Kernel Principal Component Analysis Allowing Sparse Representation and Sample Selection

Duo Wang[1] and **Toshihisa Tanaka**[2]

ABSTRACT

Kernel principal component analysis (KPCA) is a kernelized version of principal component analysis (PCA). A kernel principal component is a superposition of kernel functions. Since the number of kernel functions equals the number of samples, each component is not a sparse representation. Our purpose is to sparsify coefficients expressing in linear combination of kernel functions. Two types of sparse kernel principal component are proposed in this paper. The method for solving the sparse problem is comprised of two steps: (a) we start with the Pythagorean theorem and derive an explicit regression expression of KPCA, and (b) two types of regularization, l_1-norm or $l_{2,1}$-norm, are added into the regression expression in order to obtain two different sparsity forms, respectively. As the proposed objective function is different from elastic net-based sparse PCA (SPCA), the SPCA method cannot be directly applied to the proposed cost function. We show that the sparse representations are obtained using iterative optimization by conducting an alternating direction method of multipliers. Experiments on toy examples and real data confirm the performance and effectiveness of the proposed method.

Keywords: Principal Component Analysis, Sparse Principal Component Analysis, Kernel Principal Component Analysis, Alternating Direction Method of Multipliers

1 INTRODUCTION

Principal component analysis (PCA) [1] is a widely-used linear method for dimensionality reduction. It has been used in many engineering fields, including signal processing, image processing, statistical analysis, data compression, and pattern recognition.

Two main perspectives are usually used to characterize PCA [2]. The first viewpoint is the maximum variance derivation. The idea here is to determine a lower-dimensional linear subspace that captures the largest variance of the data under the orthonormal constraint. Another explanation is the mean squared error (MSE) derivation that minimizes the average distance between the data samples and their projection under the low rank matrix with orthonormal columns. These two different derivations result in the eigendecomposition of the covariance matrix of the data. This yields the subspace spanned by the eigenvectors of the covariance matrix corresponding to the largest eigenvalues. However, PCA cannot discover nonlinear data structures because of its linear limitation. For such a nonlinear structure, the kernelized version of principal component analysis (KPCA) [3] successfully copes with this difficulty. In KPCA, all data is embedded into a reproducing kernel Hilbert space (RKHS) [4] by means of a nonlinear map. To get rid of the nonlinear map, the kernel trick is applied to execute PCA in RKHS.

KPCA, however, encounters a problem that the principal components (PCs) involve all of the data samples, which may lead to overfitting and memory overflow. It is thus necessary to reduce the number of samples used for PCs while keeping the original quality of the standard KPCA.

The sparsification of KPCA proposed so far can be categorized into two types [5–13]. The first type follows the idea of a reduced subset of the dataset. For example, Tipping [8] solved this problem by approximating the covariance matrix in feature space. This idea comes from probabilistic PCA [14] by means of a maximum-likelihood estimation to obtain the sparse form of the covariance matrix. This kind of method is not aimed to generate sparse representation for PCs, however. "Sparse greedy matrix approximation" (SGMA) [6] has been proposed to construct a compressed matrix approximation of the kernel matrix in order to minimize the Frobenius norm of the residual between these two matrices. The reconstructed matrix is achieved by seeking a subset of the dataset and the projection matrix. SGMA can be thought of as a form of sparse training algorithm [15]. Further, Hussain and Shawe-Taylor [13] stated that SGMA aims to yield a sparse KPCA, and proved its equivalence from a matching pursuit perspective. They followed two steps—quotient maximization and deflation—to achieve compression set indices from the data set and without the need of a projection matrix. The compressed kernel matrix can be computed

Manuscript received on May 5, 2019 ; revised on May 16, 2019.

Final manuscript received on May 16, 2019.

[1,2] The authors are with the Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan, E-mail: wangduo@sip.tuat.ac.jp and tanakat@cc.tuat.ac.jp

by the compression set [16]. Note that feature extractors as expansions in terms of mapped samples, Schölkopf et al. [7] considered reduced-set methods to drop unimportant mapped data. One selection is to eliminate unimportant mapped data from the expansion while allowing for an error caused by this elimination. Another selection is to enforce an l_1-norm on an approximated expansion coefficient. Nevertheless, both selections have high computational costs. Following a similar idea, Xu et al. [12] proposed a method that selects the dissimilarity among all of the mapped data under the squared distance measurement criterion, and then used identified subset data and training data to perform KPCA. This type of sparsification mainly focuses on evaluating dissimilarities of samples while ignoring the MSE.

The second type aims to find a sparse approximation to eigenvectors in feature space. Smola et al. [5] applied an l_1-constraint on coefficients to sparse components, named sparse kernel feature analysis (SKFA). Based on SKFA, Jiang et al. [9] proposed accelerated kernel feature analysis (AKFA) to extract features, which is superior to the time complexity of SKFA. However, both method has the problem that the first component only using one training sample, leading to bad interpretation of the PC. Vollgraf et al. [10] introduced the regularization term consisting of the square of the ratio of the l_1- and the l_2-norm of the coefficient vector. The cost function was minimized by the so-called hyper-ellipsoidal conjugate gradient descent method (HECGD). This algorithm includes heuristics, however, and the solution depends on a small positive value to shrink it toward zero. Suykens et al. [17] formulated KPCA for use with least squares support vector machines (LS-SVM). The links between KPCA and LS-SVM are established through the primal and dual problem. Alzate et al. [18] extended KPCA to a generalized form of kernel component analysis (KCA) with a general underlying loss function and proposed two algorithms to sparsify KPCA. The first algorithm introduces an epsilon-insensitive zone [19] into the loss function and the sparseness can be obtained by epsilon value. The second one considers a loss function of the weighted form [20]. Sparseness is obtained by computing the weight when the value is equal to zero. However, these two algorithms need to observe the distribution of the score values to decide the epsilon value, which uses a heuristic. The preliminary work of this paper falls in this category [21]. Recently, a new sparse KPCA via sequential method (SSKPCA) has been proposed [22].

Several online adaptive extensions have been introduced to KPCA [23, 24]. The main concept is to consider whether the incoming data sample should be added into a dictionary or not, thus leading to a sparse representation. However, so far little attention has been paid to simultaneously establishing a connection between the approximation property and the sparsity of coefficients that can cope with the drawbacks of the two types of sparsification previously mentioned. In this paper, we propose a novel sparsification of KPCA that evaluates the MSE criterion while promoting sparsity in the representation of PCs. We show that in the case of KPCA, MSE criterion can be turned into a regression model. The sparse coefficients can be obtained by imposing either l_1-norm or $l_{2,1}$-norm onto the regression model. This can be regarded as an extension of the elastic net [25] regularization of the sparse linear PCA was proposed by Zou et al. [26]. Since the proposed cost function is different from sparse linear PCA, the SPCA method cannot be directly applied to the proposed cost function. To this end, we use an alternating direction method of multipliers (ADMM) [27] method in its iterative optimization to yield sparse solutions.

In the following sections, we summarize PCA, KPCA, and elastic net-based sparse PCA in Section 2. The main focus of the paper, two types of sparsity algorithms of KPCA, are described in Section 3. Experiments are provided in Section 4, and the conclusion of the paper is given in Section 5.

2 PRINCIPAL COMPONENT ANALYSIS, ITS KERNELIZATION, AND SPARSIFICATION: REVIEW

2.1 Principal Component Analysis

Denote a dataset $\{\mathbf{x}_i\}_{i=1}^N$ belong to $\mathbb{R}^d$ with zero mean. Suppose the set of vectors $\{\mathbf{u}_i\}_{i=1}^M$ is a basis for space $\mathcal{U}$, where $\mathcal{U}$ is a subspace of $\mathbb{R}^d$, that is $M < d$. We project $\mathbf{x}_i$ onto subspace $\mathcal{U}$ to obtain $U^T\mathbf{x}_i$, where $U = [\mathbf{u}_1, \ldots, \mathbf{u}_M]$. The maximum variance derivation is to capture maximum variance of the projected data. To this end, we define: $S = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T = \frac{1}{N}X^TX$ and $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$. We maximize the projected variance $\sum_{i=1}^{M}\mathbf{u}_i^TS\mathbf{u}_i$ under the constraint that $U^TU = I_{M\times M}$, where $I_{M\times M}$ is the identity matrix. The constrained maximization of the variance can be performed by solving the eigenvalue problem for S:

$$S\mathbf{u}_i = \lambda_i\mathbf{u}_i$$

where λ_i is an eigenvalue of S indexed such that $\lambda_1 \geq \ldots \geq \lambda_M \geq 0$, and $\mathbf{u}_i$ is an eigenvector corresponding to λ_i. For any sample $\mathbf{x}$, the kth principal component (PC) is $\mathbf{u}_k^T\mathbf{x}$.

