Sparse Kernel Principal Component Analysis Based on Elastic Net Regularization

Duo Wang
Department of Electronic and
Information Engineering
Tokyo University of Agriculture and Technology
Email: wangduo@sip.tuat.ac.jp

Toshihisa Tanaka
Department of Electronic and
Information Engineering
Tokyo University of Agriculture and Technology
Email: tanakat@cc.tuat.ac.jp

***Abstract*—In this paper, inspired by sparse principal component analysis (SPCA) via the elastic net regularization, we propose a new criterion for sparsification of the kernel principal component analysis (KPCA) with the elastic net regularization that can simultaneously consider the data approximation and sparsification. We first show that KPCA also can be relaxed into a regression framework optimization problem, with a quadratic penalty; l_1-norm can then be integrated into the regression criterion, leading to a new cost function. The minimization is iteratively conducted together with alternating direction method of multipliers. Experimental results for toy examples and real world data support the analysis.**

I. Introduction

Principal component analysis (PCA) [1] is a linear feature extraction method from possibly high-dimensional data. Usually, there are two viewpoints to define PCA [2]. One standpoint based on maximum variance criterion. Another view is determined by the criterion of mean square error (MSE). Although the angles of definitions are different, both can be solved by the eigenvectors of the covariance matrix of the data. The direction of these eigenvectors defines the subspace, the inner product between a data vector and eigenvectors are referred to as principal components. However, one of the shortcomings of PCA is that it fails to extract nonlinear structures in the input data. To overcome this problem, kernel principal component analysis (KPCA) [3] has been proposed and widely used. In KPCA, the input data vectors are mapped into some feature space through a nonlinear mapping. Instead of directly using nonlinear mapping, KPCA uses a so-called kernel trick to compute the inner product in the feature space, and then implements PCA on the mapped data.

However, the disadvantage of KPCA is that the principal components (PCs) are expressed by linear combinations of all kernel functions associated with data. Thus, all data must be stored, which is inefficient and impractical in real-world application for large-scale data. Hence, it is necessary to limit the number of nonzero coefficients in linear combinations and keep the original quality. From this perspective, several algorithms for sparsification of the KPCA have been studied [4]–[12]. Smola et al. [4] proposed sparse kernel feature analysis (SKFA), which utilizes an l_1-constraint on coefficients to extract features. To improve the efficiency of SFKA, the authors of [8] proposed the accelerated kernel feature analysis. However, the use of contrast function other than variance in the derivation leads to sparse representation different from KPCA. Sparse greedy matrix approximation (SGMA) [5] is considered to find a low rank matrix approximation of the Gram matrix such that the difference between the Gram matrix and approximated matrix is minimized in the case of the Frobenius norm. The approximated matrix is obtained by finding a subset of training data set and the projection matrix. SGMA is considered as sparse training algorithms [13]. In [12], the authors pointed out that SGMA in fact amounts to sparse KPCA, and introduced to use a matching pursuit algorithm for KPCA. They showed a proof that this algorithm only a subset of training set is enough to reconstruct the Gram matrix, where the subset of training data is referred to as the compression set [14]. However, this algorithm is not applicable to large data set. In [6], the authors introduced two schemes of reduced set selection to reduce space which spanned by all the mapped data. The first scheme is to consider removing some mapped data from the expansion while incurring minimum approximation error. The second scheme is to use an l_1-norm on approximated expansion coefficient. Nevertheless, both have high computational cost. Tipping [7] proposed a sparsification method for KPCA, this approach is motived by probabilistic PCA [15], which consists of applying a maximum-likelihood technique to approximate the transformed covariance matrix into a feature space to obtain the sparse form. However, such method depends on a probabilistic model for the data. Vollgraf et al. [9] considered the square of the ratio of the l_1- and the l_2-norm of the coefficient vector as the regularization term. They used the so-called hyper-ellipsoidal conjugate gradient method to minimize the cost function. However, the proposed algorithm includes heuristic, and the solution relies on some small positive value to shrink it toward zero. In [11], the authors proposed the method which eliminated irrelevant training data by using the square distance metric of training data in the feature space. However, it mainly focused on selection of samples to produce sparse representation and did not consider the MSE. Another extension of KPCA is adaptive online learning. There have been proposed several online algorithms that determine whether or not an input sample is included in a dictionary, yielding a sparse representation (see [16], [17], for instance).

