathcal{F}, \quad \boldsymbol{x} \mapsto \Phi(\boldsymbol{x}). \tag{1}$$

Then a linear operation in the feature space is a non-linear operation in the input space. The dimension of the feature space $\mathcal{F}$ is usually much larger than the input dimension d, or could be infinite. The positive definite kernel function $k(\cdot,\cdot)$ that satisfies following equation is used to avoid calculation in the feature space,

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle \Phi(\boldsymbol{x}_1), \Phi(\boldsymbol{x}_2) \rangle \quad \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product.

By using the kernel function, inner products in $\mathcal{F}$ are replaced by the kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. According to this replacement, the problem in $\mathcal{F}$ is reduced to the problem in $\mathbb{R}^n$, where n is the number of samples since the space spanned by mapped samples is at most n-dimensional subsapce. For example, the primal problem of Support vector machines (SVMs) in $\mathcal{F}$ is reduced to the Wolf dual problem in $\mathbb{R}^n$ (Vapnik, 1998).

In real problems, the number of n is sometimes too large to solve the problem in $\mathbb{R}^n$. In the case of SVMs, the optimization problem is reduced to the convex quadratic programming whose size is n. Even if n is too large, SVMs have efficient computational techniques such as chunking or the sequential minimal optimization (SMO) (Platt, 1999), since SVMs have sparse solutions for the Wolf dual problem. After the optimal solution is obtained, we only have to store limited number of learning samples so-called support vectors to evaluate input vectors.

In the case of KPCA, the optimization problem is reduced to an eigenvalue problem whose size is n. There are some efficient techniques for eigenvalue problems, such as the divide-and-conquer eigenvalue algorithm (Demmel, 1997) or the implicitly restarted Arnoldi method (IRAM) (Lehoucq et al., 1998) [1]. However, their computational complexity is still too large to solve when n is large, because KPCA does not have sparse solution. These algorithms require $O(n^2)$ working memory space and $O(rn^2)$ computational complexity, where r is the number of principal components. Moreover, we have to store all n learning samples to evaluate input vectors.

Subset KPCA (SubKPCA) approximates KPCA using the subset of samples for its basis, and all learning samples for the criterion of the cost function (Washizawa, 2009). Then the optimization problem for SubKPCA is reduced to the generalized eigenvalue problem whose size is the size of the subset, m. The size of the subset m defines the trade-off between the approximation accuracy and the computational complexity. Since all learning samples are utilized for its criterion, even if m is much smaller than n, the approximation error is small. The approximation error due to this subset approximation is discussed in this chapter. Moreover, after the construction, we only have to store the subset to evaluate input vectors.

An illustrative example is shown in Figure 1. Figure 1 (a) shows artificial 1000 2-dimensional samples, and contour lines of norms of transformed vectors onto one-dimensional subspace by PCA. Figure 1 (b) shows contour curves by KPCA (transformed to five-dimensional subspace in $\mathcal{F}$). This is non-linear analysis, however, it requires to solve an eigenvalue problem whose size is 1000. For an input vector, calculations of kernel function with all 1000 samples are required. Figure 1 (c) randomly selects 50 samples, and obtains KPCA. In this case, the size of the eigenvalue problem is only 50, and calculations of kernel function with only 50 samples are required to obtain the transform. However, the contour curves are rather different from (b). Figure 1 (d) shows contour curves of SubKPCA by using the 50 samples for its basis, and all 1000 samples for evaluation. The contour corves are almost that same with (b). In this case, the size of the eigenvalue problem is also only 50, and the number of calculations of kernel function is also 50.

There are some conventional approaches to reduce the computational complexity of KPCA. improved KPCA (IKPCA) (Xu et al., 2007) is similar approach to SubKPCA, however, the approximation error is much higher than SubKPCA. Experimental and theoretical difference are shown in this chapter. Comparisons with Sparse KPCAs (Smola et al., 1999; Tipping, 2001), Nyström method (Williams & Seeger, 2001), incomplete Cholesky decomposition (ICD) (Bach & Jordan, 2002) and adaptive approaches (Ding et al., 2010; Günter et al., 2007; Kim et al., 2005) are also diecussed.

In this chapter, we denote vectors by bold-italic lower symbols $\boldsymbol{x}, \boldsymbol{y}$, and matrices by bold-italic capital symbols $\boldsymbol{A}, \boldsymbol{B}$. In kernel methods, $\mathcal{F}$ could be infinite-dimensional space up to the selection of the kernel function. If vectors could be infinite (functions), we denote them by italic lower symbols f, g. If either domain or range of linear transforms could be infinite-dimensional space, we denote the transforms by italic capital symbols X, Y. This is summarized as follows; (i) bold symbols, $\boldsymbol{x}, \boldsymbol{A}$, are always finite. (ii) non-bold symbols, f, X, could be infinite.

[1] IRAM is implemented as "eigs" in MATLAB

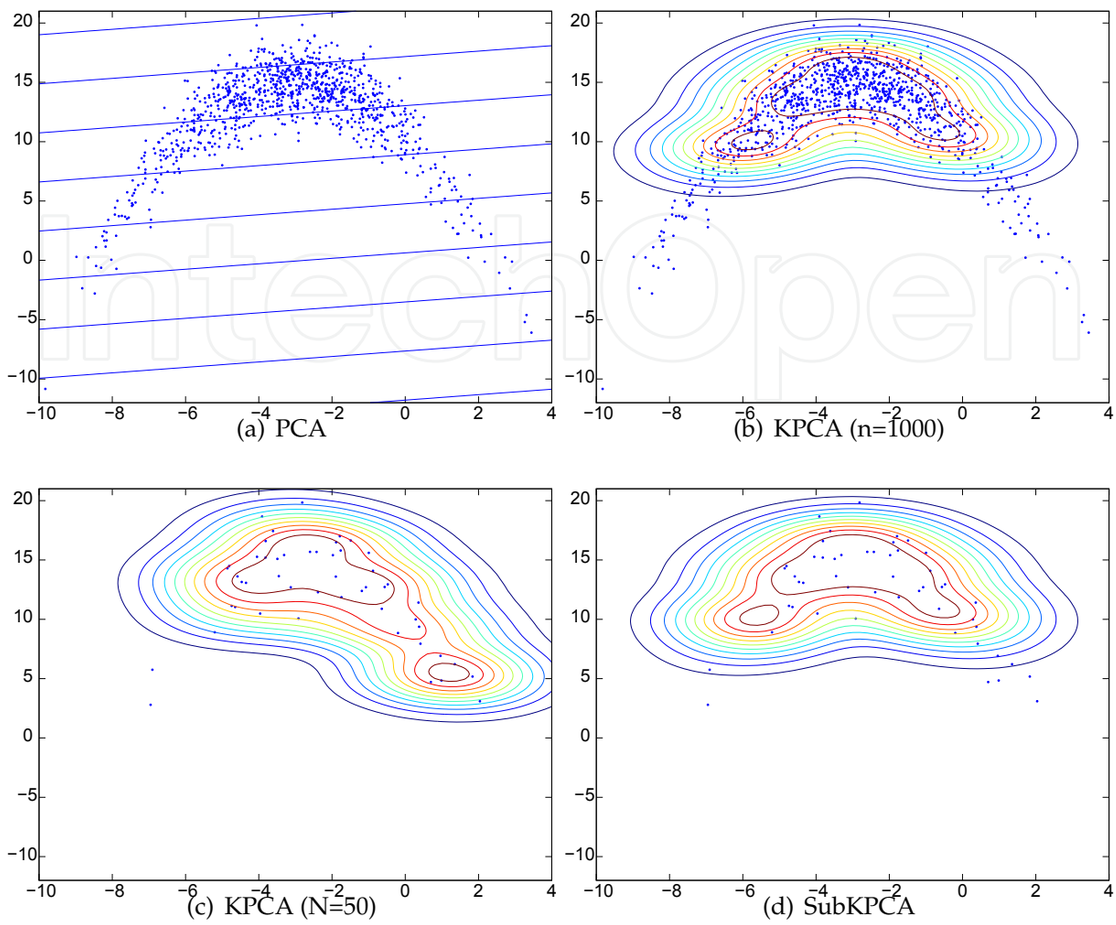


Fig. 1. Illustrative example of SubKPCA

2. Kernel PCA

This section briefly reviews KPCA, and shows some characterizations of KPCA.

2.1 Brief review of KPCA

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be d-dimensional learning samples, and $\boldsymbol{X} = [\boldsymbol{x}_1 | \ldots | \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$. Suppose that their mean is zero or subtracted. Standard PCA obtains eigenvectors of the variance-covariance matrix $\boldsymbol{\Sigma}$,

$$\boldsymbol{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top = \frac{1}{n} \boldsymbol{X} \boldsymbol{X}^\top. \tag{3}$$

Then the ith largest eigenvector corresponds to the ith principal component. Suppose $\boldsymbol{U}_{\mathrm{PCA}} = [\boldsymbol{u}_1 | \ldots | \boldsymbol{u}_r]$. The projection and the transform of $\boldsymbol{x}$ onto r-dimensional eigenspace are $\boldsymbol{U}_{\mathrm{PCA}} \boldsymbol{U}_{\mathrm{PCA}}^\top \boldsymbol{x}$ and $\boldsymbol{U}_{\mathrm{PCA}}^\top \boldsymbol{x}$ respectively.

In the case of KPCA, input vectors are mapped to feature space before the operation. Let

$$S = [\Phi(\boldsymbol{x}_1)|\ldots|\Phi(\boldsymbol{x}_n)] \tag{4}$$

$$\Sigma_{\mathcal{F}} = SS^* \tag{5}$$

$$\boldsymbol{K} = S^*S \in \mathbb{R}^{n\times n}, \tag{6}$$

where $\cdot^*$ denotes the adjoint operator [2], and $\boldsymbol{K}$ is called the kernel Gram matrix (Schölkopf et al., 1999), and i,j-component of $\boldsymbol{K}$ is $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Then the ith largest eigenvector corresponds to the ith principal component. If the dimension of $\mathcal{F}$ is large, eigenvalue decomposition (EVD) cannot be performed. Let $\{\lambda_i, u_i\}$ be the ith eigenvalue and corresponding eigenvector of $\Sigma_{\mathcal{F}}$ respectively, and $\{\lambda_i, \boldsymbol{v}_i\}$ be the ith eigenvalue and eigenvector of $\boldsymbol{K}$. Note that $\boldsymbol{K}$ and $\Sigma_{\mathcal{F}}$ have the same eigenvalues. Then the ith principal component can be obtained from the ith eigenvalue and eigenvector of $\boldsymbol{K}$,

$$u_i = \frac{1}{\sqrt{\lambda_i}} S\boldsymbol{v}_i. \tag{7}$$

Note that it is difficult to obtain u_i explicitly on a computer because the dimension of $\mathcal{F}$ is large. However, the inner product of a mapped input vector $\Phi(\boldsymbol{x})$ and the ith principal component is easily obtained from,

$$\langle u_i, \Phi(\boldsymbol{x})\rangle = \frac{1}{\sqrt{\lambda_i}} \langle \boldsymbol{v}_i, \boldsymbol{k}_{\boldsymbol{x}}\rangle, \tag{8}$$

$$\boldsymbol{k}_{\boldsymbol{x}} = [k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n)]^\top \tag{9}$$

$\boldsymbol{k}_{\boldsymbol{x}}$ is an n-dimensional vector called the empirical kernel map.