2.2 Kernel Principal Component Analysis

For nonlinear data, we employ a map ϕ to embed the data into an RKHS, that is $\mathbf{x}_i \to \phi(\mathbf{x}_i)$, which is considered to be an element of an RKHS. For the sake of simplicity, the dataset $\{\phi(\mathbf{x}_i)\}_{i=1}^N$ is assumed to be

centered. The covariance matrix C in the feature space can be expressed as:

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

The eigenvalue λ_k and eigenvector $\mathbf{v}_k$ of C satisfy:

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

From $C\mathbf{v}_k = \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i)(\phi(\mathbf{x}_i)^T\mathbf{v}_k) = \lambda_k\mathbf{v}_k$, we observe that eigenvector $\mathbf{v}_k$ belongs to the set of mapped data $\{\phi(\mathbf{x}_i)\}_{i=1}^N$. In other words, eigenvector $\mathbf{v}_k$ is the linear combinations of $\{\phi(\mathbf{x}_i)\}_{i=1}^N$:

$$\mathbf{v}_k = \sum_{i=1}^{N} a_{ik}\phi(\mathbf{x}_i) \quad (3)$$

where a_{ik} are the linear combinaton coefficients. By substituting $\mathbf{v}_k$ in (2), we obtain:

$$\frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T\sum_{j=1}^{N}a_{jk}\phi(\mathbf{x}_j) = \lambda_k\sum_{i=1}^{N}a_{ik}\phi(\mathbf{x}_i). \quad (4)$$

Since the inner product in an RKHS is given by kernel function $\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) = k(\mathbf{x}_i,\mathbf{x}_j)$, by left-multiplying $\phi(\mathbf{x}_l)^T$ on both sides of (4), we obtain:

$$\frac{1}{N}\sum_{i=1}^{N}k(\mathbf{x}_l,\mathbf{x}_i)\sum_{j=1}^{N}a_{jk}k(\mathbf{x}_i,\mathbf{x}_j) = \lambda_k\sum_{i=1}^{N}a_{ik}k(\mathbf{x}_l,\mathbf{x}_i)$$

for all $l = 1,\ldots,N$. In matrix form, we have $K^2\mathbf{a}_k = N\lambda_k K\mathbf{a}_k$, where K is a "Gram matrix" whose element is $K_{ij} = k(\mathbf{x}_i,\mathbf{x}_j)$, and $\mathbf{a}_k = (a_{k1},\ldots,a_{kN})^T$, which satisfies the following eigenvalue problem [3]:

$$K\mathbf{a}_k = N\lambda_k\mathbf{a}_k.$$

If $\mathbf{v}_k$ has the unit norm, say, $\mathbf{v}_k^T\mathbf{v}_k = 1$, then $\mathbf{a}_k$ satisfies:

$$\lambda_k N\mathbf{a}_k^T\mathbf{a}_k = 1.$$

Finally, the nonlinear PCs can be calculated by:

$$\phi(\mathbf{x})^T\mathbf{v}_k = \sum_{i=1}^{N}a_{ik}k(\mathbf{x}_i,\mathbf{x}). \quad (5)$$

Remark 1: As mentioned earlier, the data is assumed to be centered. If that condition is not true, then we replace matrix K by $\bar{K} = K - \frac{1}{N}K1_N1_N^T - \frac{1}{N}1_N1_N^TK + \frac{1}{N^2}1_N1_N^TK1_N1_N^T$, where 1_N denotes an N-dimensional column vector of all ones. $\bar{K}$ is called a conditionally positive definite matrix [15].

2.3 Elastic Net-based Sparse PCA

Zou, Hastie, and Tibshirani [26] proposed sparsification for PCA via elastic net regularization. They point out that the PCA may be turned into a regression problem where the sparse coefficients may be obtained by imposing l_1-norm regularization. Specifically, PCA minimizes the criterion of MSE:

$$\min_U \sum_{i=1}^{N}\|\mathbf{x}_i - UU^T\mathbf{x}_i\|^2 \quad \text{s.t. } U^TU = I_{M\times M} \quad (6)$$

where U and $I_{M\times M}$ as previously mentioned. The cost function (6) may be changed into the following form:

$$\min_{U,B}\sum_{i=1}^{N}\|\mathbf{x}_i - UB^T\mathbf{x}_i\|^2 + \lambda\sum_{k=1}^{M}\|\mathbf{b}_k\|^2 \quad \text{s.t. } U^TU = I_{M\times M} \quad (7)$$

where $B = [\mathbf{b}_1,\ldots,\mathbf{b}_M]$ is a matrix of rank M, and $\lambda > 0$ is a parameter. The relation between (6) and (7) has been proved in Theorem 3 in [26]. The role played by the ridge penalty is to ensure the reconstruction of PCs rather than to penalize the regression coefficients. Note that:

$$\sum_{i=1}^{N}\|\mathbf{x}_i - UB^T\mathbf{x}_i\|^2 = \|X - XBU^T\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Since U is orthonormal, let $U_\perp$ be an orthonormal matrix such that $[U;U_\perp]$ is $d\times d$ orthonormal. Then, we obtain:

$$\begin{aligned}\|X - XBU^T\|_F^2 &= \|(X - XBU^T)[U;U_\perp]\|_F^2\\ &= \|XU - XB\|_F^2 + \|XU_\perp\|_F^2\\ &= \sum_{k=1}^{M}\|X\mathbf{u}_k - X\mathbf{b}_k\|^2 + \|XU_\perp\|_F^2.\end{aligned}$$

Given U, then the optimal solution B of (7) is minimized by

$$\min_B\sum_{k=1}^{M}\|X\mathbf{u}_k - X\mathbf{b}_k\|^2 + \lambda\sum_{k=1}^{M}\|\mathbf{b}_k\|^2 \quad (8)$$

which is equivalent to the M independent suboptimal problems. According to Theorem 1 in [26], after normalization such that each column of B has unit length, the optimal solution $\mathbf{b}_k$ is proportional to $\mathbf{u}_k$. Therefore, we can weaken criterion (6) into (7) in order to cope with PCA. Furthermore, through the above explanation, the first M sparse PCs can be obtained by adding l_1-norm of $\mathbf{b}_k$ into (8):

$$\min_{\mathbf{u}_k,\mathbf{b}_k}\sum_{k=1}^{M}\|X\mathbf{u}_k - X\mathbf{b}_k\|^2 + \lambda\sum_{k=1}^{M}\|\mathbf{b}_k\|^2 + \sum_{k=1}^{M}\lambda_{1,k}\|\mathbf{b}_k\|_1 \quad \text{s.t. } U^TU = I_{M\times M} \quad (9)$$

where $\lambda_{1,k} > 0$ are parameters, thus tuning the compromise between variance and sparseness.

The optimization problem (9) can be solved through an alternating algorithm as follows [26]:

- Given U: Each $\mathbf{b}_k$ in (9) is obtained as

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

and define matrix $B = [\mathbf{b}_1, \ldots, \mathbf{b}_M]$.
- Given B: Compute the singular value decomposition (SVD) of $(X^TX)B$ as

$$(X^TX)B = EDF^T$$

and set $U = EF^T$.
These two steps are repeated alternately until convergence is achieved.

On the other hand, according to (9), an alternating algorithm for optimizing $\mathbf{u}_k$ and $\mathbf{b}_k$ is introduced sequentially in [28]. First, for fixed $\mathbf{u}_k$, which is assumed to be orthonormal, say, $\mathbf{u}_k^T\mathbf{u}_k = 1$, then $\mathbf{b}_k$ can be calculated by:

$$\min_{\mathbf{b}_k} \|X\mathbf{u}_k - X\mathbf{b}_k\|^2 + \lambda\|\mathbf{b}_k\|^2 + \lambda_{1,k}\|\mathbf{b}_k\|_1. \quad (10)$$

For fixed $\mathbf{b}_k$, the solution of $\mathbf{u}_k$ is given by:

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

where $U_{(k-1)}$ is a $d \times (k-1)$ matrix corresponding to the previously found $(k-1)$ solutions $\mathbf{u}_k$, such that $U_{(k-1)}^T U_{(k-1)} = I$. The optimal $\mathbf{u}_k$ is given by $\mathbf{u}_k = \frac{\mathbf{s}}{\sqrt{\mathbf{s}^T\mathbf{s}}}$, where $\mathbf{s} = (I - U_{(k-1)}U_{(k-1)}^T)X^TX\mathbf{b}_k$. Compared to the method introduced by Zou et al. [26], Sjöstrand et al. [28] deem this method a sequential method. Consider estimating k components and $(k+1)$ components, respectively, the simultaneous algorithm provides different PCs results from k to $k+1$. The sequential method, however, keeps the first k PCs unchanged when the $(k+1)$th PC is computed.