978-1-5090-0620-5/16/$31.00

In summary, there have been little attention to simultaneously evaluating the approximation property as well as the sparsity of coefficients as of yet. For sparsification of a linear PCA (SPCA), this problem has been addressed by Zou et al. [18]. The sparse solution can be obtained by relaxing the MSE into a regression-type problem and adding the l_1-norm on the coefficients. This paper extend the concept of the SPCA to sparsification of the KPCA. In the proposed approach, the KPCA is also reformulated into a regression-type problem, with a quadratic penalty; l_1-norm can be integrated into the regression criterion, leading to a new cost function. In order to obtain sparse solution, we develop an alternating quadratic optimization to minimize the proposed cost function, which involves an iterative minimization by alternating direction method of multipliers (ADMM) [19].

This paper is organized as follows: in Section II, we briefly review the PCA, KPCA, and SPCA. Section III develops the proposed sparse KPCA (SKPCA) algorithm and Section IV gives the experimental results. Finally, Section V concludes the proposed work.

II. PCA KPCA AND SPARSE PCA

In this section we briefly review the existing methods. We begin with PCA to introduce relevant theory.

A. PCA

Let $\{\mathbf{x}_i\}_{i=1}^N$ be a set of N input data vectors in a d-dimensional space, which are centered, with no loss of generality, at the origin, $\sum_{i=1}^{N} \mathbf{x}_i = 0$. We suppose that $\mathcal{U}$ is the M-dimensional subspace spanned by $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M$ $(M < d)$. The projection of $\mathbf{x}_i$ on $\mathcal{U}$ can be written as $U^T\mathbf{x}_i$, where $U = [\mathbf{u}_1, \ldots, \mathbf{u}_M]$. The underlying idea behind PCA is to maximize variance of the projected data. For simplicity we assume $M = 1$ and $\mathbf{u}_1^T\mathbf{u}_1 = 1$. We maximize the projected variance $\mathbf{u}_1^T S\mathbf{u}_1$ with respect to $\mathbf{u}_1$ (normalized to unit norm), where $S = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\mathbf{x}_i^T = \frac{1}{N}X^TX$ and $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$. By introducing the Lagrange multiplier λ_1 and taking the derivative, we obtain $S\mathbf{u}_1 = \lambda_1\mathbf{u}_1$, which means that $\mathbf{u}_1$ is eigenvector of S corresponding to eigenvalue λ_1. It provides a way of finding successive basis for the M-dimensional subspace under orthogonal constraints. Since S is positive matrix, there exists M orthogonal eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_M$ of S corresponding to the M largest eigenvalues $\lambda_1, \ldots, \lambda_M$, these eigenvectors form the basis of the M-dimensional subspace. For data $\mathbf{x}$, inner product $\mathbf{x}^T\mathbf{u}_k$ is called the kth principal component (PC) of the data set.

B. KPCA

Consider a function $\phi(\mathbf{x})$ mapping each data $\mathbf{x}_i$ into the feature space $\phi(\mathbf{x}_i)$ (assumed centered). The first step is to perform PCA in feature space. The covariance matrix of the mapped data C is defined by

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

Its eigenvalue λ_k and eigenvector $\mathbf{v}_k$ that satisfies

$$C\mathbf{v}_k = \lambda_k\mathbf{v}_k \quad k = 1, \ldots, M. \tag{2}$$

As $C\mathbf{v}_k = \frac{1}{N}\sum_{i=1}^{N} \phi(\mathbf{x}_i)(\phi(\mathbf{x}_i)^T\mathbf{v}_k)$, all solutions $\mathbf{v}_k$ must lie in the span of $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)$. Thus, $\mathbf{v}_k$ is the superposition of $\{\phi(\mathbf{x}_i)\}_{i=1}^N$:

$$\mathbf{v}_k = \sum_{i=1}^{N} a_{ik}\phi(\mathbf{x}_i) \quad k = 1, \ldots, M. \tag{3}$$

In order to extract nonlinear PCs, we compute the projections of test vector $\phi(\mathbf{x})$ onto the eigenvectors $\mathbf{v}_k$.

$$\phi(\mathbf{x})^T\mathbf{v}_k = \sum_{i=1}^{N} a_{ik}k(\mathbf{x}_i, \mathbf{x}). \tag{4}$$

Here we use kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$.

C. Sparse PCA

From the KPCA described in Section II-B, KPCA also suffers from the fact that $\mathbf{v}_k$ is a linear combination of all these mappings $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)$ as given in (3), thus it is difficult to interpret PCs when N is large enough. In order to interpret PCs easily, Zou et al. [18] proposed the sparse principal component analysis to reduce the number of used variables. First, formulate PCA as the minimization of the criterion of MSE:

$$\arg\min_{U} \sum_{i=1}^{N} \|\mathbf{x}_i - UU^T\mathbf{x}_i\|^2 \text{ s.t. } U^TU = I_{M\times M}, \tag{5}$$