Let us summarize using matrix notations. Let

$$\boldsymbol{\Lambda}_{\mathrm{KPCA}} = \mathrm{diag}([\lambda_1, \ldots, \lambda_r]) \tag{10}$$

$$U_{\mathrm{KPCA}} = [u_1|\ldots|u_r] \tag{11}$$

$$\boldsymbol{V}_{\mathrm{KPCA}} = [\boldsymbol{v}_1|\ldots|\boldsymbol{v}_r]. \tag{12}$$

Then the projection and the transform of $\boldsymbol{x}$ onto the r-dimensional eigenspace are

$$U_{\mathrm{KPCA}} U_{\mathrm{KPCA}}^* \Phi(\boldsymbol{x}) = S\boldsymbol{V}_{\mathrm{KPCA}} \boldsymbol{\Lambda}^{-1} \boldsymbol{V}_{\mathrm{KPCA}}^\top \boldsymbol{k}_{\boldsymbol{x}}, \tag{13}$$

$$U_{\mathrm{KPCA}}^* \Phi(\boldsymbol{x}) = \boldsymbol{\Lambda}^{-1/2} \boldsymbol{V}_{\mathrm{KPCA}}^\top \boldsymbol{k}_{\boldsymbol{x}}. \tag{14}$$

2.2 Characterization of KPCA

There are some characterizations or definitions for PCA (Oja, 1983). SubKPCA is extended from the least mean square (LMS) error criterion [3].

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & J_0(\boldsymbol{X}) = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{X}\boldsymbol{x}_i\|^2 \\ \text{Subject to } & \mathrm{rank}(\boldsymbol{X}) \le r. \end{aligned} \tag{15}$$

[2] In real finite dimensional space, the adjoint and the transpose $\cdot^\top$ are equivalent. However, in infinite dimensional space, the transpose is not defined

[3] Since all definitions of PCA lead to the equivalent solution, SubKPCA is also defined by the other definitions. However, in this chapter, only LMS criteria is shown.

From this definition, $\boldsymbol{X}$ that minimizes the averaged distance between $\boldsymbol{x}_i$ and $\boldsymbol{X}\boldsymbol{x}_i$ over i is obtained under the rank constraint. Note that from this criterion, each principal component is not characterized, i.e., the minimum solution is $\boldsymbol{X} = \boldsymbol{U}_{\mathrm{PCA}}\boldsymbol{U}_{\mathrm{PCA}}^\top$, and the transform $\boldsymbol{U}_{\mathrm{PCA}}$ is not determined.

In the case of KPCA, the criterion is

$$\begin{aligned} \min_X \quad & J_1(X) = \frac{1}{n}\sum_{i=1}^{n} \|\Phi(\boldsymbol{x}_i) - X\Phi(\boldsymbol{x}_i)\|^2 \\ \text{Subject to } & \mathrm{rank}(X) \le r, \ \ \mathcal{N}(X) \supset \mathcal{R}(S)^\perp, \end{aligned} \tag{16}$$

where $\mathcal{R}(A)$ denotes the range or the image of the matrix or the operator A, and $\mathcal{N}(A)$ denotes the null space or the kernel of the matrix or the operator A. In linear case, we can assume that the number of samples n is sufficiently larger than r and d, and the second constraint $\mathcal{N}(X) \supset \mathcal{R}(S)^\perp$ is often ignored. However, since the dimension of the feature space is large, r could be larger than the dimension of the space spanned by mapped samples $\Phi(\boldsymbol{x}_1), \ldots, \Phi(\boldsymbol{x}_n)$. For such cases, the second constraint is introduced.

2.2.1 Solution to the problem (16)

Here, brief derivation of the solution to the problem (16) is shown. Since the problem is in $\mathcal{R}(S)$, X can be parameterized by $X = S\boldsymbol{A}S^*$, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Accordingly, J_1 yields

$$\begin{aligned} J_1(A) &= \frac{1}{n}\|S - SAS^*S\|_F^2 = \frac{1}{n}\mathrm{Trace}[\boldsymbol{K} - \boldsymbol{K}\boldsymbol{A}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{A}^\top\boldsymbol{K} + \boldsymbol{A}^\top\boldsymbol{K}\boldsymbol{A}\boldsymbol{K}] \\ &= \frac{1}{n}\|\boldsymbol{K}\boldsymbol{A}\boldsymbol{K}^{1/2} - \boldsymbol{K}^{1/2}\|_F^2 \end{aligned} \tag{17}$$

where $\cdot^{1/2}$ denotes the square root matrix, and $\|\cdot\|_F$ denotes the Frobenius norm. The eigenvalue decomposition of $\boldsymbol{K}$ is $\boldsymbol{K} = \sum_{i=1}^{n}\lambda_i\boldsymbol{v}_i\boldsymbol{v}_i^\top$. From the Schmidt approximation theorem (also called Eckart-Young theorem) (Israel & Greville, 1973), J_1 is minimized when

$$\boldsymbol{K}\boldsymbol{A}\boldsymbol{K}^{1/2} = \sum_{i=1}^{r}\sqrt{\lambda_i}\boldsymbol{v}_i\boldsymbol{v}_i^\top \tag{18}$$

$$\boldsymbol{A} = \sum_{i=1}^{r}\frac{1}{\lambda_i}\boldsymbol{v}_i\boldsymbol{v}_i^\top = \boldsymbol{V}_{\mathrm{KPCA}}\boldsymbol{\Lambda}^{-1}\boldsymbol{V}_{\mathrm{KPCA}}^\top \tag{19}$$

2.3 Computational complexity of KPCA

The procedure of KPCA is as follows;

1. Calculate $\boldsymbol{K}$ from samples. $[O(n^2)]$
2. Perform EVD for $\boldsymbol{K}$, and obtain the r largest eigenvalues and eigenvectors, $\lambda_1, \ldots, \lambda_r$, $\boldsymbol{v}_1, \ldots \boldsymbol{v}_r$. $[O(rn^2)]$
3. Obtain $\boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}_{\mathrm{KPCA}}^\top$, and store all training samples.
4. For an input vector $\boldsymbol{x}$, calculate the empirical kernel map $\boldsymbol{k}_{\boldsymbol{x}}$ from Eq. (9). $[O(n)]$
5. Obtain transformed vector Eq. (14). $[O(rn)]$

The procedures 1, 2, and 3 are called the learning (training) stage, and the procedures 4 and 5 are called the evaluation stage.

The dominant computation for the learning stage is EVD. In realistic situation, n should be less than several tens of thousands. For example, if $n = 100,000$, 20Gbyte RAM is required to store $\boldsymbol{K}$ on four byte floating point system. This computational complexity is sometimes too heavy to use for real large-scale problems. Moreover, in the evaluation stage, response time of the system depends on the number of n.

3. Subset KPCA

3.1 Definition

Since the problem of KPCA in the feature space $\mathcal{F}$ is in the subspace spanned by the mapped samples, $\Phi(\boldsymbol{x}_1), \ldots, \Phi(\boldsymbol{x}_n)$, i.e., $\mathcal{R}(S)$, the problem in $\mathcal{F}$ is transformed to the problem in $\mathbb{R}^n$. SubKPCA seeks the optimal solution in the space spanned by smaller number of samples, $\Phi(\boldsymbol{y}_1), \ldots, \Phi(\boldsymbol{y}_m)$, $m \leq n$ that is called a basis set. Let $T = [\Phi(\boldsymbol{y}_1), \ldots, \Phi(\boldsymbol{y}_m)]$, then the optimization problem of SubKPCA is defined as

$$\begin{aligned} &\min_{X} \quad J_1(X) \\ &\text{Subject to } \operatorname{rank}(X) \leq r, \;\; \mathcal{N}(X) \supset \mathcal{R}(T)^{\perp}, \;\; \mathcal{R}(X) \subset \mathcal{R}(T). \end{aligned} \tag{20}$$

The third and the fourth constraints indicate that the solution is in $\mathcal{R}(T)$. It is worth noting that SubKPCA seeks the solution in the limited space, however, the objective function is the same as that of KPCA, i.e., all training samples are used for the criterion. We call the set of all training samples the criterion set. The selection of the basis set $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$ is also important problem, however, here we assume that it is given, and the selection is discussed in the next section.

3.2 Solution of SubKPCA

At first, the minimal solutions to the problem (20) are shown, then their derivations are shown. If $\mathcal{R}(T) \subset \mathcal{R}(S)$, its solution is simplified. Note that if the set $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$ the subset of $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, $\mathcal{R}(T) \subset \mathcal{R}(S)$ is satisfied. Therefore, solutions for two cases are shown, ($\mathcal{R}(T) \subset \mathcal{R}(S)$ and all cases)

3.2.1 The case $\mathcal{R}(T) \subset \mathcal{R}(S)$

Let $\boldsymbol{K_y} = T^*T \in \mathbb{R}^{m \times m}$, $(\boldsymbol{K_y})_{i,j} = k(\boldsymbol{y}_i, \boldsymbol{y}_j)$, $\boldsymbol{K_{xy}} = X^*T \in \mathbb{R}^{n \times m}$, $(\boldsymbol{K_{xy}})_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{y}_j)$. Let $\kappa_1, \ldots, \kappa_r$ and $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_r$ be sorted eigenvalues and corresponding eigenvectors of the generalized eigenvalue problem,

$$\boldsymbol{K}_{\boldsymbol{xy}}^{\top} \boldsymbol{K_{xy}} \boldsymbol{z} = \kappa \boldsymbol{K_y} \boldsymbol{z} \tag{21}$$

respectively, where each eigenvector $\boldsymbol{z}_i$ is normalized by $\boldsymbol{z}_i \leftarrow \boldsymbol{z}_i / \sqrt{\langle \boldsymbol{z}_i, \boldsymbol{K_y} \boldsymbol{z}_i \rangle}$, that is $\langle \boldsymbol{z}_i, \boldsymbol{K_y} \boldsymbol{z}_j \rangle = \delta_{ij}$ (Kronecker delta). Let $\boldsymbol{Z} = [\boldsymbol{z}_1 | \ldots | \boldsymbol{z}_r]$, then the problem (20) is minimized by

$$P_{\text{SubKPCA}} = T \boldsymbol{Z} \boldsymbol{Z}^{\top} T^*. \tag{22}$$

The projection and the transform of SubKPCA for an input vector $\boldsymbol{x}$ are

$$P_{\text{SubKPCA}}\Phi(\boldsymbol{x}) = T\boldsymbol{Z}\boldsymbol{Z}^\top \boldsymbol{h}_{\boldsymbol{x}} \tag{23}$$

$$U_{\text{SubKPCA}}\Phi(\boldsymbol{x}) = \boldsymbol{Z}^\top \boldsymbol{h}_{\boldsymbol{x}}, \tag{24}$$

where $\boldsymbol{h}_{\boldsymbol{x}} = [k(\boldsymbol{x}, \boldsymbol{y}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{y}_m)] \in \mathbb{R}^m$ is the empirical kernel map of $\boldsymbol{x}$ for the subset.