So far we have introduced the PCA, KPCA and elastic net-based sparse PCA (SPCA). PCA is explained via an eigendecomposition of the covariance matrix S. For KPCA, the data is mapped into RKHS and we perform PCA in terms of a kernel trick. On the other hand, because of elastic net regularization, SPCA is more convenient than PCA for interpreting PCs. In the following section we will explain how KPCA can be rewritten as a regression problem with explicit expression, and then we wil employ the l_1-norm and l_{21}-norm of coefficients to achieve a sparse representation.

3 TWO NOVEL SPARSENESS ALGORITHMS FOR KPCA

Denote $\Phi = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]$, $V = [\mathbf{v}_1, \ldots, \mathbf{v}_M]$, and $A = [\mathbf{a}_1, \ldots, \mathbf{a}_M]$ for notational simplicity. Then we have matrix notation $V = \Phi A$ according to (3) and the Gram matrix $K = \Phi^T\Phi$. Our intent is to sparsify the coefficient matrix A and unveil PCs with matrix A. In other words, each $\mathbf{v}_i$ can be represented by a smaller number of observed samples than is the case with standard KPCA.

To this end, first we rewrite the cost function using the Pythagorean theorem. In detail, we reformulate KPCA according to reference [29]. In the same way as PCA, KPCA minimizes the MSE:

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

Suppose that Φ is fixed, such that optimizing V is equivalent to optimizing A. Let $R(\cdot)$ be the span of an operator, and $P_{R(\Phi)}$ be the orthogonal projector onto $R(\Phi)$. From the Pythagorean theorem, we have:

$$J[V] = \sum_{i=1}^{N}\{\|P_{R(\Phi)}(\phi(\mathbf{x}_i) - VV^T\phi(\mathbf{x}_i))\|^2 + \|P_{R(\Phi)^\perp}(\phi(\mathbf{x}_i) - VV^T\phi(\mathbf{x}_i))\|^2\},$$

since $P_{R(\Phi)} = \Phi(\Phi^T\Phi)^{-1}\Phi^T$. The first term in the brackets may be written as [29]:

$$\|P_{R(\Phi)}(\phi(\mathbf{x}_i) - VV^T\phi(\mathbf{x}_i))\|^2 = \|K^{-\frac{1}{2}}(\mathbf{h}_i - KAA^T\mathbf{h}_i)\|^2,$$

where $\mathbf{h}_i = \Phi^T\phi(\mathbf{x}_i)$. If K is rank-deficient, then we employ the pseudoinverse of $K^{\frac{1}{2}}$ [7]. The second term in the brackets may be written as:

$$\|P_{R(\Phi)^\perp}(\phi(\mathbf{x}_i) - VV^T\phi(\mathbf{x}_i)))\|^2 = \|P_{R(\Phi)^\perp}\phi(\mathbf{x}_i)\|^2.$$

Therefore, $J[V]$ can be rewritten as $J[V] = J_1[A] + J_2$, where:

$$J_1[A] = \sum_{i=1}^{N} \|K^{-\frac{1}{2}}\mathbf{h}_i - K^{\frac{1}{2}}AA^T\mathbf{h}_i\|^2$$

and:

$$J_2 = \sum_{i=1}^{N} \|P_{R(\Phi)^\perp}\phi(\mathbf{x}_i)\|^2.$$

Again, J_2 is a constant, which can be ignored, so we focus on the $J_1[A]$. If we set $\mathbf{y}_i = K^{-\frac{1}{2}}\mathbf{h}_i$, then we have $A^T\mathbf{h}_i = A^TK^{\frac{1}{2}}\mathbf{y}_i$. $J_1[A]$ can be rewritten as:

$$J_1[A] = \sum_{i=1}^{N} \|\mathbf{y}_i - QQ^T\mathbf{y}_i\|^2 \quad \text{s.t. } Q^TQ = I_{M\times M} \quad (11)$$

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

$$\mathbf{q}_k = K^{\frac{1}{2}}\mathbf{a}_k. \quad (12)$$

Note that:

$$\begin{aligned}[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] &= [K^{-\frac{1}{2}}\mathbf{h}_1, K^{-\frac{1}{2}}\mathbf{h}_2, \ldots, K^{-\frac{1}{2}}\mathbf{h}_N] \\ &= K^{-\frac{1}{2}}\Phi^T[\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_N)] \\ &= K^{-\frac{1}{2}}K = K^{\frac{1}{2}}.\end{aligned}$$

Obviously, matrix $K^{\frac{1}{2}}$ is composed of N column vectors $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$. Based on the above analysis, we see that from the Pythagorean theorem, optimizing $J[V]$ is equivalent to optimizing $J_1[A]$. It should also be noted that the condition links between A and Q via (12). We propose two types of sparsifications for A. One is based on the l_1-norm, while the other is based on the $l_{2,1}$-norm. We have shown a preliminary result based on the l_1-norm in [21]. In the following, we describe a more detailed derivation for the l_1-norm regularization and a newly proposed sparsification based on the $l_{2,1}$-norm.

3.1 Sparse KPCA via l_1-norm

As was done in the derivation of sparse PCA, to obtain a sparse solution, we weaken (11) as a regression-type optimization problem as follows:

$$\begin{aligned}\min_{P,Q} \sum_{i=1}^{N} \|\mathbf{y}_i - PQ^T\mathbf{y}_i\|^2 + \lambda \sum_{k=1}^{M} \|\mathbf{q}_k\|^2 \\ \text{s.t. } P^TP = I_{M\times M}\end{aligned} \tag{13}$$

where $P = [\mathbf{p}_1, \ldots, \mathbf{p}_M]$, and $\lambda > 0$ is a parameter. In fact in (11), it is just another form of (6) except for the different denotation, so in (13) we first follow the procedure of sparse PCA to make sure P has orthonormal columns, then relax $P = Q$ and add a ridge penalty term $\lambda \sum_{k=1}^{M} \|\mathbf{q}_k\|^2$. Suppose P is given, the optimal solution Q of (13) can be obtained by minimizing:

$$\frac{1}{2}\sum_{k=1}^{M} \|K^{\frac{1}{2}}\mathbf{p}_k - K^{\frac{1}{2}}\mathbf{q}_k\|^2 + \frac{\lambda}{2}\sum_{k=1}^{M} \|\mathbf{q}_k\|^2. \tag{14}$$

The factor of 1/2 is included for convenience. As shown in Section 2.3, after normalization the $\mathbf{q}_k$ is proportional to $\mathbf{p}_k$. Our next step is to obtain sparse solutions for the coefficient matrix A. Note that $\mathbf{a}_k$ is a coefficient vector in terms of (3) and satisfies (12). We enforce l_1-norm of $\mathbf{a}_k$ onto (14) and minimize the cost function:

$$\begin{aligned}J_{l_1}[P, Q] =& \frac{1}{2}\sum_{k=1}^{M} \|K^{\frac{1}{2}}\mathbf{p}_k - K^{\frac{1}{2}}\mathbf{q}_k\|^2 + \frac{\lambda}{2}\sum_{k=1}^{M} \|\mathbf{q}_k\|^2 \\ &+ \sum_{k=1}^{M} \lambda_{1,k}\|\mathbf{a}_k\|_1\end{aligned}$$

subject to $P^TP = I_{M\times M}$ or we can consider optimizing them sequentially:

$$\begin{aligned}J_{l_1}[\mathbf{p}_k, \mathbf{q}_k] =& \frac{1}{2}\|K^{\frac{1}{2}}\mathbf{p}_k - K^{\frac{1}{2}}\mathbf{q}_k\|^2 + \frac{\lambda}{2}\|\mathbf{q}_k\|^2 \\ &+ \lambda_{1,k}\|\mathbf{a}_k\|_1 \\ &\text{s.t. } P^TP = I_{M\times M}\end{aligned} \tag{15}$$

where $\lambda_{1,k} > 0$ is the trade-off parameter of MSE and sparseness. Cost function (15) is quadratic with respect to either $\mathbf{p}_k$ or $\mathbf{q}_k$, both of which are minimized by alternating minimization. For $\mathbf{p}_k$, only the first term is considered: $\mathbf{p}_k$ can be solved using the same technique as $\mathbf{u}_k$ for SPCA, as shown in Section 2.3. For $\mathbf{q}_k$, we cannot apply the same scenario as SPCA described in Section 2.3, since the l_1-norm includes a matrix $K^{-\frac{1}{2}}$, which is generally not diagonal. For this case, the ADMM [27] can be applied to (12) and (15) using an augmented Lagrangian method. We form the augmented Lagrangian by combining (12) and (15):