where $U = [\mathbf{u}_1, \ldots, \mathbf{u}_M]$ is rank M matrix and $I_{M\times M}$ is identity matrix. This criterion can be relaxed into the following:

$$\arg\min_{U,B} \sum_{i=1}^{N} \|\mathbf{x}_i - UB^T\mathbf{x}_i\|^2 + \lambda\sum_{k=1}^{M} \|\mathbf{b}_k\|^2 \text{ s.t. } U^TU = I_{M\times M}, \tag{6}$$

where $B = [\mathbf{b}_1, \ldots, \mathbf{b}_M]$ is rank M matrix and $\lambda > 0$ is constant ensuring the reconstruction of principal components. It is ensured that we can soften criterion (5) as (6) to deal with PCA. Next, the first M sparse PCs can be solved by the following optimization problem:

$$\arg\min_{\mathbf{u}_k, \mathbf{b}_k} \sum_{k=1}^{M} \|X\mathbf{u}_k - X\mathbf{b}_k\|^2 + \lambda\sum_{k=1}^{M} \|\mathbf{b}_k\|^2 + \sum_{k=1}^{M} \lambda_{1,k}\|\mathbf{b}_k\|_1 \text{ s.t. } U^TU = I_{M\times M}. \tag{7}$$

where $\lambda_{1,k}$ is constant determining the trade-off between variance and sparsity.

There are two methods to solve the optimization problem (7). The first method is introduced in [18], by alternating optimization over U and B in two stages until convergence. Another is introduced in [21], by optimizing $\mathbf{u}_k$ and $\mathbf{b}_k$ sequentially. We now introduce the method of [21]. First,

assume $\mathbf{u}_k^T\mathbf{u}_k = 1$ and fix $\mathbf{u}_k$ and X, then $\mathbf{b}_k$ is the solution of

$$\arg\min_{\mathbf{b}_k} \|X\mathbf{u}_k - X\mathbf{b}_k\|^2 + \lambda\|\mathbf{b}_k\|^2 + \lambda_{1,k}\|\mathbf{b}_k\|_1.$$

For fixed $\mathbf{b}_k$, the optimal $\mathbf{u}_k$ is solved by the following criterion.

$$\arg\min_{\mathbf{u}_k} \|X\mathbf{u}_k - X\mathbf{b}_k\|^2 \quad \text{s.t. } \mathbf{u}_k^T\mathbf{u}_k = 1,\ \mathbf{u}_k^T U_{(k-1)} = \mathbf{0},$$

where $U_{(k-1)}$ is $d \times (k-1)$ matrix consisting of the previous $(k-1)$ solutions $\mathbf{u}_k$ and $U_{(k-1)}^T U_{(k-1)} = I$. The solution of $\mathbf{u}_k$ is given by $\mathbf{u}_k = \frac{s}{\sqrt{s^T s}}$, where $s = (I - U_{(k-1)}U_{(k-1)}^T)X^T X\mathbf{b}_k$. The authors of [21] refer to this algorithm as sequential approach, compare to simultaneous algorithm proposed by Zou et al. The difference between sequential algorithm and simultaneous algorithm is that when evaluate the $(k+1)$th PC, the former yields exactly the same as the first k components, while the latter gives different results for all PCs. In the next section, we will show that KPCA can be recast into a regression-type problem and harness the l_1-norm on coefficients to yield sparse solution.

III. Sparse KPCA with elastic net regularization

In this section, we introduce our proposed method. Denote $\Phi = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]$, $V = [\mathbf{v}_1, \ldots, \mathbf{v}_M]$, $A = [\mathbf{a}_1, \ldots, \mathbf{a}_M]$, $\mathbf{a}_k = (a_{k1}, \ldots, a_{kN})^T$, and Gram matrix $K = \Phi^T\Phi$, whose element is $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Then, we have matrix form $V = \Phi A$ according to (3). For the sake of simplicity, data are assumed centered. If data are uncentered, we need to use centered Gram matrix $\bar{K} = (I - 1_N J_N^T)K(I - J_N 1_N^T)$, where 1_N is N-dimensional column vector of all ones, J_N is N-dimensional column vector whose elements are all $\frac{1}{N}$ and I is the $N \times N$ identity matrix. $\bar{K}$ is called a conditionally positive definite matrix [13]. Note that $\bar{K}$ can be rank deficient due to centering of K, if $\bar{K}$ is singular, we use the pseudoinverse of $\bar{K}$ [6]. KPCA minimizes the MSE in feature space:

$$J[V] = \sum_{i=1}^{N} \|\phi(\mathbf{x}_i) - VV^T\phi(\mathbf{x}_i)\|^2$$
$$\text{s.t. } V^T V = A^T K A = I_{M\times M}.$$

In the same way as [22], the minimization of $J[V]$ can be modified into the minimization of

$$J_1[A] = \sum_{i=1}^{N} \|\mathbf{y}_i - QQ^T\mathbf{y}_i\|^2 \quad \text{s.t. } Q^T Q = I_{M\times M} \tag{8}$$

where $Q = [\mathbf{q}_1, \ldots, \mathbf{q}_M] = K^{\frac{1}{2}}A$ or

$$\mathbf{q}_k = K^{\frac{1}{2}}\mathbf{a}_k. \tag{9}$$

Note that $[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] = K^{\frac{1}{2}}$, which means that $\mathbf{y}_i$ is the ith column vector of matrix $K^{\frac{1}{2}}$.