A matrix or an operator A that satisfies $AA = A$ and $A^\top = A$ ($A^* = A$), is called a projector (Harville, 1997). If $\mathcal{R}(T) \subset \mathcal{R}(S)$, P_{SubKPCA} is a projector since $P^*_{\text{SubKPCA}} = P_{\text{SubKPCA}}$, and

$$P_{\text{SubKPCA}}P_{\text{SubKPCA}} = T\boldsymbol{Z}\boldsymbol{Z}^\top \boldsymbol{K}_{\boldsymbol{y}}\boldsymbol{Z}\boldsymbol{Z}^\top T^* = T\boldsymbol{Z}\boldsymbol{Z}^\top T^* = P_{\text{SubKPCA}}. \tag{25}$$

3.2.2 All cases

The Moore-Penrose pseudo inverse is denoted by $\cdot^\dagger$. Suppose that EVD of $(\boldsymbol{K}_{\boldsymbol{y}})^\dagger \boldsymbol{K}_{\boldsymbol{xy}}^\top \boldsymbol{K}_{\boldsymbol{xy}}(\boldsymbol{K}_{\boldsymbol{y}})^\dagger$ is

$$(\boldsymbol{K}_{\boldsymbol{y}})^\dagger \boldsymbol{K}_{\boldsymbol{xy}}^\top \boldsymbol{K}_{\boldsymbol{xy}}(\boldsymbol{K}_{\boldsymbol{y}})^\dagger = \sum_{i=1}^{m} \xi_i \boldsymbol{w}_i \boldsymbol{w}_i^\top, \tag{26}$$

and let $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_r]$. Then the problem (20) is minimized by

$$P_{\text{SubKPCA}} = T(\boldsymbol{K}_{\boldsymbol{y}}^{1/2})^\dagger \boldsymbol{W}\boldsymbol{W}^\top (\boldsymbol{K}_{\boldsymbol{y}}^{1/2})^\dagger (\boldsymbol{K}_{\boldsymbol{xy}}^\top \boldsymbol{K}_{\boldsymbol{xy}})(\boldsymbol{K}_{\boldsymbol{xy}}^\top \boldsymbol{K}_{\boldsymbol{xy}})^\dagger T^*. \tag{27}$$

Since the solution is rather complex, and we don't find any advantages to use the basis set $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$ such that $\mathcal{R}(T) \not\subset \mathcal{R}(S)$, we henceforth assume that $\mathcal{R}(T) \subset \mathcal{R}(S)$.

3.2.3 Derivation of the solutions

Since the problem (20) is in $\mathcal{R}(T)$, the solution can be parameterized as $X = T\boldsymbol{B}T^*$, $\boldsymbol{B} \in \mathbb{R}^{m\times m}$. Then the objective function is

$$\begin{aligned} J_1(\boldsymbol{B}) =& \frac{1}{n}\|S - T\boldsymbol{B}T^*S\|_F^2 && (28)\\ =& \frac{1}{n}\text{Trace}[\boldsymbol{B}\boldsymbol{K}_{\boldsymbol{xy}}^\top \boldsymbol{K}_{\boldsymbol{xy}}\boldsymbol{B}^\top \boldsymbol{K}_{\boldsymbol{y}} - \boldsymbol{B}^\top \boldsymbol{K}_{\boldsymbol{xy}}^\top \boldsymbol{K}_{\boldsymbol{xy}} - \boldsymbol{B}\boldsymbol{K}_{\boldsymbol{xy}}^\top \boldsymbol{K}_{\boldsymbol{xy}} + \boldsymbol{K}] \\ =& \frac{1}{n}\|\boldsymbol{K}_{\boldsymbol{y}}^{1/2}\boldsymbol{B}\boldsymbol{K}_{\boldsymbol{xy}}^\top - (\boldsymbol{K}_{\boldsymbol{y}}^{1/2})^\dagger \boldsymbol{K}_{\boldsymbol{xy}}^\top\|_F^2 + \frac{1}{n}\text{Trace}[\boldsymbol{K} - \boldsymbol{K}_{\boldsymbol{xy}}\boldsymbol{K}_{\boldsymbol{y}}^\dagger \boldsymbol{K}_{\boldsymbol{xy}}^\top,] && (29) \end{aligned}$$

where the relations $\boldsymbol{K}_{\boldsymbol{xy}}^\top = \boldsymbol{K}_{\boldsymbol{y}}^{1/2}(\boldsymbol{K}_{\boldsymbol{y}}^{1/2})^\dagger \boldsymbol{K}_{\boldsymbol{xy}}^\top$ and $\boldsymbol{K}_{\boldsymbol{xy}} = \boldsymbol{K}_{\boldsymbol{xy}}(\boldsymbol{K}_{\boldsymbol{y}}^{1/2})^\dagger \boldsymbol{K}_{\boldsymbol{y}}^{1/2}$ are used. Since the second term is a constant for $\boldsymbol{B}$, from the Schmidt approximation theorem, The minimum solution is given by the singular value decomposition (SVD) of $(\boldsymbol{K}_{\boldsymbol{y}}^{1/2})^\dagger \boldsymbol{K}_{\boldsymbol{xy}}^\top$,

$$(\boldsymbol{K}_{\boldsymbol{y}}^{1/2})^\dagger \boldsymbol{K}_{\boldsymbol{xy}}^\top = \sum_{i=1}^{m} \sqrt{\xi_i}\boldsymbol{w}_i \boldsymbol{\nu}_i^\top. \tag{30}$$

Then the minimum solution is given by

$$\boldsymbol{K}_{\boldsymbol{y}}^{1/2}\boldsymbol{B}\boldsymbol{K}_{\boldsymbol{xy}}^\top = \sum_{i=1}^{r} \sqrt{\xi_i}\boldsymbol{w}_i \boldsymbol{\nu}_i^\top. \tag{31}$$

From the matrix equation theorem (Israel & Greville, 1973), the minimum solution is given by Eq. (27).

Let us consider the case that $\mathcal{R}(T) \subset \mathcal{R}(S)$.

Lemma 1 (Harville (1997))**.** *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be non-negative definite matrices that satisfy $\mathcal{R}(\boldsymbol{A}) \subset \mathcal{R}(\boldsymbol{B})$. Consider an EVD and a generalized EVD,*

$$(\boldsymbol{B}^{1/2})^{\dagger}\boldsymbol{A}(\boldsymbol{B}^{1/2})^{\dagger}\boldsymbol{v} = \lambda\boldsymbol{v}$$
$$\boldsymbol{A}\boldsymbol{u} = \sigma\boldsymbol{B}\boldsymbol{u},$$

and suppose that $\{(\lambda_i, \boldsymbol{v}_i)\}$ and $\{(\sigma_i, \boldsymbol{u}_i)\}$, $i = 1, 2, \ldots$ are sorted pairs of the eigenvalues and the eigenvectors respectively. Then

$$\begin{aligned}\lambda_i =& \sigma_i \\ \boldsymbol{u}_i =& \alpha(\boldsymbol{B}^{1/2})^{\dagger}\boldsymbol{v}_i, \quad \forall\alpha \in \mathbb{R} \\ \boldsymbol{v}_i =& \beta\boldsymbol{B}^{1/2}\boldsymbol{u}_i, \quad \forall\beta \in \mathbb{R}\end{aligned}$$

are satisfied.

If $\mathcal{R}(T) \subset \mathcal{R}(S)$, $\mathcal{R}(\boldsymbol{K}_{\boldsymbol{xy}}^{\top}) = \mathcal{R}(\boldsymbol{K}_{\boldsymbol{y}})$. Since $(\boldsymbol{K}_{\boldsymbol{xy}}^{\top}\boldsymbol{K}_{\boldsymbol{xy}})(\boldsymbol{K}_{\boldsymbol{xy}}^{\top}\boldsymbol{K}_{\boldsymbol{xy}})^{\dagger}$ is a projector onto $\mathcal{R}(\boldsymbol{K}_{\boldsymbol{y}})$, $(\boldsymbol{K}_{\boldsymbol{y}}^{1/2})^{\dagger}(\boldsymbol{K}_{\boldsymbol{xy}}^{\top}\boldsymbol{K}_{\boldsymbol{xy}})(\boldsymbol{K}_{\boldsymbol{xy}}^{\top}\boldsymbol{K}_{\boldsymbol{xy}})^{\dagger} = (\boldsymbol{K}_{\boldsymbol{y}}^{1/2})^{\dagger}$ in Eq. (27). From Lemma 1, the solution Eq. (22) is derived.

3.3 Computational complexity of SubKPCA

The procedures and computational complexities of SubKPCA are as follows,

1. Select the subset from training samples (discussed in the next Section)
2. Calculate $\boldsymbol{K}_{\boldsymbol{y}}$ and $\boldsymbol{K}_{\boldsymbol{xy}}^{\top}\boldsymbol{K}_{\boldsymbol{xy}}$ [$O(m^2) + O(nm^2)$]
3. Perform generalized EVD, Eq. (21). [$O(rm^2)$]
4. Store $\boldsymbol{Z}$ and the samples in the subset.
5. For an input vector $\boldsymbol{x}$, calculate the empirical kernel map $\boldsymbol{h}_{\boldsymbol{x}}$. [$O(m)$]
6. Obtain transformed vector Eq. (24).

The procedures 1, 2 and 3 are the construction, and 4 and 5 are the evaluation. The dominant calculation in the construction stage is the generalized EVD. In the case of standard KPCA, the size of EVD is n, whereas for SubKPCA, the size of generalized EVD is m. Moreover, for evaluation stage, the computational complexity depends on the size of the subset, m, and required memory to store $\boldsymbol{Z}$ and the subset is also reduced. It means the response time of the system using SubKPCA for an input vector $\boldsymbol{x}$ is faster than standard KPCA.