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

where $\mathbf{t}_k$ is the Lagrange multiplier and $\rho > 0$ is a penalty parameter. ADMM consists of the iterations:

$$\begin{aligned}\mathbf{q}_k^{j+1} &= \arg\min_{\mathbf{q_k}} L_\rho(\mathbf{q}_k, \mathbf{a}_k^j, \mathbf{t}_k^j) \\ &= (K + \lambda I + \rho K^{-1})^{-1}[K\mathbf{p}_k + \rho K^{-\frac{1}{2}}(\mathbf{a}_k^j - \frac{\mathbf{t}_k^j}{\rho})], \\ \mathbf{a}_k^{j+1} &= \arg\min_{\mathbf{a_k}} L_\rho(\mathbf{q}_k^{j+1}, \mathbf{a}_k, \mathbf{t}_k^j) \\ &= S_{\frac{\lambda_{1,k}}{\rho}}(K^{-\frac{1}{2}}\mathbf{q}_k^{j+1} + \frac{\mathbf{t}_k^j}{\rho}), \\ \mathbf{t}_k^{j+1} &= L_\rho(\mathbf{q}_k^{j+1}, \mathbf{a}_k^{j+1}, \mathbf{t}_k) \\ &= \mathbf{t}_k^j + \rho(K^{-\frac{1}{2}}\mathbf{q}_k^{j+1} - \mathbf{a}_k^{j+1})\end{aligned}$$

where $S_\tau(\alpha) = \dfrac{\max\{|\alpha| - \tau, 0\}}{\max\{|\alpha| - \tau, 0\} + \tau}\alpha$ [30] is the element-wise soft thresholding function, and $\mathbf{q}_k^{j+1}$ means the $(j+1)$th iteration of $\mathbf{q}_k$. We consider stopping criteria [27] when

$$\begin{aligned}\|\mathbf{e}_p^{j+1}\| &\le \sqrt{N}\epsilon^{abs} + \epsilon^{rel}\max\{\|K^{-\frac{1}{2}}\mathbf{q}_k^{j+1}\|, \|\mathbf{a}_k^{j+1}\|\} \\ \|\mathbf{e}_d^{j+1}\| &\le \sqrt{N}\epsilon^{abs} + \epsilon^{rel}\|K^{-\frac{1}{2}}\mathbf{t}_k^{j+1}\|\end{aligned}$$

where $\epsilon^{abs} > 0$, $\epsilon^{rel} > 0$, and $\mathbf{e}_p^{j+1} = K^{-\frac{1}{2}}\mathbf{q}_k^{j+1} - \mathbf{a}_k^{j+1}$, $\mathbf{e}_d^{j+1} = -\rho K^{-\frac{1}{2}}(\mathbf{a}_k^{j+1} - \mathbf{a}_k^j)$. The sparse coefficients $\mathbf{a}_k$ can be obtained by applying the soft thresholding function. Then we compute $\mathbf{p}_k$ by the same scheme $\mathbf{u}_k$ as mentioned in Section 2.3. The algorithm of sparse KPCA via l_1-norm is summarized in Algorithm 1. In Step 11 of the Algorithm 1, $P_{(k-1)}$ is

Algorithm 1 Sparse KPCA via l_1-norm

Input: matrices K and P, the number of PCs M
for $k = 1$ **to** M **do**
  while not convergent or within the preset iteration **do**
    Initialize $\mathbf{q}_k, \mathbf{a}_k, \mathbf{t}_k$
    while stopping criteria is not satisfied and within the preset iteration **do**
      $\mathbf{q}_k \leftarrow (K + \lambda I + \rho K^{-1})^{-1}[K\mathbf{p}_k + \rho K^{-\frac{1}{2}}(\mathbf{a}_k - \frac{\mathbf{t}_k}{\rho})]$
      $\mathbf{a}_k \leftarrow S_{\frac{\lambda_{1,k}}{\rho}}(K^{-\frac{1}{2}}\mathbf{q}_k + \frac{\mathbf{t}_k}{\rho})$
      $\mathbf{t}_k \leftarrow \mathbf{t}_k + \rho(K^{-\frac{1}{2}}\mathbf{q}_k - \mathbf{a}_k)$
    end while
    $\mathbf{q}_k \leftarrow \frac{\mathbf{q}_k}{\sqrt{\mathbf{q}_k^T\mathbf{q}_k}}$
    $\mathbf{p}_k = (I - P_{(k-1)}P_{(k-1)}^T)K\mathbf{q}_k$
    $\mathbf{p}_k \leftarrow \frac{\mathbf{p}_k}{\sqrt{\mathbf{p}_k^T\mathbf{p}_k}}$
  end while
end for
Output the coefficient $A = [\mathbf{a}_1, \ldots, \mathbf{a}_M]$

defined as the submatrix that consists of the previous $(k-1)$ solutions $\mathbf{p}_k$.

Remark 2: We calculate the eigendecomposition of K and sort the eigenvalues in descending order. The first M corresponding eigenvectors are set to $P = [\mathbf{p_1}, \ldots, \mathbf{p}_M]$.

3.2 Sparse KPCA via $l_{2,1}$-norm

The l_1-norm regularization leads to element-wise sparsity via a soft-threshold operator, so matrix A is not sparse in rows. Namely, each PC includes different samples to represent a sparse solution. We aim to encourage the sparsity of matrix A at the row level, so we adopt $l_{2,1}$-norm regularization. The $l_{2,1}$-norm regularization penalizes all the coefficients for a given set of training data to become zero simultaneously. In this way, we can greatly reduce the number of samples to represent all PCs.

First, we express (14) in matrix form:

$$\frac{1}{2}\|K^{\frac{1}{2}}P - K^{\frac{1}{2}}Q\|_F^2 + \frac{\lambda}{2}\|Q\|_F^2 \tag{16}$$

The $l_{2,1}$-norm of A (a.k.a. group l_1-norm) is defined as $\|A\|_{2,1} = \sum_{i=1}^{N}\|\mathbf{a}^i\|$, where $\mathbf{a}^i$ denotes the ith row of A. We add the $l_{2,1}$-norm of A to (16):

$$\begin{aligned} J_{l_{2,1}}[P,Q] =& \frac{1}{2}\|K^{\frac{1}{2}}P - K^{\frac{1}{2}}Q\|_F^2 + \frac{\lambda}{2}\|Q\|_F^2 \\ &+ \mu_{21}\|A\|_{2,1} \\ &\text{s.t. } P^TP = I_{M\times M} \end{aligned} \tag{17}$$

where μ_{21} is a constant that determines the trade-off between MSE and sparsity. Combining (12) and (17), we form the augmented Lagrangian:

$$\begin{aligned} L_{\rho_{21}}(Q,A,T) =& \frac{1}{2}\|K^{\frac{1}{2}}P - K^{\frac{1}{2}}Q\|_F^2 + \frac{\lambda}{2}\|Q\|_F^2 \\ &+ \mu_{21}\|A\|_{2,1} - tr(T^T(K^{-\frac{1}{2}}Q - A)) \\ &+ \frac{\rho_{21}}{2}\|K^{-\frac{1}{2}}Q - A\|_F^2 \end{aligned}$$

where T is the Lagrange multiplier, $tr(\cdot)$ stands for the trace of the matrix, and $\rho_{21} > 0$. By using the scaled dual variable $W = \frac{T}{\rho_{21}}$, $L_{\rho_{21}}(Q,A,T)$ can be reformed as:

$$\begin{aligned} L_{\rho_{21}}(Q,A,W) =& \frac{1}{2}\|K^{\frac{1}{2}}P - K^{\frac{1}{2}}Q\|_F^2 + \frac{\lambda}{2}\|Q\|_F^2 \\ &+ \mu_{21}\|A\|_{2,1} + \frac{\rho_{21}}{2}\|K^{-\frac{1}{2}}Q - A - W\|_F^2 \\ &- \frac{\rho_{21}}{2}\|W\|_F^2. \end{aligned}$$