So we can soften (8) as a regression-type optimization problem and add regularization to obtain sparse solution. In the same way as done in (6), we relax (8) as follows:

$$\sum_{i=1}^{N} \|\mathbf{y}_i - PQ^T\mathbf{y}_i\|^2 + \frac{\lambda}{2}\sum_{k=1}^{M}\|\mathbf{q}_k\|^2$$
$$= \sum_{k=1}^{M} \|K^{\frac{1}{2}}\mathbf{p}_k - K^{\frac{1}{2}}\mathbf{q}_k\|^2 + \frac{\lambda}{2}\sum_{k=1}^{M}\|\mathbf{q}_k\|^2 \tag{10}$$

subject to $P^T P = I_{M\times M}$, where $P = [\mathbf{p}_1, \ldots, \mathbf{p}_M]$ and $\lambda > 0$ is constant ensuring the reconstruction of principal components. If we set $P = Q$, then it is equivalent to $J_1[A]$. To promote sparsity, we add the l_1-norm regularization term into the (10) and minimize the cost function:

$$J[P, Q] = \frac{1}{2}\sum_{k=1}^{M} \|K^{\frac{1}{2}}\mathbf{p}_k - K^{\frac{1}{2}}\mathbf{q}_k\|^2 + \frac{\lambda}{2}\sum_{k=1}^{M}\|\mathbf{q}_k\|^2 + \sum_{k=1}^{M}\lambda_{1,k}\|\mathbf{a}_k\|_1$$

subject to $P^T P = I_{M\times M}$ or consider to optimize them sequentially

$$J[\mathbf{p}_k, \mathbf{q}_k] = \frac{1}{2}\|K^{\frac{1}{2}}\mathbf{p}_k - K^{\frac{1}{2}}\mathbf{q}_k\|^2 + \frac{\lambda}{2}\|\mathbf{q}_k\|^2 + \lambda_{1,k}\|\mathbf{a}_k\|_1 \quad \text{s.t. } \mathbf{p}_k^T\mathbf{p}_k = 1 \tag{11}$$

where $\lambda_{1,k}$ is constant determining the trade-off between variance and sparsity, the factor of $1/2$ is included for convenience and $\mathbf{a}_k$ is referred to as a coefficient vector since its elements are used as the coefficients of the linear combination in (3). Note that $\mathbf{a}_k$ satisfies (9). Cost function (11) is quadratic with respect to either $\mathbf{p}_k$ or $\mathbf{q}_k$, which is minimized by alternating minimization.

For $\mathbf{p}_k$, the first term is only considered. $\mathbf{p}_k$ can be solved using the same scheme as $\mathbf{u}_k$ for SPCA shown in Section II-C. For $\mathbf{q}_k$, we cannot apply the same scenario as SPCA described in Section II-C, since the l_1-norm includes a matrix $K^{-\frac{1}{2}}$, which is generally not diagonal. For this case, the ADMM [19] can be applied to (9) and (11) with the augmented Lagrangian:

$$L_\rho(\mathbf{q}_k, \mathbf{a}_k, \mathbf{t}_k) = \frac{1}{2}\|K^{\frac{1}{2}}\mathbf{p}_k - K^{\frac{1}{2}}\mathbf{q}_k\|^2 + \frac{\lambda}{2}\|\mathbf{q}_k\|^2 + \lambda_{1,k}\|\mathbf{a}_k\|_1 + \mathbf{t}_k^T(K^{-\frac{1}{2}}\mathbf{q}_k - \mathbf{a}_k) + \frac{\rho}{2}\|K^{-\frac{1}{2}}\mathbf{q}_k - \mathbf{a}_k\|^2.$$