3.4 Approximation error

It should be shown the approximation error due to the subset approximation. In the case of KPCA, the approximation error, that is the value of the objective function of the problem (16). From Eqs. (17) and (19), The value of J_1 at the minimum solution is

$$J_1 = \frac{1}{n}\sum_{i=r+1}^{n}\lambda_i. \tag{32}$$

In the case of SubKPCA, the approximation error is

$$J_1 = \frac{1}{n}\sum_{i=r+1}^{n}\xi_i + \frac{1}{n}\mathrm{Trace}[\boldsymbol{K} - \boldsymbol{K}_{\boldsymbol{xy}}(\boldsymbol{K}_{\boldsymbol{y}})^{\dagger}\boldsymbol{K}_{\boldsymbol{xy}}^{\top}]. \tag{33}$$

The first term is due to the approximation error for the rank reduction and the second term is due to the subset approximation. Let $P_{\mathcal{R}(S)}$ and $P_{\mathcal{R}(T)}$ be orthogonal projectors onto $\mathcal{R}(S)$ and $\mathcal{R}(T)$ respectively. The second term yields that

$$\mathrm{Trace}[\boldsymbol{K} - \boldsymbol{K}_{\boldsymbol{xy}}(\boldsymbol{K}_{\boldsymbol{y}})^{\dagger}\boldsymbol{K}_{\boldsymbol{xy}}^{\top}] = \mathrm{Trace}[S^*(P_{\mathcal{R}(S)} - P_{\mathcal{R}(T)})S], \tag{34}$$

since $\boldsymbol{K} = S^* P_{\mathcal{R}(S)} S$. Therefore, if $\mathcal{R}(S) = \mathcal{R}(T)$ (for example, the subset contains all training samples), the second term is zero. If the range of the subset is far from the range of the all training set, the second term is large.

3.5 Pre-centering

Although we have assumed that the mean of training vector in the feature space is zero so far, it is not always true in real problems. In the case of PCA, we subtract the mean vector from all training samples when we obtain the variance-covariance matrix $\boldsymbol{\Sigma}$. On the other hand, in KPCA, although we cannot obtain the mean vector in the feature space, $\bar{\Phi} = \frac{1}{n}\sum_{i=1}^{n}\Phi(\boldsymbol{x}_i)$, explicitly, the pre-centering can be set in the algorithm of KPCA. The pre-centering can be achieved by using subtracted vector $\bar{\Phi}(\boldsymbol{x}_i)$, instead of a mapped vector $\Phi(\boldsymbol{x}_i)$,

$$\bar{\Phi}(\boldsymbol{x}_i) = \Phi(\boldsymbol{x}_i) - \bar{\Phi}, \tag{35}$$

that is to say, S and K in Eq. (17) are respectively replaced by

$$\bar{S} = S - \bar{\Phi}\mathbf{1}_n^{\top} = S(\boldsymbol{I} - \frac{1}{n}\mathbf{1}_{n,n}) \tag{36}$$

$$\bar{K} = \bar{S}^*\bar{S} = (\boldsymbol{I} - \frac{1}{n}\mathbf{1}_{n,n})K(\boldsymbol{I} - \frac{1}{n}\mathbf{1}_{n,n}) \tag{37}$$

where $\boldsymbol{I}$ denotes the identify matrix, and $\mathbf{1}_n$ and $\mathbf{1}_{n,n}$ are an n-dimensional vector and an $n \times n$ matrix whose elements are all one, respectively.

For SubKPCA, following three methods to estimate the centroid can be considered,

1. $\bar{\Phi}_1 = \frac{1}{n}\sum_{i=1}^{n}\Phi(\boldsymbol{x}_i)$
2. $\bar{\Phi}_2 = \frac{1}{m}\sum_{i=1}^{m}\Phi(\boldsymbol{y}_i)$
3. $\bar{\Phi}_3 = \underset{\Psi\in\mathcal{R}(T)}{\mathrm{argmin}}\|\Psi - \bar{\Phi}_1\| = \frac{1}{n}T\boldsymbol{K}_{\boldsymbol{y}}^{\dagger}\boldsymbol{K}_{\boldsymbol{xy}}^{\top}\mathbf{1}_n.$

The first one is the same as that of KPCA. The second one is the mean of the basis set. If the basis set is the subset of the criterion set, the estimation accuracy is not as good as $\bar{\Phi}_1$. The third one is the best approximation of $\bar{\Phi}_1$ in $\mathcal{R}(T)$. Since SubKPCA is discussed in $\mathcal{R}(T)$, $\bar{\Phi}_1$ and $\bar{\Phi}_3$ are equivalent. However, for the post-processing such as pre-image, they are not equivalent.

For SubKPCA, only S in Eq. (28) has to be modified for per-centering [4]. If $\bar{\Phi}_3$ is used, S and $\boldsymbol{K}_{\boldsymbol{xy}}$ are replaced by

$$\bar{S} = S - \bar{\Phi}_3 \mathbf{1}_n^\top \tag{38}$$

$$\bar{\boldsymbol{K}}_{\boldsymbol{xy}} = \bar{S}^* T = (\boldsymbol{I} - \frac{1}{n}\mathbf{1}_{n,n})\boldsymbol{K}_{\boldsymbol{xy}}. \tag{39}$$

4. Selection of samples

Selection of samples for the basis set is an important problem in SubKPCA. Ideal criterion for the selection depends on applications such as classification accuracy or PSNR for denoising. We, here, show a simple criterion using empirical error,

$$\begin{aligned} &\min_{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_m} \min_X J_1(X) \\ &\text{Subject to } \operatorname{rank}(X) \le r,\ \mathcal{N}(X) \supset \mathcal{R}(T)^\perp,\ \mathcal{R}(X) \subset \mathcal{R}(T), \\ &\qquad \{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_m\} \subset \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\},\ T = [\Phi(\boldsymbol{y}_1)|\ldots|\Phi(\boldsymbol{y}_m)]. \end{aligned} \tag{40}$$

This criterion is a combinatorial optimization problem for the samples, and it is hard to obtain to global solution if n and m are large. Instead of solving directly, following techniques can be introduced,

1. Greedy forward search
2. Backward search
3. Random sampling consensus (RANSAC)
4. Clustering,

and their combinations.

4.1 Sample selection methods

4.1.1 Greedy forward search

The greedy forward search adds a sample to the basis set one by one or bit by bit. The algorithm is as follows, If several samples are added at 9 and 10, the algorithm is faster, but the cost function may be larger.

4.1.2 Backward search

On the other hand, a backward search removes samples that have the least effect on the cost function. In this case, the standard KPCA using the all samples has to be constructed at the beginning, and this may have very high computational complexity. However, the backward search may be useful in combination with the greedy forward search. In this case, the size of the temporal basis set does not become large, and the value of the cost function is monotonically decreasing.

Sparse KPCA (Tipping, 2001) is a kind of backward procedures. Therefore, the kernel Gram matrix $\boldsymbol{K}$ using all training samples and its inverse have to be calculated in the beginning.

[4] Of course, for KPCA, we can also consider the criterion set and the basis set, and perform pre-centering only for the criterion set. It produces the equivalent result.

Algorithm 1 Greedy forward search (one-by-one version)

Set initial basis set $\mathcal{T} = \phi$, size of current basis set $\tilde{m} = 0$, residual set $\mathcal{S} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, size of the residual set $\tilde{n} = n$.
while $\tilde{m} < m$, **do**
  for $i = 1, \ldots, \tilde{n}$ **do**
    Let temporal basis set be $\tilde{\mathcal{T}} = \mathcal{T} \cup \{\boldsymbol{x}_i\}$
    Obtain SubKPCA using the temporal basis set
    Store the empirical error $E_i = J_1(X)$.
  end for
  Obtain the smallest E_i, $k = \mathrm{argmin}_i E_i$.
  Add $\boldsymbol{x}_k$ to the current basis set, $\mathcal{T} \leftarrow \mathcal{T} \cup \{\boldsymbol{x}_k\}$, $\tilde{m} \leftarrow \tilde{m} + 1$
  Remove $\boldsymbol{x}_k$ from the residual set, $\mathcal{S} \leftarrow \mathcal{S} \backslash \{\boldsymbol{x}_k\}$, $\tilde{n} \leftarrow \tilde{n} - 1$.
end while

4.1.3 Random sampling consensus

RANSAC is a simple sample (or parameter) selection technique. The best basis set is chosen from many random sampling trials. The algorithm is simple to code.

4.1.4 Clustering

Clustering techniques also can be used for sample selection. When the subset is used for the basis set, i) a sample that is the closest to each centroid should be used, or ii) centroids should be included to the criterion set. Clustering in the feature space $\mathcal{F}$ is also proposed (Girolami, 2002).

5. Comparison with conventional methods

This section compares SubKPCA with related conventional methods.

5.1 Improved KPCA

Improved KPCA (IKPCA) (Xu et al., 2007) directly approximates $u_i \simeq T\tilde{\boldsymbol{v}}_i$ in Eq. (7). From $SS^*\boldsymbol{u}_i = \lambda_i u_i$, the approximated eigenvalue problem is

$$SS^*T\tilde{\boldsymbol{v}} = \lambda_i T\tilde{\boldsymbol{v}}_i. \tag{41}$$

By multiplying T^* from left side, one gets the approximated generalized EVD, $\boldsymbol{K}_{\boldsymbol{xy}}^\top \boldsymbol{K}_{\boldsymbol{xy}}\tilde{\boldsymbol{v}} = \lambda_i \boldsymbol{K}_{\boldsymbol{y}}\tilde{\boldsymbol{v}}_i$. The parameter vector $\boldsymbol{v}_i$ is substituted to the relation $\tilde{u}_i = T\tilde{\boldsymbol{v}}_i$, hence, the transform of an input vector $\boldsymbol{x}$ is

$$U^*_{\mathrm{IKPCA}}\Phi(\boldsymbol{x}) = \left(\mathrm{diag}([\frac{1}{\sqrt{\kappa_1}}, \ldots, \frac{1}{\sqrt{\kappa_r}}])\right)\boldsymbol{Z}^\top \boldsymbol{h}_{\boldsymbol{x}}, \tag{42}$$

where κ_i is the ith largest eigenvalue of (21).

This approximation has no guarantee to be good approximation of u_i. In our experiments in the next section, IKPCA showed worse performance than SubKPCA. In so far as feature extraction, each dimension of the feature vector is multiplied by $\frac{1}{\sqrt{\kappa_i}}$ comparing to SubKPCA. If the classifier accepts such linear transforms, the classification accuracy of feature vectors

may be the same with SubKPCA. Indeed, (Xu et al., 2007) uses IKPCA only for feature extraction of a classification problem, and IKPCA shows good performance.

5.2 Sparse KPCA

Two methods to obtain a sparse solution to KPCA are proposed (Smola et al., 1999; Tipping, 2001). Both approaches focus on reducing the computational complexity in the evaluation stage, and do not consider that in the construction stage. In addition, the degree of sparsity cannot be tuned directly for these sparse KPCAs, where as the number of the subset m can be tuned for SubKPCA.

As mentioned in Section 4.1.2, (Tipping, 2001) is based on a backward search, therefore, it requires to calculate the kernel Gram matrix using all training samples, and its inverse. These procedures have high computational complexity, especially, when n is large.

(Smola et al., 1999) utilizes l_1 norm regularization to make the solution sparse. The principal components are represented by linear combinations of mapped samples, $u_i = \sum_{j=1}^{n} \alpha_i^j \Phi(\boldsymbol{x}_j)$. The coefficients α_i^j have many zero entry due to l_1 norm regularization. However, since α_i^j has two indeces, even if each principal component u_i is represented by a few samples, it may not be sparse for many i.