ADMM optimizes $L_{\rho_{21}}(Q,A,W)$ with respect to Q, A, and updates the dual variable W sequentially. First, optimizing Q^{j+1} for fixed A^j and W^j:

$$\begin{aligned} Q^{j+1} &= \arg\min_Q L_{\rho_{21}}(Q, A^j, W^j) \\ &= (K + \lambda I + \rho_{21}K^{-1})^{-1}[KP + \rho_{21}K^{-\frac{1}{2}}(A + W)]. \end{aligned}$$

Then, with fixed Q^{j+1} and W^j, A^{j+1} is computed by:

$$\begin{aligned} A^{j+1} &= \arg\min_A L_{\rho_{21}}(Q^{j+1}, A, W^j) \\ &= \arg\min_A \mu_{21}\|A\|_{2,1} + \frac{\rho_{21}}{2}\|A - (K^{-\frac{1}{2}}Q - W)\|_F^2 \\ &= \arg\min_{\mathbf{a}^i} \sum_{i=1}^{N}(\mu_{21}\|\mathbf{a}^i\| + \frac{\rho_{21}}{2}\|\mathbf{a}^i - \mathbf{r}^i\|^2) \end{aligned} \tag{18}$$

where $\mathbf{r}^i$ is the ith row vector of $R = K^{-\frac{1}{2}}Q - W$. Note that different row vectors $\mathbf{a}^i$ of A can be solved for by an independent sub-problem, i.e.:

$$\mathbf{a}^i = \arg\min_{\mathbf{a}^i}(\mu_{21}\|\mathbf{a}^i\| + \frac{\rho_{21}}{2}\|\mathbf{a}^i - \mathbf{r}^i\|^2) \quad i = 1, \ldots, N.$$

The solution of $\mathbf{a}^i$ is the vectorial soft-threshold operator [30] given by:

$$\mathbf{a}^i = \mathbf{r}^i\frac{\max\{\|\mathbf{r}^i\| - \beta, 0\}}{\max\{\|\mathbf{r}^i\| - \beta, 0\} + \beta} \tag{19}$$

where $\beta = \frac{\mu_{21}}{\rho_{21}}$. So A^{j+1} can be written as $A^{j+1} = [\mathbf{a}^1; \mathbf{a}^2; \cdots; \mathbf{a}^N]$. Using the vectorial soft-threshold operator, many rows of the optimal A corresponding to (18) shrink to zero, which makes A suitable for sample data selection. Finally, we update the dual variable.

$$W^{j+1} = L_{\rho_{21}}(Q^{j+1}, A^{j+1}, W) = W - K^{-\frac{1}{2}}Q + A.$$

Based on the above derivation, the details of sparse KPCA via $l_{2,1}$-norm are summarized in Algorithm 2. Again, the initial matrix P is set as shown in Algorithm 1. After computing Q, we calculate the SVD of $KQ = E_1 D_1 F_1^T$, and set $P = E_1 F_1^T$ in the loop. Primal residual norms $\|E_p^{j+1}\|_F = \|K^{-\frac{1}{2}} Q^{j+1} - A^{j+1}\|_F$ and dual residual norms $\|E_d^{j+1}\|_F = \|\rho_1 K^{-\frac{1}{2}}(A^{j+1} - A^j)\|_F$ are used in Algorithm 2. With the goal of enhancing the convergence speed in the algorithm in mind, we consider the self-adaptive rule for penalty parameter ρ_1 [27]:

Algorithm 2 Sparse KPCA via $l_{2,1}$-norm

1: Input: matrices K and P, the number of PCs M
2: **while** not convergent or within the preset iteration **do**
3: Initialize Q, A, W
4: **while** within the preset iteration and stopping criteria is not satisfied **do**
5: $Q \leftarrow (K + \lambda I + \rho_{21} K^{-1})^{-1}[KP + \rho_{21} K^{-\frac{1}{2}}(A + W)]$
6: $A \leftarrow [\mathbf{a}^1; \mathbf{a}^2; \cdots; \mathbf{a}^N]$ where each $\mathbf{a}^i$ is given by (19)
7: $W \leftarrow W - K^{-\frac{1}{2}} Q + A$
8: **end while**
9: $KQ = E_1 D_1 F_1^T$
10: $P = E_1 F_1^T$
11: **end while**
12: Output the coefficient $A = [\mathbf{a}^1; \cdots; \mathbf{a}^N]$

$$\rho_1^{k+1} = \begin{cases} \tau^{incr} \rho_{21}^k, & \text{if } \|E_p^{j+1}\|_F > \eta \|E_d^{j+1}\|_F \\ \rho_{21}^k / \tau^{decr}, & \text{if } \|E_d^{j+1}\|_F > \eta \|E_p^{j+1}\|_F \\ \rho_{21}^k, & \text{otherwise} \end{cases}$$

where $\eta > 1$, $\tau^{incr} > 1$, and $\tau^{decr} > 1$ are parameters. In the ADMM scheme, we set $\eta = 10$, and $\tau^{incr} = \tau^{decr} = 2$. Meanwhile, in addition to updating the parameter ρ_{21}, we also need to rescale W. When ρ_{21} is halved, W should be doubled before proceeding. Conversely, if ρ_{21} is doubled, then W should be halved before proceeding. The convergence criterion in ADMM is $\|E_p^j\|_F < \delta_{l_{2,1}}$ or $\|E_d^j\|_F < \delta_{l_{2,1}}$ with a declared maximal number of iterations.

4 EXPERIMENTS

We present the sparsity results of the proposed approach through several experiments. Gaussian kernels of the form $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$ will be applied to implement the dataset. In Section 4.1, we use two unsupervised toy examples—Gaussian mixture data and nonlinear data—to illustrate the effectiveness of the sparsity. In Section 4.2, three real datasets from the UCI Machine Learning Repository [32] are processed to corroborate performance of the proposed approach. Herein, we refer to sparse KPCA via l_1-norm and sparse KPCA via $l_{2,1}$-norm as SKPCA-l_1 and SKPCA-$l_{2,1}$, respectively.

4.1 Toy Examples

4.1.1 Gaussian Mixture Data

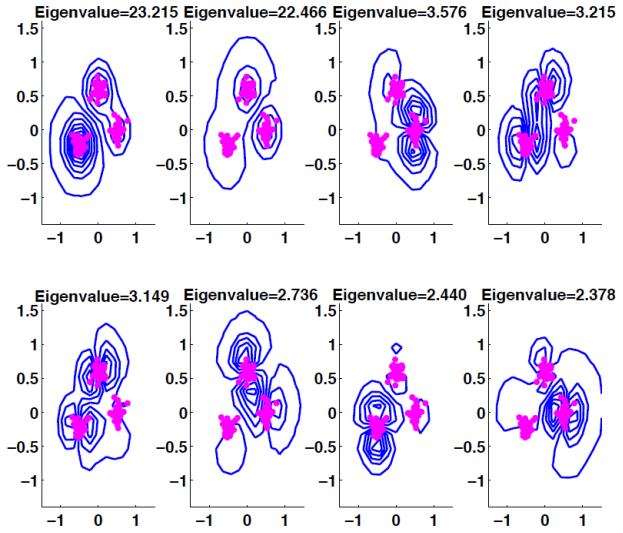
In the first toy example, we conduct the experiment on the 90 data samples. The data values are generated from three Gaussian sources centered at $(-0.5, -0.2)$, $(0, 0.6)$, and $(0.5, 0)$ (30 samples each), with standard deviation 0.1. The parameter of the kernel function is set to $\gamma = 10$ [3].