ADMM consists of the iterations

$$\mathbf{q}_k^{j+1} = \arg\min_{\mathbf{q}_k} L_\rho(\mathbf{q}_k, \mathbf{a}_k^j, \mathbf{t}_k^j)$$
$$\mathbf{a}_k^{j+1} = \arg\min_{\mathbf{a}_k} L_\rho(\mathbf{q}_k^{j+1}, \mathbf{a}_k, \mathbf{t}_k^j)$$
$$\mathbf{t}_k^{j+1} = \arg\min_{\mathbf{t}_k} L_\rho(\mathbf{q}_k^{j+1}, \mathbf{a}_k^{j+1}, \mathbf{t}_k)$$

where $\rho > 0$, the superscript j is the iteration counter. Since the algorithm utilizes ADMM, we consider criteria based on the primal and dual residuals for ADMM. The primal and dual residual at iteration $(j+1)$ are $e_p^{j+1} = K^{-\frac{1}{2}}\mathbf{q}_k^{j+1} -$

Algorithm 1 Algorithm of the sparse KPCA

1: Input: Gram matrix K, orthogonal matrix P, # of PCs M
2: **for** $k = 1$ **to** M **do**
3: **while** iterations are not converge or a maximum number of iterations is not exceed **do**
4: Initialize $\mathbf{q}_k, \mathbf{a}_k, \mathbf{t}_k$
5: **while** ADMM has not converged **do**
6: $\mathbf{q}_k \leftarrow (K+\lambda I+\rho K^{-1})^{-1}[K\mathbf{p}_k+\rho K^{-\frac{1}{2}}(\mathbf{a}_k-\frac{\mathbf{t}_k}{\rho})]$
7: $\mathbf{a}_k \leftarrow S_{\frac{\lambda_{1,k}}{\rho}}(K^{-\frac{1}{2}}\mathbf{q}_k + \frac{\mathbf{t}_k}{\rho})$
8: $\mathbf{t}_k \leftarrow \mathbf{t}_k + \rho(K^{-\frac{1}{2}}\mathbf{q}_k - \mathbf{a}_k)$
9: **end while**
10: $\mathbf{q}_k \leftarrow \frac{\mathbf{q}_k}{\sqrt{\mathbf{q}_k^T\mathbf{q}_k}}$
11: $\mathbf{p}_k = (I - P_{(k-1)}P_{(k-1)}^T)K\mathbf{q}_k$
12: $\mathbf{p}_k \leftarrow \frac{\mathbf{p}_k}{\sqrt{\mathbf{p}_k^T\mathbf{p}_k}}$
13: **end while**
14: **end for**
15: Output the coefficient $A = [\mathbf{a}_1, \ldots, \mathbf{a}_M]$

$\mathbf{a}_k^{j+1}$, $e_d^{j+1} = -\rho K^{-\frac{1}{2}}(\mathbf{a}_k^{j+1} - \mathbf{a}_k^j)$, we terminate the ADMM algorithm when

$$\|e_p^{j+1}\| \le \sqrt{N}\epsilon^{abs} + \epsilon^{rel}\max\{\|K^{-\frac{1}{2}}\mathbf{q}_k^{j+1}\|, \|\mathbf{a}_k^{j+1}\|\}$$
$$\|e_d^{j+1}\| \le \sqrt{N}\epsilon^{abs} + \epsilon^{rel}\|K^{-\frac{1}{2}}\mathbf{t}_k^{j+1}\|$$

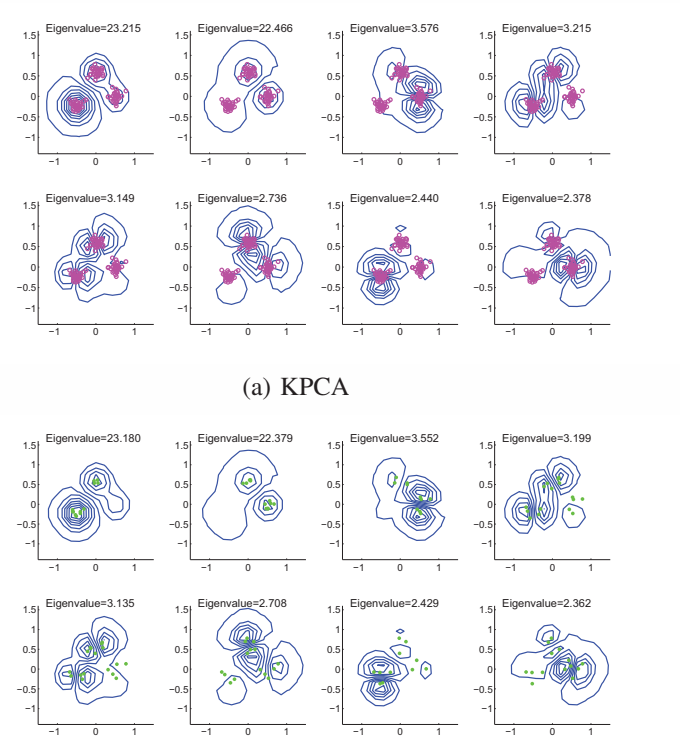
where $\epsilon^{abs} > 0$ is an absolute tolerance and $\epsilon^{rel} > 0$ is a relative tolerance [19]. The detailed algorithm is summarized in Algorithm 1 including the explicit updating formula. In Steps 7 and 11 of the algorithm, the $S_\tau(\alpha)$ is soft-threshold operator defined as $S_\tau(\alpha) = \text{sign}(\alpha)\max\{|\alpha| - \tau, 0\}$, and $P_{(k-1)}$ is defined as the submatrix that consists of the previous $(k-1)$ solutions $\mathbf{p}_k$, respectively.