5.3 Nyström approximation

Nyström approximation is a method to approximate EVD, and it is applied to KPCA (Williams & Seeger, 2001). Let $\tilde{\boldsymbol{u}}_i$ and $\boldsymbol{u}_i$ be the ith eigenvectors of $\boldsymbol{K_y}$ and $\boldsymbol{K}$ respectively. Nyström approximation approximates

$$\tilde{\boldsymbol{v}}_i = \sqrt{\frac{m}{n}} \frac{1}{\lambda_i} \boldsymbol{K_{xy}} \boldsymbol{v}_i, \tag{43}$$

where λ_i is the ith eigenvalue of $\boldsymbol{K_y}$. Since the eigenvector of $\boldsymbol{K_x}$ is approximated by the eigenvector of $\boldsymbol{K_y}$, the computational complexity in the construction stage is reduced, but that in the evaluation stage is not reduced. In our experiments, SubKPCA shows better performance than Nyström approximation.

5.4 Iterative KPCA

There are some iterative approaches for KPCA (Ding et al., 2010; Günter et al., 2007; Kim et al., 2005). They update the transform matrix $\boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}_{\mathrm{KPCA}}^{\top}$ in Eq. (14) for incoming samples.

Iterative approaches are sometimes used for reduction of computational complexities. Even if optimization step does not converge to the optimal point, early stopping point may be a good approximation of the optimal solution. However, Kim et al. (2005) and Günter et al. (2007) do not compare their computational complexity with standard KPCA. In the next section, comparisons of run-times show that iterative KPCAs are not faster than batch approaches.

5.5 Incomplete Cholesky decomposition

ICD can also be used for reduction of computational complexity of KPCA. ICD approximates the kernel Gram matrix $\boldsymbol{K}$ by

$$\boldsymbol{K} \simeq \boldsymbol{G}\boldsymbol{G}^{\top}, \tag{44}$$

where $\boldsymbol{G} \in \mathbb{R}^{n\times m}$ whose upper triangle part is zero, and m is a parameter that specifies the trade-off between approximation accuracy and computational complexity. Instead of performing EVD of $\boldsymbol{K}$, eigenvectors of $\boldsymbol{K}$ is obtained from EVD of $\boldsymbol{G}^\top\boldsymbol{G} \in \mathbb{R}^{m\times m}$ using the relation Eq. (7) approximately. Along with Nyström approximation, ICD reduces computational complexity in the construction stage, but not in evaluation stage, and all training samples have to be stored for the evaluation.

In the next section, our experimental results indicate that ICD is slower than SubKPCA for very large dataset, n is more than several thousand.

ICD can also be applied to SubKPCA. In Eq. (21), $\boldsymbol{K}_{\boldsymbol{xy}}^\top\boldsymbol{K}_{\boldsymbol{xy}}$ is approximated by

$$\boldsymbol{K}_{\boldsymbol{xy}}^\top\boldsymbol{K}_{\boldsymbol{xy}} \simeq \boldsymbol{G}\boldsymbol{G}^\top. \tag{45}$$

Then approximated $\boldsymbol{z}$ is obtained from EVD of $\boldsymbol{G}^\top\boldsymbol{K}_{\boldsymbol{y}}\boldsymbol{G}$.

6. Numerical examples

This section presents numerical examples and numerical comparisons with the other methods.

6.1 Methods and evaluation criteria

At first, methods to be compared and evaluation criteria are described. Following methods are compared,

1. SubKPCA [SubKp]
2. Full KPCA [FKp]
   Standard KPCA using all training samples.
3. Reduced KPCA [RKp]
   Standard KPCA using subset of training samples.
4. Improved KPCA (Xu et al., 2007) [IKp]
5. Sparse KPCA (Tipping, 2001) [SpKp]
6. Nyström approximation (Williams & Seeger, 2001) [Nys]
7. ICD (Bach & Jordan, 2002) [ICD]
8. Kernel Hebbian algorithm with stochastic meta-decent (Günter et al., 2007) [KHA-SMD]

Abbreviations in [] are used in Figures and Tables.

For evaluation criteria, the empirical error that is J_1, is used.

$$E_{\text{emp}}(X) = J_1(X) = \frac{1}{n}\sum_{i=1}^{n}\|\Phi(\boldsymbol{x}_i) - X\Phi(\boldsymbol{x}_i)\|^2, \tag{46}$$

where X is replaced by each operator. Note that full KPCA gives the minimum values for $E_{\text{emp}}(X)$ under the rank constraint. Since $E_{\text{emp}}(X)$ depends on the problem, normalized by that of full KPCA is also used, $E_{\text{emp}}(X)/E_{\text{emp}}(P_{\text{Fkp}})$, where P_{FKp} is a projector of full KPCA. Validation error E_{val} that uses validation samples instead of training samples in the empirical error is also used.

Sample selection	SubKPCA	Reduced KPCA	Improved KPCA	Nyström method
Random	1.0025±0.0019	1.1420±0.0771	4.7998±0.0000	2.3693±0.6826
K-means	1.0001±0.0000	1.0282±0.0114	4.7998±0.0000	1.7520±0.2535
Forward	1.0002±0.0001	1.3786±0.0719	4.7998±0.0000	14.3043±9.0850
Random	0.0045±0.0035	0.2279±0.1419	0.9900±0.0000	0.3583±0.1670
K-means	0.0002±0.0001	0.0517±0.0318	0.9900±0.0000	0.1520±0.0481
Forward	0.0002±0.0001	0.5773±0.0806	0.9900±0.0000	1.7232±0.9016

Table 1. Mean values and standard deviations over 10 trials of E_{emp} and D in Experiment 1: Upper rows are $E_{\mathrm{emp}}(X)/E_{\mathrm{emp}}(P_{\mathrm{FKp}})$; lower rows are D; Sparse KPCA does not require sample selection.

The alternative criterion is operator distance from full KPCA. Since these methods are approximation of full KPCA, an operator that is closer to that of full KPCA is the better one. In the feature space, the distance between projectors is measured by the Frobenius distance,

$$D(X, P_{\mathrm{FKp}}) = \|X - P_{\mathrm{FKp}}\|_F. \tag{47}$$

For example, if $X = P_{\mathrm{SubKPCA}} = T\boldsymbol{Z}\boldsymbol{Z}^\top T^*$ (Eq. (27)),

$$\begin{aligned} D^2(P_{\mathrm{SubKPCA}}, P_{\mathrm{FKp}}) =& \|T\boldsymbol{Z}\boldsymbol{Z}^\top T^* - S\boldsymbol{V}_{\mathrm{KPCA}}\boldsymbol{\Lambda}^{-1}\boldsymbol{V}_{\mathrm{KPCA}}^\top\|_F^2 \\ =& \mathrm{Trace}[\boldsymbol{Z}^\top \boldsymbol{K}_{\boldsymbol{y}}\boldsymbol{Z} + \boldsymbol{V}_{\mathrm{KPCA}}^\top \boldsymbol{K}\boldsymbol{V}_{\mathrm{KPCA}}\boldsymbol{\Lambda}^{-1} \\ & - 2\boldsymbol{Z}^\top \boldsymbol{K}_{\boldsymbol{xy}}^\top \boldsymbol{V}_{\mathrm{KPCA}}\boldsymbol{\Lambda}^{-1}\boldsymbol{V}_{\mathrm{KPCA}}^\top \boldsymbol{K}_{\boldsymbol{xy}}\boldsymbol{Z}]. \end{aligned}$$

6.2 Artificial data

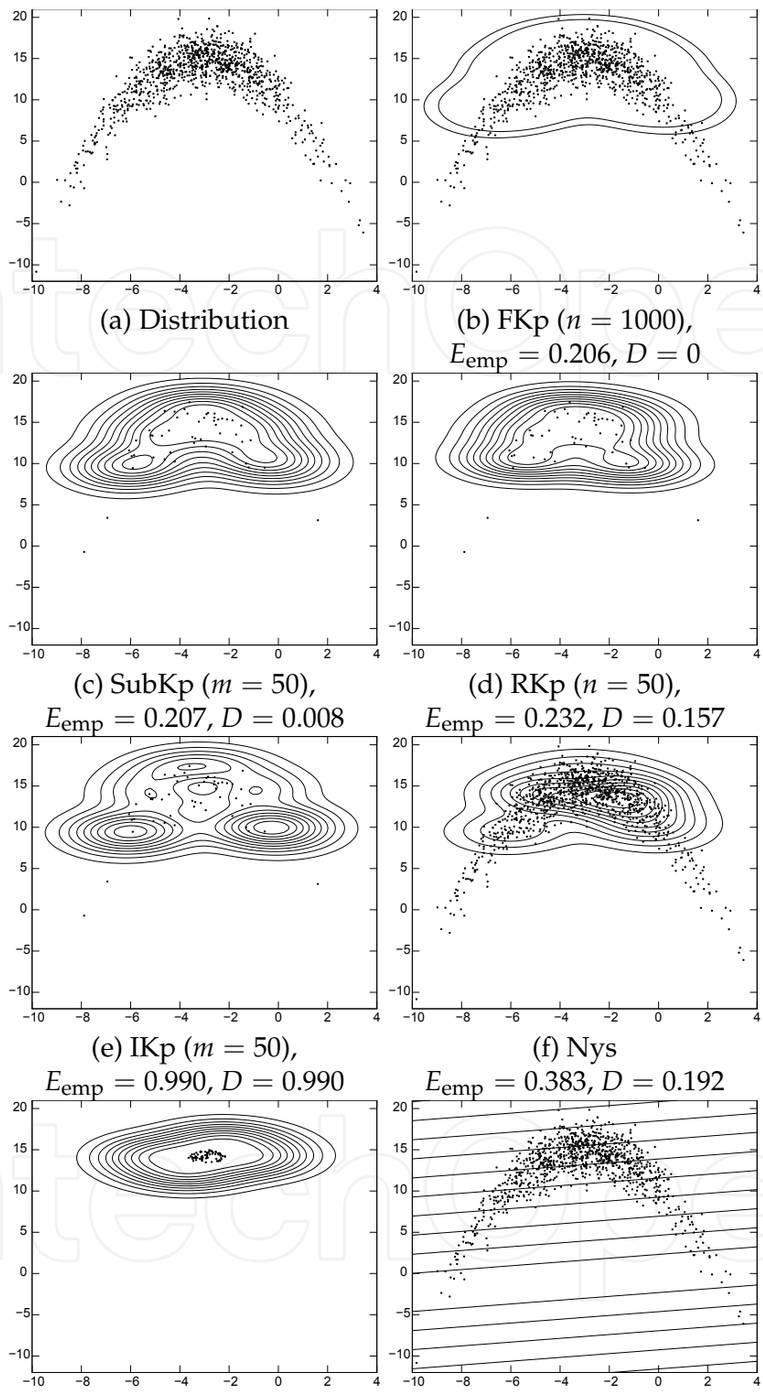
Two-dimensional artificial data described in Introduction is used again with more comparisons and quantitative evaluation. Gaussian kernel function $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-0.1\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2)$ and the number of principal components, $r = 5$ are chosen. Training samples of Reduced KPCA and the basis set of SubKPCA, Nyström approximation, and IKPCA are identical, and chosen randomly. For Sparse KPCA (SpKp), a parameter σ is chosen to have the same sparsity level with SubKPCA. Figure 2 shows contour curves and values of evaluation criteria. From evaluation criteria E_{emp} and D, SubKp shows the best approximation accuracy among these methods.