In the case of SKPCA-l_1, we set $\lambda = 0.1$, $\rho = 0.01$, $\epsilon^{abs} = 10^{-4}$, $\epsilon^{rel} = 10^{-4}$, and $\lambda_1 = (0.02, 0.02, 0.009, 0.008, 0.006, 0.007, 0.01, 0.01)$. The sparse solutions can be achieved when the convergence criterion $\|\mathbf{q}_k^{new} - \mathbf{q}_k^{old}\|^2 < \epsilon_{l_1}$ is satisfied or the preset maximum number of loop iterations have occurred, where $\mathbf{q}_k^{new}$ and $\mathbf{q}_k^{old}$ can be thought of as the $\mathbf{q}_k$ during the update procedure in Steps 4 and 13 in Algorithm 1. We set $\epsilon_{l_1} = 10^{-2}$, and the inner and outer loop counters are 300. In the case of SKPCA-$l_{2,1}$, we set $\lambda = 0.01$, $\mu_{21} = 0.01$, and the initial $\rho_{21} = 10^{-2}$, $\delta_{l_{2,1}} = 90 \times 10^{-4}$. Also, the sparse solutions can be achieved when $\|Q^{old} - Q^{new}\|_F^2 < \epsilon_{l_{2,1}}$ or the preset maximum number of loop iterations have occurred, where $\epsilon_{l_{2,1}} = 10^{-2}$. Q^{old} and Q^{new} can be regarded as the Q during the loop in Steps 3 and 8 in Algorithm 2. The inner iteration limit is set to 300, and the outer iteration limit is 10. In the proposed methods, all the parameters are set manually.

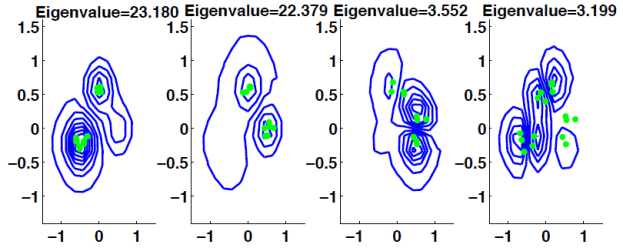
Fig. 1(a) shows the result when KPCA is performed on this data. The dots denote data, while the contour lines shown in each part of the figure represent constant value, calculated by (5). SKPCA-l_1 and SKPCA-$l_{2,1}$ are shown in Fig. 1(b) and 1(c), respectively. The dots are data samples selected by the proposed method, in which the corresponding coefficients are nonzero. Fig. 1 shows that SKPCA-l_1 captures a similar structure with different data samples, whereas SKPCA-$l_{2,1}$ extracts a similar structure with same data samples. Compared to KPCA, both types of sparsity algorithms yield desirable results using less data.

4.1.2 Non-linear Data

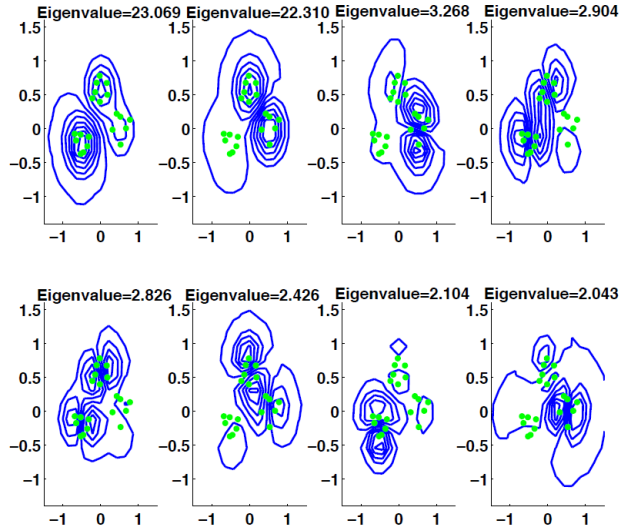
We generate $N = 500$ training data from a two-dimensional parabola in the second instance, where x is generated from $[-1, 1]$ with uniform distribution. The y-values are generated from $y_i = x_i^2 + \xi$, where ξ is normal noise with standard deviation 0.2 [31]. Fig. 2(a) shows the first four PCs obtained by KPCA with $\gamma = 2$; the dots represent data samples. In the case of SKPCA-l_1, we set $\lambda = 0.05$, $\lambda_{1,4} = 0.05$, $\rho = 0.09$, $\epsilon^{rel} = 10^{-4}$, $\epsilon^{abs} = 10^{-4}$, and $\epsilon_{l_1} = 10^{-2}$ to extract the first four PCs. The inner loop counter is 300, and the outer loop counter is 60. The result is shown in Fig. 2(b), where the dots represent samples that have nonzero coefficients. It is obvious that the SKPCA-l_1 method only uses a few data samples to capture the same performance



(a) KPCA



(b) SKPCA-l_1



(c) SKPCA-$l_{2,1}$

Fig.1: *Visualization of the first eight PCs captured by KPCA, SKPCA-l_1, and SKPCA-$l_{2,1}$.*

as the KPCA. To investigate how the parameter $\lambda_{1,k}$ affects the sparsity and variance, we again fixed ρ and set $\lambda_{1,4} = (0.05, 0.07, 0.08, 0.09)$ for the original data. Fig. 2(c) shows this influence. The first PC is the same as the first figure of Fig. 2(b) because $\lambda_{1,1}$ is unchanged. By increasing the parameter value $\lambda_{1,k}$ from second to fourth, the different $\lambda_{1,k}$ leads to different numbers of data points, as well as decreasing the variance, which confirmed the connection between the approximation property and the sparsity of coefficients. Through use of the element-wise soft thresholding function $S_\tau(\alpha) = \dfrac{\max\{|\alpha| - \tau, 0\}}{\max\{|\alpha| - \tau, 0\} + \tau}\alpha$, below some threshold $\tau = \dfrac{\lambda_{1,k}}{\rho}$, the data will be eliminated in feature extraction, thus leading to sparser results. In the case of SKPCA-$l_{2,1}$, we set $\lambda = 0.05$, $\mu_{21} = 0.05$, the initial value $\rho_{21} = 0.09$, and $\delta_{l_{2,1}} = 500 \times 10^{-4}$. The iteration numbers are the same as SKPCA-l_1. Fig. 2(d) illustrates the result. The first four PCs extracted by the same data and the structure captured by SKPCA-$l_{2,1}$ is essentially the same as with KPCA. As shown in Fig. 2, sample selection and sparse representation can be achieved by using a small number of samples, which supports the effectiveness of the sparseness approach.

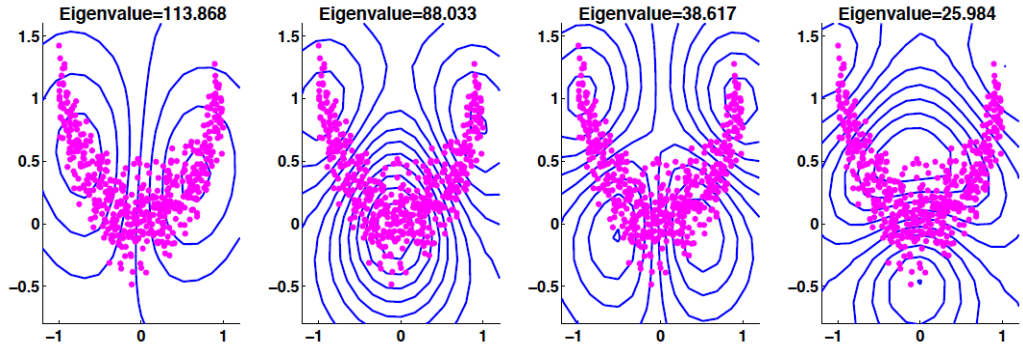
4.2 UCI Dataset

To validate the usefulness of the proposed algorithm, we processed three real datasets which are available from the UCI Machine Learning Repository [32]. Those datasets are used for classification. The details of the datasets are shown in Table 1.