IV. Experiments

Experimental results are presented to illustrate the proposed method. We use Gaussian kernels of the form $k(x, y) = \exp(-\|x-y\|^2/(2\sigma^2))$ to run the algorithm on both toy data and real world data.

A. Toy Examples

Consider a data set from three clusters at $(-0.5, -0.2)$, $(0, 0.6)$, $(0.5, 0)$. We generate 30 points for each cluster with standard deviation 0.1. The KPCA and the proposed SKPCA ($\sigma^2 = 0.05$) are applied to it. We set parameters $\lambda_{1,8} = (0.02, 0.02, 0.009, 0.008, 0.006, 0.007, 0.01, 0.01)$, $\lambda = 0.1$, $\rho = 0.01$,$\epsilon^{abs} = 10^{-4}$, $\epsilon^{rel} = 10^{-4}$. We consider that the convergence is achieved if the squared norm of $\mathbf{q}_k^{new} - \mathbf{q}_k^{old}$ is less than ϵ, where $\mathbf{q}_k^{new}$ and $\mathbf{q}_k^{old}$ denote the $\mathbf{q}_k$ after and before the update in the loop between Steps 4 and 13 in Algorithm 1. During the process of iteration, if the squared norm is not smaller than ϵ, the number of iterations maybe exceeds the predefined maximal number of iterations, and iteration is over. We set $\epsilon = 10^{-2}$, The maximum iteration numbers of ADMM and a whole loop are 300 and 300, respectively.



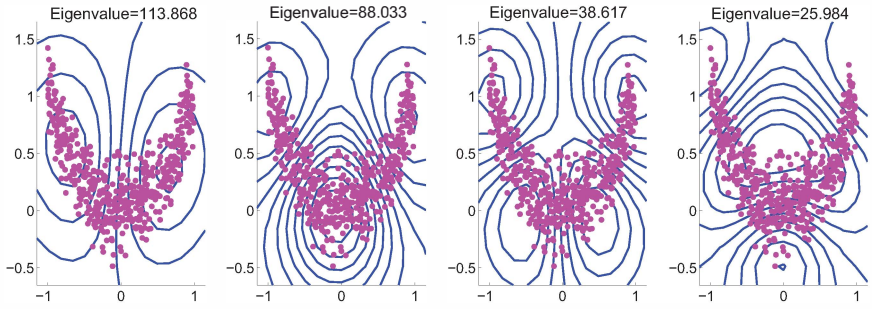
(a) KPCA

(b) Sparse KPCA obtained by proposed method

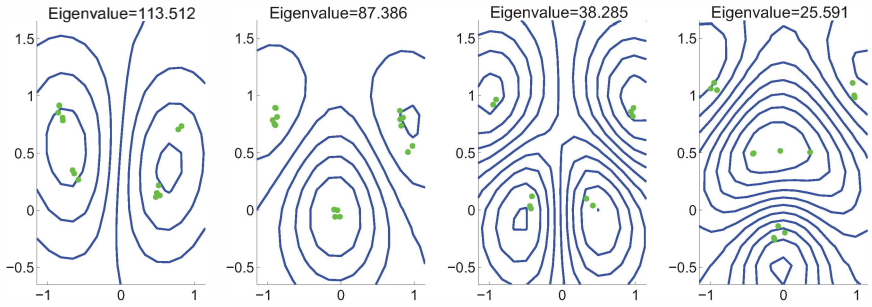
Fig. 1. Contour plots of first eight feature extractors of KPCA and the proposed SKPCA for a data set of 90 samples respectively. The magenta points denote data, and the green points denote the corresponding coefficients being nonzero.

Fig. 1 shows the KPCA and the proposed SKPCA. The contours are lines along which the projection onto the corresponding principal component, defined by (4) is constant. The green points represent corresponding coefficients being nonzero. It is seen that the nonzero coefficients approximation used for feature extraction are close resemblance to KPCA.