Table 1 compares sample selection methods. The values in the table are the mean values and standard deviations over 10 trials using different random seeds or initial point. SubKPCA performed better than the other methods. Regarding sample selection, K-means and forward search give almost the same results for SubKPCA.

6.3 Open dataset

Three open benchmark datasets, "concrete," "housing," and "tic" from UCI (University of California Irvine) machine learning repository are used [5] (Asuncion & Newman, 2007). Table 2 shows properties of the datasets.

[5] As of Oct. 2011, the datasets are available from `http://archive.ics.uci.edu/ml/index.html`



(a) Distribution

(b) FKp ($n = 1000$), $E_{\text{emp}} = 0.206$, $D = 0$

(c) SubKp ($m = 50$), $E_{\text{emp}} = 0.207$, $D = 0.008$

(d) RKp ($n = 50$), $E_{\text{emp}} = 0.232$, $D = 0.157$

(e) IKp ($m = 50$), $E_{\text{emp}} = 0.990$, $D = 0.990$

(f) Nys $E_{\text{emp}} = 0.383$, $D = 0.192$

(g) SpKp ($\sigma = 0.3$, $n = 69.$) $E_{\text{emp}} = 0.357$, $D = 0.843$

(h) $r = 1$

Fig. 2. Contour curves of projection norms

Gaussian kernel $k(\boldsymbol{x}_1.\boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2/(2\sigma^2))$ whose σ^2 is set to be the variance of for all elements of each dataset is used for the kernel function. The number of principal

dataset	no. of dim.	no. of samples
concrete	9	1030
housing	14	506
tic	85	9822

Table 2. Open dataset

components, r, is set to be the input dimension of each dataset. 90% of samples are used for training, and the remaining 10% of samples are used for validation. The division of the training and the validation sets is repeated 50 times randomly.

Figures 3-(a) and (b) show the averaged squared distance from KPCA using all samples. SubKPCA shows better performance than Reduced KPCA and the Nyström method, especially SubKPCA with a forward search performed the best of all. In both datasets, even if the number of basis is one of tenth that of all samples, the distance error of SubKPCA is less than 1%.

Figures 3-(c) and (d) show the average normalized empirical error, and Figures (e) and (f) show the averaged validation error. SubKPCA with K-means or forward search performed the best, and its performance did not change much with 20% more basis. The results for the Nyström method are outside of the range illustrated in the figures.

Figures 4-(a) and (b) show the calculation times for construction. The simulation was done on the system that has an Intel Core 2 Quad CPU 2.83GHz and an 8Gbyte RAM. The routines `dsygvx` and `dsyevx` in the Intel math kernel library (MKL) were respectively used for the generalized eigenvalue decomposition of SubKPCA and the eigenvalue decomposition of KPCA. The figures indicate that SubKPCA is faster than Full KPCA if the number of basis is less than 80%.

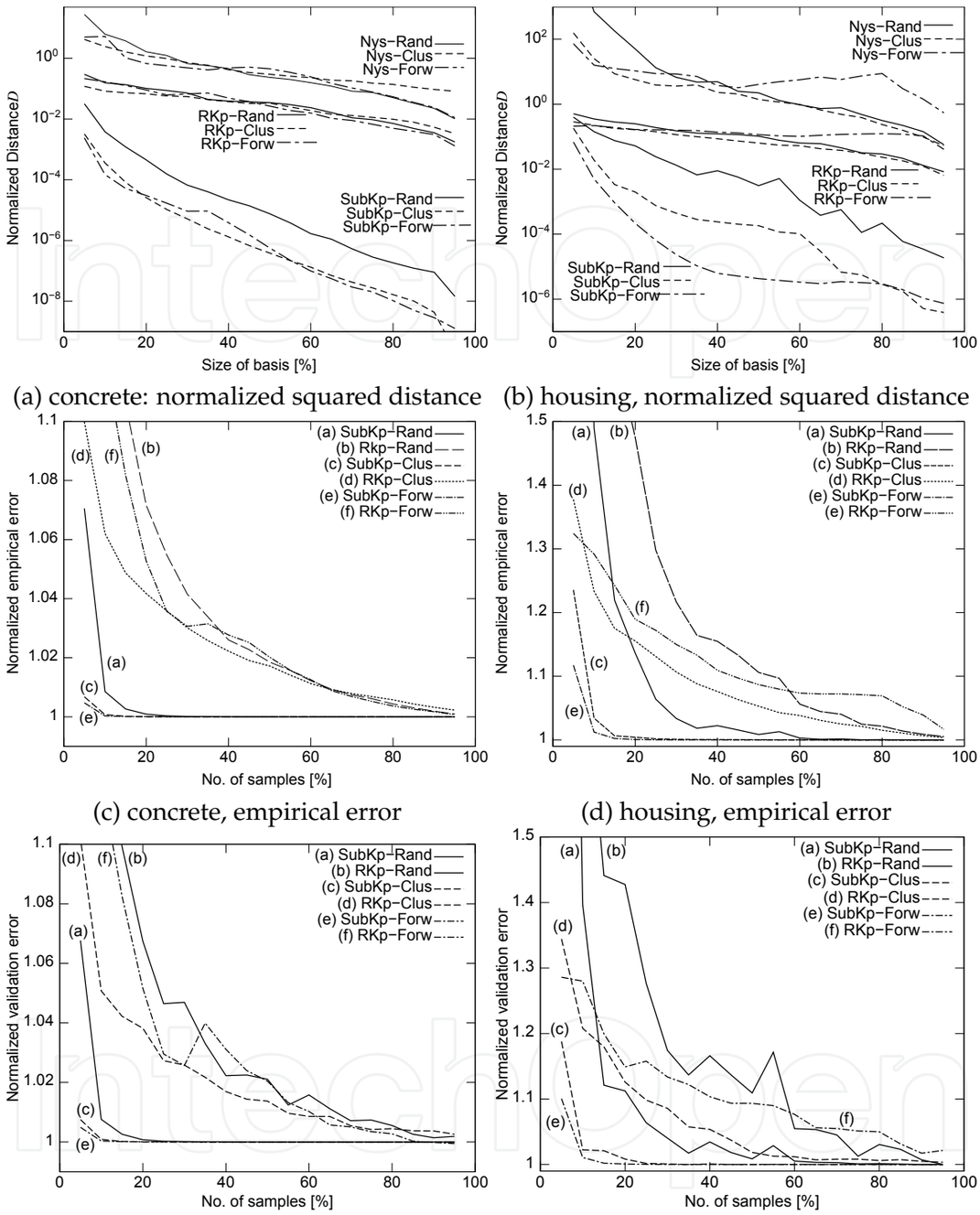
Figure 5 shows the relation between runtime [s] and squared distance from Full KPCA. In this figure, "kmeans" includes runtime for K-means clustering. The vertical dotted line stands for run-time of full KPCA. For (a) concrete and (b) housing, incomplete Cholesky decomposition is faster than our method. However, for a larger dataset, (c) tic, incomplete Cholesky decomposition is slower than our method. KHA-SMD Günter et al. (2007) is slower than full KPCA in these three methods.

6.4 Classification

PCA and KPCA are also used for classifier as subspace methods (Maeda & Murase, 1999; Oja, 1983; Tsuda, 1999). Subspace methods obtain projectors onto subspaces that correspond with classes. Let $\boldsymbol{P}_i$ be a projector onto the subspace of the class i. In the class feature information compression (CLAFIC) that is one of the subspace methods, $\boldsymbol{P}_i$ is a projector of PCA for each class. Then an input sample $\boldsymbol{x}$ is classified to a class k whose squared distance is the largest, that is,

$$k = \underset{i=1,\ldots,c}{\operatorname{argmax}} \|\boldsymbol{x} - \boldsymbol{P}_i \boldsymbol{x}\|^2, \tag{48}$$

where c is the number of classes. Binary classifiers such as SVM cannot be applied to multi-class problems directly, therefore, some extentions such as one-against-all strategy have to be used. However, subspace methods can be applied to many-class problems



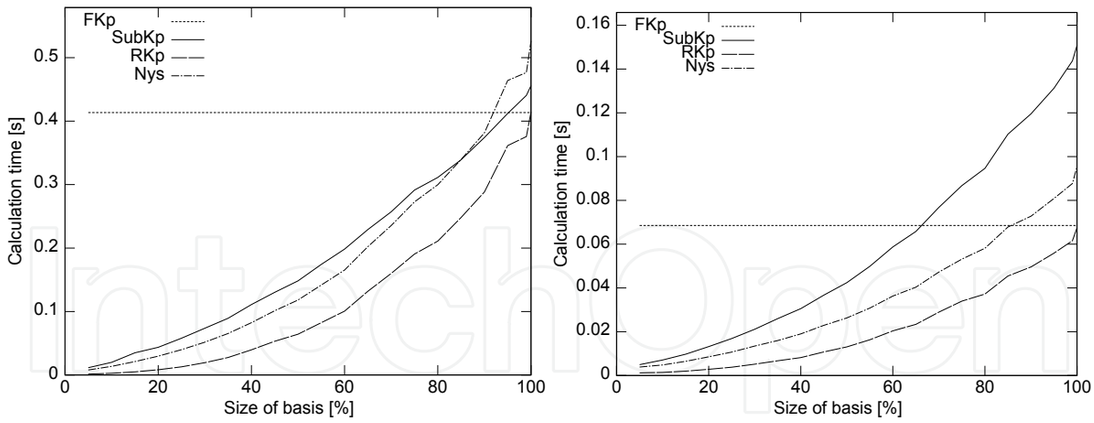
(a) concrete: normalized squared distance (b) housing, normalized squared distance

(c) concrete, empirical error (d) housing, empirical error

(e) concrete, validation error (f) housing, validation error

Fig. 3. Results for open datasets. Rand: random, Clus: Clustering (K-means), Forw: Forward search

easily. Furthermore, subspace methods are easily to be applied to multi-label problems or class-addition/reduction problems. CLAFIC is easily extended to KPCA (Maeda & Murase, 1999; Tsuda, 1999).