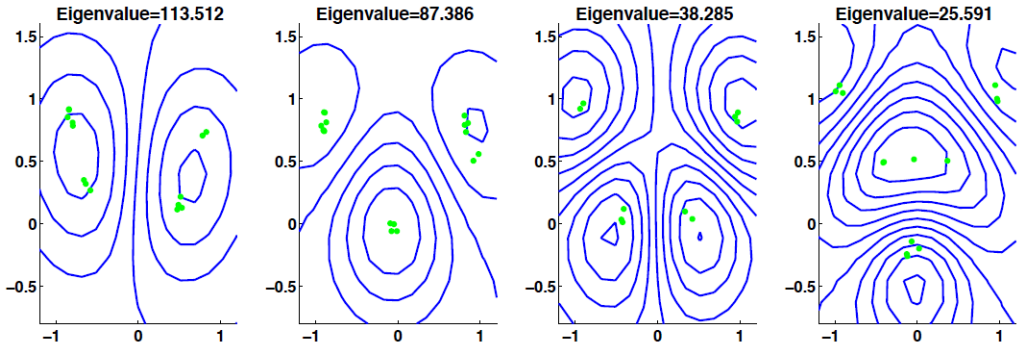
Table 1: *Dataset used in classification experiments*

Dataset	N(#num)	m(feature)
Australian	690	14
breast-cancer	699	10
climate	540	20

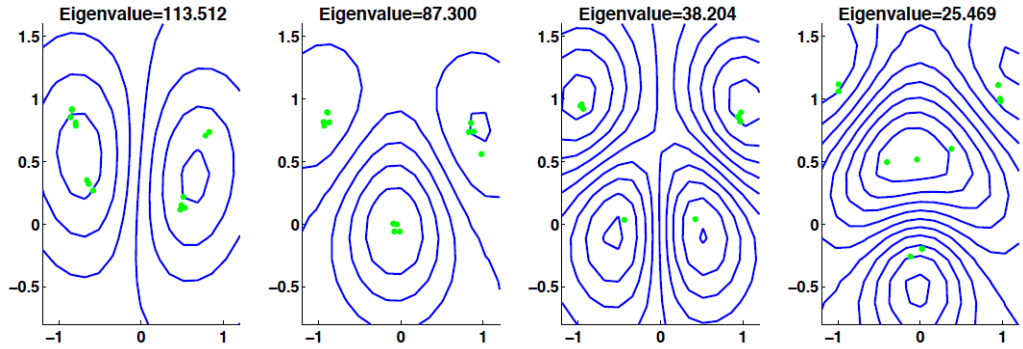
First we used a support vector machine (SVM) to classify the dataset. The second classification method is to employ KPCA as preprocessing step to reduce dimensionality and then to implement SVM (KPCA+SVM). The proposed methods are implemented for comparision, referred to as SKPCA-l_1+SVM and SKCPA-l_{21}+SVM, respectively. Each dataset is divided into two parts: the training and test data. The training data comes from half of the entire data which is used for training and for optimizing the parameters, and the rest for the final evaluation. The training data is used normalized to the range [0,1]. The SVM parameters C_{svm} as well as γ_{svm} are selected from the candidate set $\{10^{-3}, 10^{-2}, 10^{-1}, 0.5, 1, 5, 10, 10^2\}$ and $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. In KPCA, the Gaussian kernel function parameter γ is cho-



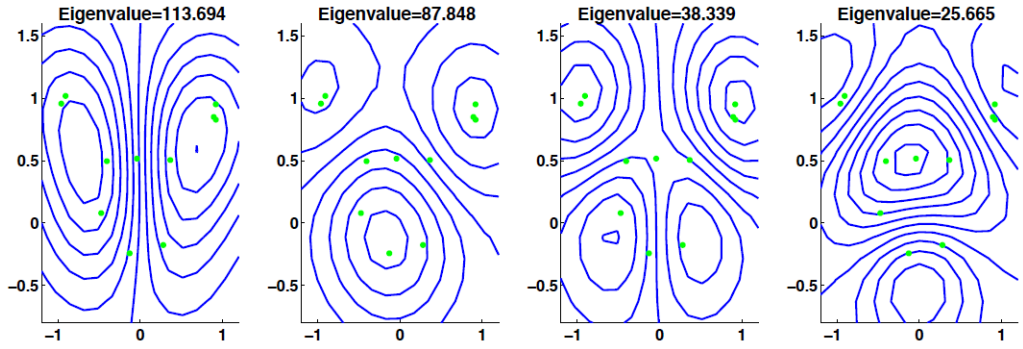
(a) KPCA



(b) SKPCA-l_1 with $\lambda_{1,4} = 0.05$, $\rho = 0.09$



(c) SKPCA-l_1 with $\lambda_{1,4} = (0.05, 0.07, 0.08, 0.09)$, $\rho = 0.09$



(d) SKPCA-$l_{2,1}$

Fig.2: *The first four PCs contrast results among KPCA, SKPCA-l_1, and SKPCA-$l_{2,1}$.*

Table 2: *Some hyperparameters in the proposed method*

	SKPCA-l_1	SKPCA-$l_{2,1}$
Australian	$\lambda_1 = 10^{-3}$, $\rho = 10^{-2}$	$\mu_{21} = 10^{-2}$, $\rho_{21} = 10^{-2}$
breast-cancer	$\lambda_1 = 10^{-3}$, $\rho = 10^{-2}$	$\mu_{21} = 10^{-2}$, $\rho_{21} = 10^{-2}$
climate	$\lambda_1 = 10^{-2}$, $\rho = 10^{-1}$	$\mu_{21} = 10^{-1}$, $\rho_{21} = 10^{-2}$

Table 3: *Average result of accuracy, precision, recall, and F1 score in Australian dataset*

Australian	accuracy	precision	recall	F1 score	sparsity
SVM	$\mathbf{0.8461 \pm 0.0177}$	$\mathbf{0.9118 \pm 0.0308}$	0.8011 ± 0.0378	0.8519 ± 0.0213	0.5183 ± 0.2119
KPCA+SVM	0.8452 ± 0.0233	0.8892 ± 0.0371	$\mathbf{0.8252 \pm 0.0351}$	$\mathbf{0.8552 \pm 0.0235}$	0
SKPCA-l_1+SVM	0.8394 ± 0.0178	0.8950 ± 0.0427	0.8071 ± 0.0200	0.8480 ± 0.0175	0.0006 ± 0.0017
SKPCA-$l_{2,1}$+SVM	0.8455 ± 0.0190	0.8967 ± 0.0357	0.8169 ± 0.0312	0.8541 ± 0.0205	$\mathbf{0.5954 \pm 0.1980}$

Table 4: *Average result of accuracy, precision, recall, and F1 score in breast-cancer dataset*

breast-cancer	accuracy	precision	recall	F1 score	sparsity
SVM	0.9611 ± 0.0068	0.9730 ± 0.0093	0.9668 ± 0.0053	0.9699 ± 0.0050	0.8516 ± 0.0542
KPCA+SVM	0.9620 ± 0.0084	0.9734 ± 0.0074	0.9678 ± 0.0078	0.9706 ± 0.0063	0
SKPCA-l_1+SVM	0.9614 ± 0.0112	0.9700 ± 0.0104	$\mathbf{0.9704 \pm 0.0075}$	0.9702 ± 0.0085	0.3630 ± 0.1865
SKPCA-$l_{2,1}$+SVM	$\mathbf{0.9626 \pm 0.0080}$	$\mathbf{0.9750 \pm 0.0119}$	0.9672 ± 0.0089	$\mathbf{0.9710 \pm 0.0058}$	$\mathbf{0.8937 \pm 0.0443}$

Table 5: *Average result of accuracy, precision, recall, and F1 score in climate dataset*

climate	accuracy	precision	recall	F1 score	sparsity
SVM	0.9470 ± 0.0170	$\mathbf{0.7911 \pm 0.0899}$	0.5469 ± 0.1121	0.6364 ± 0.0992	$\mathbf{0.8074 \pm 0.0323}$
KPCA+SVM	0.9456 ± 0.0142	0.7343 ± 0.0732	0.5701 ± 0.1319	0.6325 ± 0.1036	0
SKPCA-l_1+SVM	0.9411 ± 0.0234	0.6758 ± 0.2419	0.5111 ± 0.2053	0.5732 ± 0.2165	0.0037 ± 0.0033
SKPCA-$l_{2,1}$+SVM	$\mathbf{0.9493 \pm 0.0155}$	07859 ± 0.0959	$\mathbf{0.5846 \pm 0.1154}$	$\mathbf{0.6618 \pm 0.0889}$	0.5541 ± 0.0649

sen from the set $\{10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 6 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$ and the number of PCs are selected from 1 to 30. In each method, we use the combination of each parameter set and obtain the accuracy with 5 fold cross validation on the training data. The optimal parameters are choosed by the largest average accuracy. Some hyperparameters in the proposed method are set as follows: $\lambda = 0.001$, $\epsilon_{l_1} = \epsilon_{l_{2,1}} = 0.01$, $\epsilon^{abs} = \epsilon^{rel} = 10^{-4}$, $\delta_{l_{21}} = N \times 10^{-4}$, where N is the number of training data. The loop numbers of inner and outer loops for SKPCA-l_1 are set to 10 and 5, respectively. In the SKPCA-$l_{2,1}$ case, 30 and 10 are used for breast and climate, 10 and 10 are used for Australian. The other parameters are displayed in Table 2. Those parameters are set manually. We repeated each dataset 10 times and reported the final average metrics on the test data. Tables 3, 4, and 5 shown the accuracy, precision, recall, F1 score, and sparsity with standard deviation of each dataset. The best estimate is marked in bold.