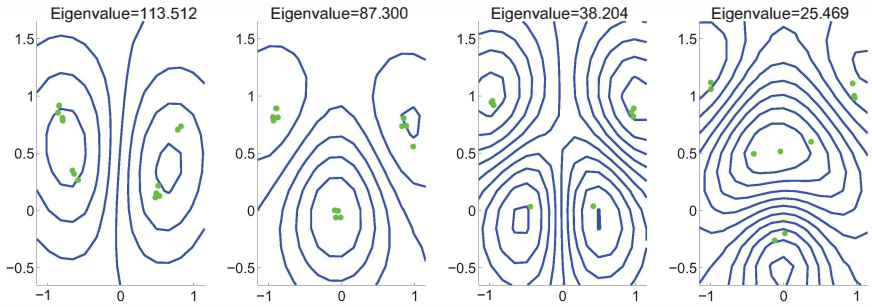
In the second example, the $N = 500$ two dimensional data points (x, y) are generated, where x-values have uniform distribution in $[-1, 1]$, y-values are generated from $y_i = x_i^2 + \xi$, where ξ is normal noise with standard deviation 0.2. We set $\sigma = 0.5$, $\lambda = 0.05$, $\lambda_{1,4} = 0.05$, $\rho = 0.09$, $\epsilon^{rel} = 10^{-4}$, $\epsilon^{abs} = 10^{-4}$, and $\epsilon = 10^{-2}$ to extract the first four PCs. The maximum iteration numbers of ADMM and a whole loop are 300 and 60, respectively. To investigate the influence of regularization parameter in the overall results, we set $\lambda_{1,4} = (0.05, 0.07, 0.08, 0.09)$ for the same data again. The experiment results are given in Fig. 2. Contour lines of the first four PCs are obtained by KPCA and proposed SKPCA. It is obvious that the proposed method with small part of data captures the same structure as the KPCA. Meanwhile, the sparsity of the proposed method explained by the four PCs, depends on the choice of the parameter $\lambda_{1,k}$ and ρ. The parameters should be set to the appropriate value that converges to the optimal results.



(a) KPCA



(b) Proposed method with $\lambda_{1,4} = 0.05$, $\rho = 0.09$



(c) Proposed method with $\lambda_{1,4} = (0.05, 0.07, 0.08, 0.09)$, $\rho = 0.09$

Fig. 2. The first four PCs obtained through KPCA and the proposed SKPCA from 500 data. Data are given by magenta points, while points achieving by proposed method are illustrated by green points.

B. MNIST Data

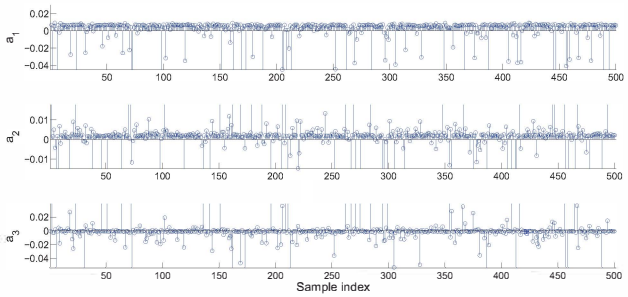
We illustrate our approach on real world data, with the MNIST database of handwritten digits [24]. This dataset consists of the handwritten digits, form 0 to 9, given in 28-by-28 pixels. We studied sparseness, where 500 samples were randomly chosen from the training set of 60,000 examples and remained the same in experiment, and a test set of 10,000 examples were used. The data without any processing or normalization.

Again, we set parameter $\rho = 0.01$, $\sigma = 700$, $\epsilon^{abs} = 10^{-2}$, $\epsilon^{rel} = 10^{-4}$, $\lambda = 0.001$, $\lambda_{1,3} = (0.002, 0.002, 0.004)$, $\epsilon = 10^{-6}$. The maximum iteration numbers of ADMM and a whole loop are 300 and 30 respectively.

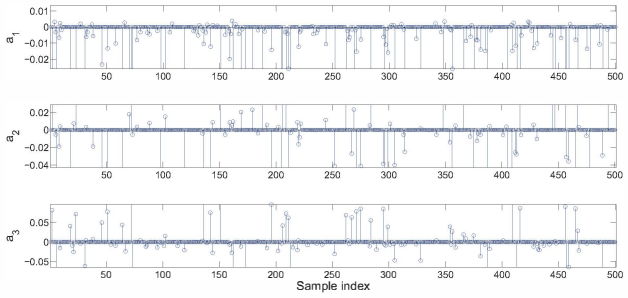
The resulting coefficients are shown in Fig. 3, which exhibits coefficient vector $\mathbf{a}_k$ obtained by the KPCA and the proposed SKPCA in (3) for three principal components. It is especially apparent in the case of sparseness where most of the coefficients are zero. On the other hand, for evaluation measurement, we compute MSE for all training samples, and compare it to the optimal solution of $J[V]$. The result is shown in Fig. 4. We see that the sparse coefficient vector yielded by the proposed method is slightly higher than optimal case, which is equivalent to optimal solution of $J[V]$.