(a) Concrete (b) Housing

Fig. 4. Calculation time

	No. of basis			
Method	10%	30%	50%	100%
SubKp (nc)	3.89±0.82	2.87±0.71	2.55±0.64	2.03±0.56
SubKp (c)	3.93±0.82	2.83±0.70	2.55±0.64	2.02±0.56
RKp (nc)	4.92±0.95	3.17±0.71	2.65±0.60	2.03±0.56
RKp (c)	4.91±0.92	3.16±0.70	2.65±0.62	2.02±0.56
CLAFIC (nc)	5.24±0.92	3.64±0.79	3.25±0.70	2.95±0.73
CLAFIC (c)	5.24±0.89	3.71±0.76	3.38±0.68	3.06±0.74

Table 3. Minimum validation errors [%] and standard deviations I; random selection, nc: non-centered, c: centered

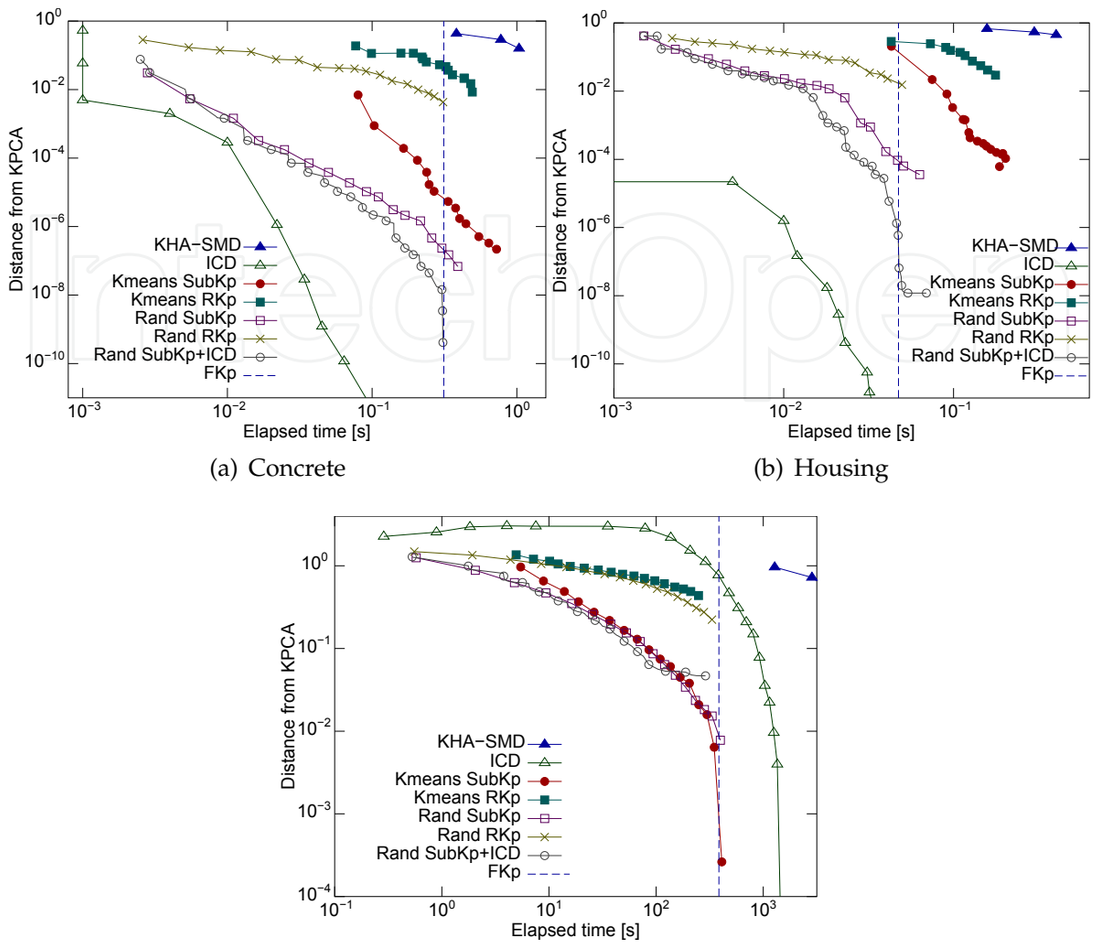
A handwritten digits database, USPS (U.S. postal service database), is used for the demonstration. The database has 7291 images for training, and 2001 images for testing. Each image is 16x16 pixel gray-scale, and has a label $(0, \ldots, 9)$.

10% of samples (729 samples) from training set are extracted for validation, and rest 90% (6562 samples) are used for training. This division is repeated 100 times, and obtained the optimal parameters from several picks, width of Gaussian kernel $c \in \{10^{-4.0}, 10^{-3.8}, \ldots, 10^{0.0}\}$, the number of principal components $r \in \{10, 20, \ldots, 200\}$.

Tables 3 and 4 respectively show the validation errors [%] and standard deviations over 100 validations when the samples of the basis are selected randomly and by k-means respectively. SubKPCA has lower error rate than reduced KPCA when the number of basis is small. Tables 5 and 6 show the test errors when the optimum parameters are given by the validation.

6.5 Denoising using a huge dataset

KPCA is also used for image denoising (Kim et al., 2005; Mika et al., 1999). This subsection demonstrate image denoising by KPCA using MNIST database. The database has 60000 images for training, and 10000 samples for testing. Each image is a 28x28 pixel gray-scale image of a handwritten digit. Each pixel value of the original image is scaled from 0 to 255.



(a) Concrete (b) Housing

(c) Tic

Fig. 5. Relation between runtime [s] and squared distance from Full KPCA

	No. of basis			
Method	10%	30%	50%	100%
SubKp (nc)	2.69±0.58	2.30±0.61	2.18±0.58	2.03±0.56
SubKp (c)	2.68±0.60	2.29±0.61	2.15±0.55	2.02±0.56
RKp (nc)	2.75±0.61	2.35±0.60	2.22±0.57	2.03±0.56
RKp (c)	2.91±0.63	2.40±0.59	2.24±0.58	2.02±0.56
PCA (nc)	3.60±0.66	3.38±0.60	3.21±0.56	3.03±0.60

Table 4. Minimum validation errors [%] and standard deviations II; K-means, nc: non-centered, c: centered

Before the demonstration of image denoising, comparisons of computational complexities are presented since the database has rather large data. The Gaussian kernel function $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-10^{-5.1}\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2)$ and the number of principal components $r = 145$ are used because these parameters show the best result in latter denoising experiment. The random selection

	No. of basis			
Method	10%	30%	50%	100%
SubKp (nc)	6.50±0.36	5.69±0.15	5.20±0.14	4.78±0.00
SubKp (c)	6.54±0.36	5.52±0.16	5.20±0.14	4.83±0.00
RKp (nc)	7.48±0.44	5.71±0.31	5.26±0.20	4.78±0.00
RKp (c)	7.50±0.43	5.76±0.31	5.28±0.20	4.83±0.00

Table 5. Test errors [%] and standard deviations; random selection, nc: non-centered, c: centered

	No. of basis			
Method	10%	30%	50%	100%
SubKp (nc)	5.14±0.17	4.99±0.14	4.97±0.13	4.78±0.00
SubKp (c)	5.14±0.18	4.99±0.14	4.87±0.14	4.83±0.00
RKp (nc)	5.18±0.21	5.01±0.16	4.89±0.15	4.78±0.00
RKp (c)	5.36±0.24	5.07±0.17	4.93±0.15	4.83±0.00

Table 6. Test errors [%] and standard deviations; K-means, nc: non-centered, c: centered

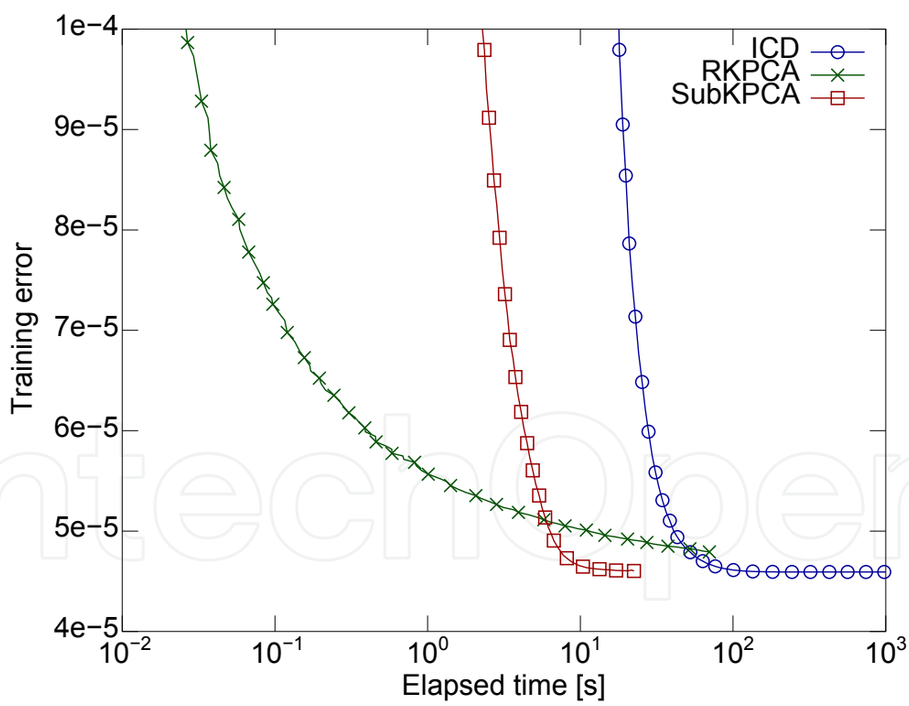


Fig. 6. Relation between training error and elapsed time in MNIST dataset

is used for basis of the SubKPCA. Figure 6 shows relation between run-time and training error. SubKPCA achieves lower training error $E_{\text{emp}} = 4.57 \times 10^{-5}$ in 28 seconds, whereas ICD achieves $E_{\text{emp}} = 4.59 \times 10^{-5}$ in 156 seconds.

Denoising is done by following procedures,

1. Rescale each pixel value from 0 to 1.
2. Obtain the subset using K-means clustering from 60000 training samples.
3. Obtain operators,
   (a) Obtain centered SubKPCA using 60000 training samples and the subset.
   (b) Obtain centered KPCA using the subset.
4. Prepare noisy images using 10000 test samples;
   (a) Add Gaussian noise whose variance is σ^2.
   (b) Add salt-and-pepper noise with a probability of p (a pixel flips white (1) with probability $p/2$, and flips black (0) with probability $p/2$).
5. Obtain each transformed vector and pre-image using the method in (Mika et al., 1999).
6. Rescale each pixel value from 0 to 255, and truncate values if the values less than 0 or grater than 255.

The evaluation criterion is the mean squared error

$$E_{\mathrm{MSE}} = \frac{1}{10000} \sum_{i=1}^{10000} \|\boldsymbol{f}_i - \hat{\boldsymbol{f}}_i\|^2, \tag{49}$$

where $\boldsymbol{f}_i$ is the ith original test image, and $\hat{\boldsymbol{f}}$ is its denoising image. The optimal parameters, r: the number of principal components, and c: parameter of the Gaussian kernel, are chosen to show the best performance in several picks $r \in \{5, 10, 15, \ldots, m\}$ and $c \in \{10^{-6.0}, 10^{-5.9}, \ldots, 10^{-2.0}\}$.