V. Conclusion

We have proposed a novel approach for SKPCA, where the elastic net regularization is utilized for MSE function. An alternating minimization algorithm including ADMM has been developed. We presented experimental results to support the proposed approach that can promote the sparsity as well



(a) Original coefficient



(b) Sparseness coefficient

Fig. 3. Comparison between original coefficient and spareness coefficient.

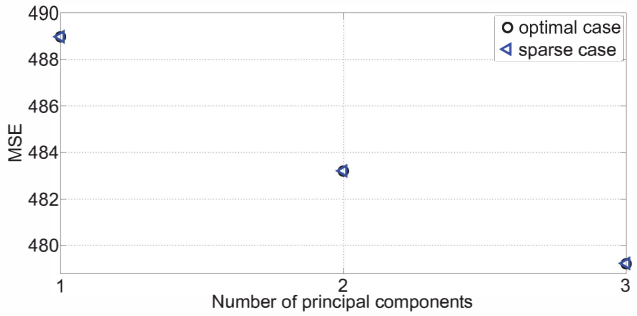


Fig. 4. MSE for KPCA and the proposed SKPCA as a function of the number of principal components.

as minimizing the MSE, which is a characteristic of PCA. Future work may include application to practical data and the theoretical evaluation of the convergence.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grant-in-Aid (B) Number 26280054.

REFERENCES

[1] I. Jolliffe, *Principal Component Analysis.* Springer-Verlag, 1986.
[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[3] B. Schölkopf, A. Smola, and K. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, July 1998.
[4] A. Smola, O. Mangasarian, and B. Schölkopf, "Sparse kernel feature analysis," Tech. Rep., 1999.
[5] A. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA: Morgan Kaufmann, 2000, pp. 911–918.
[6] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K. Müller, G. Rätsch, and A. Smola, "Input space versus feature space in kernel-based methods," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1000–1017, Sep 1999.
[7] M. E. Tipping, "Sparse kernel principal component analysis," in *Advances in Neural Information Processing Systems 13*. MIT Press, 2001, pp. 633–639.
[8] X. Jiang, Y. Motai, R. Snapp, and X. Zhu, "Accelerated kernel feature analysis," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, 2006, pp. 109–116. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.43
[9] R. Vollgraf and K. Obermayer, "Sparse optimization for second order kernel methods," in *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, 2006, pp. 145–152.
[10] Z. K. Gon, J. Feng, and C. Fyfe, "A comparison of sparse kernel principal component analysis methods," in *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*, vol. 1, 2000, pp. 309–312.
[11] Y. Xu, D. Zhang, J. Yang, Z. Jin, and J. Y. Yang, "Evaluate dissimilarity of samples in feature space for improving kpca," *International Journal of Information Technology & Decision Making*, vol. 10, no. 03, pp. 479–495, 2011.
[12] Z. Hussain and J. Shawe-Taylor, "Theory of matching pursuit," in *Advances in Neural Information Processing Systems 21*. Curran Associates, Inc., 2009, pp. 721–728.
[13] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Cambridge, MA, USA: MIT Press, 2001.
[14] Z. Hussain, J. Shawe-Taylor, D. Hardoon, and C. Dhanjal, "Design and generalization analysis of orthogonal matching pursuit algorithms," *Information Theory, IEEE Transactions on*, vol. 57, no. 8, pp. 5326–5341, Aug 2011.
[15] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, vol. 61, pp. 611–622, 1999.
[16] Y. Washizawa, "Adaptive subset kernel principal component analysis for time-varying patterns," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 12, pp. 1961–1973, Dec 2012.
[17] T. Tanaka, "Dictionary-based online kernel principal subspace analysis with double orthogonality preservation," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, April 2015, pp. 4045–4049.
[18] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.
[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
[20] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005.
[21] K. Sjöstrand, L. Clemmensen, R. Larsen, and B. Ersbøll, "Spasm: A matlab toolbox for sparse statistical modeling," *Journal of Statistical Software*, 2012.
[22] T. Tanaka, Y. Washizawa, and A. Kuh, "Adaptive kernel principal components tracking," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 1905–1908.
[23] P. Honeine, "Online kernel principal component analysis: A reduced-order model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1814–1826, 2012. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.270
[24] Y. LeCun and C. Cortes. (1998) MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. [Online]. Available: http://yann.lecun.com/exdb/mnist/