Tables 7 and 8 are denoising results. SubKPCA shows always lower errors than errors of Reduced KPCA. Figures 7 show the original images, noisy images, and de-noised images. Fields of experts (FoE) Roth & Black (2009) and block-matching and 3D filtering (BM3D) Dabov et al. (2007) are state-of-the-art denoising methods for natural images [6]. FoE and BM3D

σ	20	50	80	100
SubKp (100)	3.38±1.37	4.64±1.49	6.73±1.70	8.33±2.17
RKp (100)	3.48±1.42	4.71±1.51	6.80±1.74	8.55±2.11
SubKp (500)	0.99±0.24	3.64±0.82	6.22±1.43	7.95±1.91
RKp (500)	1.01±0.27	3.73±0.81	6.39±1.51	8.14±2.01
SubKp (1000)	0.93±0.22	3.20±0.83	5.11±1.67	6.18±2.02
RKp (1000)	0.94±0.20	3.27±0.87	5.49±1.60	7.18±1.88
WF	0.88±0.24	3.14±0.81	5.49±1.43	7.01±1.84
FoE	1.15±2.08	8.48±0.78	23.29±1.90	36.53±2.81
BM3D	1.07±1.80	7.17±1.09	17.39±2.96	25.49±4.17

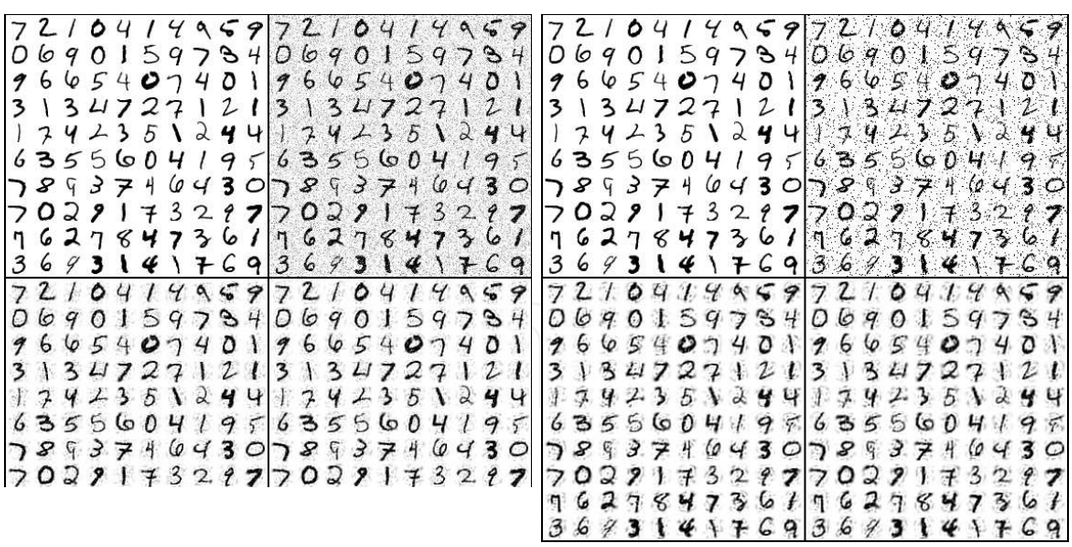
Table 7. Denoising results for Gaussian noise , mean and SD of squared errors, values are divided by 10^5; the numbers in brackets denote the numbers of basis

[6] MATLAB codes were downloaded from `http://www.gris.tu-darmstadt.de/~sroth/research/foe/downloads.html`. and `http://www.cs.tut.fi/~foi/GCF-BM3D/index.html`

p	0.05	0.10	0.20	0.40
SubKp (100)	4.73±1.51	6.45±1.73	10.07±2.16	18.11±3.06
RKp (100)	4.79±1.54	6.52±1.72	10.51±2.30	18.80±3.30
SubKp (500)	3.61±1.00	5.73±1.34	9.66±2.02	17.87±2.95
RKp (500)	3.72±1.00	6.00±1.39	10.04±2.13	18.31±3.05
SubKp (1000)	3.22±0.98	4.99±1.46	7.93±2.22	13.58±4.72
RKp (1000)	3.25±1.00	5.15±1.53	8.78±2.22	18.21±2.96
WF	3.15±0.87	5.15±1.17	8.93±1.72	17.07±2.61
Median	3.26±1.44	4.34±1.70	7.36±2.45	21.78±5.62

Table 8. Denoising results for salt-and-pepper noise , mean and SD of squared errors, values are divided by 10^5; the numbers in brackets denote the numbers of basis

are assumed that the noise is Gaussian whose mean is zero and variance is known. Thus these two methods are compared only in Gaussian noise case. Since the datasets is not natural images, these methods are not better than SubKPCA. "WF" and "Median" denote Wiener filter and median filter respectively. When noise is relatively small, ($\sigma = 20 \sim 50$ in Gaussian or $p = 0.05 \sim 0.10$), these classical methods show better performance. On the other hand, when noise is large, our method shows better performance. Note that Wiener filter is known to be the optimal filter in terms of the mean squared error among linear operators. From different point of view, Wiener filter is optimal among all linear and non-linear operators if both signal and noise are Gaussian. However, KPCA is non-linear because of non-linear mapping Φ, and pixel values of images and salt-and-pepper noise are not Gaussian in this case.



(a) Gaussian noise (b) Salt-and-pepper-noise

Fig. 7. Results of denoising (first 100 samples), top-left: original image, top-right: noisy image (Gaussian, $\sigma = 50$), bottom-left: image de-noised by SubKPCA, bottom-right: image de-noised by KPCA.

7. Conclusion

Theories, properties, and numerical examples of SubKPCA have been presented in this chapter. SubKPCA has a simple solution form Eq. (22) and no constraint for its kernel functions. Therefore, SubKPCA can be applied to any applications of KPCA. Furthermore, it should be emphasized that SubKPCA is always better than reduced KPCA in the sense of the empirical errors if the subset is the same.

8. References

Asuncion, A. & Newman, D. (2007). UCI machine learning repository. URL: *http://www.ics.uci.edu/∼mlearn/MLRepository.html*

Bach, F. R. & Jordan, M. I. (2002). Kernel independent component analysis, *Journal of Machine Learning Research* 3: 1–48.

Dabov, K., Foi, A., Katkovnik, V. & Egiazarian, K. (2007). Image denoising by sparse 3D transform-domain collaborative filtering, *IEEE Trans. on Image Processing* 16(8): 2080–2095.

Demmel, J. (1997). *Applied Numerical Linear Algebra*, Society for Industrial Mathematics.

Ding, M., Tian, Z. & Xu, H. (2010). Adaptive kernel principal component analysis, *Signal Processing* 90(5): 1542–1553.

Girolami, M. (2002). Mercer kernel-based clustering in feature space, *IEEE Trans. on Neural Networks* 13(3): 780–784.

Gorban, A., Kégl, B., Wunsch, D. & (Eds.), A. Z. (2008). *Principal Manifolds for Data Visualisation and Dimension Reduction*, LNCSE 58, Springer.

Günter, S., Schraudolph, N. N. & Vishwanathan, S. V. N. (2007). Fast iterative kernel principal component analysis, *Journal of Machine Learning Research* 8: 1893–1918.

Harville, D. A. (1997). *Matrix Algebra From a Statistician's Perspective*, Springer-Verlag.

Hastie, T. & Stuetzle, W. (1989). Principal curves, *Journal of the American Statistical Association* Vol. 84(No. 406): 502–516.

Israel, A. B. & Greville, T. N. E. (1973). *Generalized inverses, Theorey and applications*, Springer.

Kim, K., Franz, M. O. & Schölkopf, B. (2005). Iterative kernel principal component analysis for image modeling, *IEEE Trans. Pattern Analysis and Machine Intelligence* 27(9): 1351–1366.

Lehoucq, R. B., Sorensen, D. C. & Yang, C. (1998). *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Software, Environments, and Tools 6, SIAM.

Maeda, E. & Murase, H. (1999). Multi-category classification by kernel based nonlinear subspace method, *IEEE International Conference On Acoustics, speech, and signal processing (ICASSP)*, Vol. 2, IEEE press., pp. 1025–1028.

Mika, S., Schölkopf, B. & Smola, A. (1999). Kernel PCA and de-noising in feature space, *Advances in Neural Information Processing Systems (NIPS)* 11: 536–542.

Oja, E. (1983). *Subspace Methods of Pattern Recognition*, Wiley, New-York.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization, *in* B. Scholkopf, C. Burges & A. J. Smola (eds), *Advances in Kernel Methods - Support Vector Learning*, MIT press, pp. 185–208.

Roth, S. & Black, M. J. (2009). Fields of experts, *International Journal of Computer Vision* 82(2): 205–229.

Roweis, S. T. & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding, *Science* 290: 2323–2326.

Schölkopf, B., Mika, S., Burges, C., Knirsch, P., Müller, K.-R., Rätsch, G. & Smola, A. (1999). Input space vs. feature space in kernel-based methods, *IEEE Trans. on Neural Networks* 10(5): 1000–1017.

Schölkopf, B., Smola, A. & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation* 10(5): 1299–1319.

Smola, A. J., Mgngasarian, O. L. & Schölkopf, B. (1999). Sparse kernel feature analysis, *Technical report 99-04, University of Wisconsin* .

Tenenbaum, J. B., de Silva, V. & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction, *Science* 290: 2319–2323.

Tipping, M. E. (2001). Sparse kernel principal component analysis, *Advances in Neural Information Processing Systems (NIPS)* 13: 633–639.

Tsuda, K. (1999). Subspace classifier in the Hilbert space, *Pattern Recognition Letters* 20: 513–519.

Vapnik, V. (1998). *Statistical Learning Theory*, Wiley, New-York.

Washizawa, Y. (2009). Subset kernel principal component analysis, *Proceedings of 2009 IEEE International Workshop on Machine Learning for Signal Processing*, IEEE, pp. 1–6.

Williams, C. K. I. & Seeger, M. (2001). Using the Nyström method to speed up kernel machines, *Advances in Neural Information Processing Systems (NIPS)* 13: 682–688.

Xu, Y., Zhang, D., Song, F., Yang, J., Jing, Z. & Li, M. (2007). A method for speeding up feature extraction based on KPCA, *Neurocomputing* pp. 1056–1061.

Principal Component Analysis

Edited by Dr. Parinya Sanguansat

ISBN 978-953-51-0195-6

Hard cover, 300 pages

Publisher InTech

Published online 02, March, 2012

Published in print edition March, 2012

This book is aimed at raising awareness of researchers, scientists and engineers on the benefits of Principal Component Analysis (PCA) in data analysis. In this book, the reader will find the applications of PCA in fields such as image processing, biometric, face recognition and speech processing. It also includes the core concepts and the state-of-the-art methods in data analysis and feature extraction.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yoshikazu Washizawa (2012). Subset Basis Approximation of Kernel Principal Component Analysis, Principal Component Analysis, Dr. Parinya Sanguansat (Ed.), ISBN: 978-953-51-0195-6, InTech, Available from: http://www.intechopen.com/books/principal-component-analysis/subset-basis-approximation-of-kernel-principal-component-analysis

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821