We first compare the accuracy of SVM, KPCA, and the proposed method. In Australian dataset, SVM gives the best accuracy. SKPCA-l_1 decreases by 0.0067 and SKPCA-$l_{2,1}$ decreases by 0.0006. The KPCA is slightly higher than SKPCA-l_1, while relatively lower than SKPCA-$l_{2,1}$. In breast-cancer dataset, SKPCA-$l_{2,1}$ shows the best result. SKPCA-l_1 decreases by 0.0012 and KPCA decreases by 0.0006. SVM gives realtively low accuracy. In climate dataset, SKPCA-$l_{2,1}$ expresses the best result. SVM decreases by 0.0023, SKPCA-l_1 decreases by 0.0082, and KPCA decreases by 0.0037. It can be seen that the accuracy of KPCA lies between SKPCA-l_1 and SKPCA-$l_{2,1}$. These results suggest that the accuracy obtained by the proposed method is close to KPCA.

Secondly, from the sparsity results, the SKPCA-l_1 gives the very low sparsity, while the SKPCA-$l_{2,1}$ gives relatively higher sparsity. This is because that each PC is expressed by different samples in the l_1-norm case, resulting to the low average sparsity. However, in the $l_{2,1}$ case, each PC is explained by the same samples, leading to the high average sparsity. KPCA uses all the samples to express the PC, so the sparsity is zero. Compared with SVM, SKPCA-$l_{2,1}$ gives relatively higher sparsity results on Australian and breast-cancer, while the results have lower sparsity than climate. It can be seen that the proposed method enhances the interpretation of PC and obtains similar accuracy as KPCA.

Finally, precision, recall, and F1 score are calculated to compare the performance. The F1 score is a harmonic mean of the precision and recall. We check the performance based on F1 score. As shown in Table 3, KPCA gives the best score in australian classification, while in Tables 4 and 5, SKPCA-$l_{2,1}$ achieves

the best score. It can be seen that the KPCA is higher than SKPCA-l_1 and lower than SKPCA-$l_{2,1}$ in the breast-cancer and climate datasets. This further confirms the effectiveness of the proposed method. Though the F1 score is lower than KPCA in australian dataset, the F1 score decreases by less than 0.01, indicating that there exists a need to improve the classification by adjusting the hyperparameters.

5 CONCLUSION

In this paper, we proposed two types of sparsity approaches to KPCA, i.e. SKPCA-l_1 and SKPCA-l_{21}. First, we reformulate the MSE function into a regression-framework optimization problem and then incorporate the l_1-norm and $l_{2,1}$-norm into the regression criterion, respectively. With the introduction of the SKPCA-l_1 and SKPCA-$l_{2,1}$, we developed an algorithm for the proposed method that includes ADMM, to obtain a sparse coefficient matrix using a thresholding function. The training data which contributes little to the representation of kernel function can be reduced via zero elements in the coefficient matrix. The performance is demonstrated by comparison with standard KPCA. In toy examples, the proposed approach makes PC interpretation easier with less training data. For the real datasets, in combination with SVM, the classification accuracy is similar to standard KPCA. The SKPCA-$l_{2,1}$ method yields much more sparsity than SVM on Australian and breast-cancer datasets in the UCI repository thanks to $l_{2,1}$-norm. Although the proposed method obtained the sparsification results, one needs to choose the tuning parameters such as λ_1, ρ, μ_{21}, and ρ_{21} appropriately. The threshold values are determined by $\frac{\lambda_1}{\rho}$ in SKPCA-l_1 and $\frac{\mu_{21}}{\rho_{21}}$ in SKPCA-$l_{2,1}$. These parameters affect the sparsity and the classification task. In future work, we will investigate how to set them appropriately and reduce the cost of computation.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grant Number 26280054. The authors would like to thank the reviewers, as well as Associate Professor Dr. Theekapun Charoenpong, Srinakharinwirot University, for a critical reading of the manuscript, which led to a significant improvement of this paper.

References

[1] I. Jolliffe, *Principal Component Analysis.* Springer, 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. 299–1319, July, 1998.

[4] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[5] A. J. Smola, O. L. Mangasarian, and B. Schölkopf, "Sparse kernel feature analysis," In *Classification, Automation, and New Media*, Springer, pp. 167–178, 2002.

[6] A. Smola, and B. Schölkopf, (2000). "Sparse greedy matrix approximation for machine learning," In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, San Francisco, CA. Morgan Kaufmann, pp. 911–918, 2000.

[7] 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.

[8] M. E. Tipping, "Sparse kernel principal component analysis," In *Advances in Neural Information Processing Systems 13*, Cambridge, MA: MIT Press, pp. 633–639, 2001.

[9] 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*, pp. 109–116.

[10] R. Vollgraf, and K. Obermayer, "Sparse optimization for second order kernel methods," In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pp. 145–152, 2006.

[11] 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, pp. 309–312.

[12] Y. Xu, D. Zhang, J. Yang, J. Zhong, and J. 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.

[13] Z. Hussain, and J. Shawe-Taylor, "Theory of matching pursuit," In *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Vancouver, BC, Canada: MIT Press, pp. 721–728, 2009.

[14] 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.

[15] B. Schölkopf, and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press,

Cambridge, MA, 2001.

[16] 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, 2011.

[17] J. A. K. Suykens, T. V. Gestel, J. Vandewalle, and B. D. Moor, "A support vector machine formulation to pca analysis and its kernel version," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 447–450, 2003.

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

[19] V. Vapnik *Statistical Learning Theory* New York: Wiley, 1998.

[20] J. A. K. Suykens, and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[21] D. Wang, and T. Tanaka, "Sparse kernel principal component analysis based on elastic net regularization," In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 3703–3708, 2016.

[22] L. Guo, P. Wu, J. Gao, and S. Lou, "Sparse Kernel Principal Component Analysis via Sequential Approach for Nonlinear Process Monitoring," In *IEEE Access*, vol. 7, pp. 47550–47563, 2019.

[23] 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, 2012.

[24] 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*, pp. 4045–4049, 2015.

[25] 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.

[26] 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.

[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, J, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[28] K. Sjöstrand, L. Clemmensen, R. Larsen, and B. Ersbøll, "Spasm: A matlab toolbox for sparse statistical modeling," *Journal of Statistical Software*, pp. 1-24. 2012.

[29] T. Tanaka, Y. Washizawa, and A. Kuh, "Adaptive kernel principal components tracking," In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 1905–1908, 2012.

[30] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[31] P. Honeine, "Online kernel principal component analysis: A reduced-order model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1814–1826, 2012.

[32] M. Lichman, UCI machine learning repository.

Duo Wang received the B.S. degree from Henan University of Technology, China, in 2007, and the M.S. degree from the Northeasten University, China, in 2009. Currently, he is pursuing his PhD degree in Tokyo University of Agriculture and Technology, Japan. His research interests include kernel methods, outlier detection, signal processing and machine learning.

Toshihisa Tanaka received the B.E., the M.E., and the Ph.D. degrees from the Tokyo Institute of Technology in 1997, 2000, and 2002, respectively. From 2000 to 2002, he was a JSPS Research Fellow. From October 2002 to March 2004, he was a Research Scientist at RIKEN Brain Science Institute. In April 2004, he joined Department of Electrical and Electronic Engineering, the Tokyo University of Agriculture and Technology, where he is currently a Professor. In 2005, he was a Royal Society visiting fellow at the Communications and Signal Processing Group, Imperial College London, U.K. From June 2011 to October 2011, he was a visiting faculty member in Department of Electrical Engineering, the University of Hawaii at Manoa. His research interests include a broad area of signal processing and machine learning including brain and biomedical signal processing, brain-machine interfaces and adaptive systems. He is a co-editor of Signal Processing Techniques for Knowledge Extraction and Information Fusion (with Mandic, Springer), 2008 and a leading co-editor of Signal Processing and Machine Learning for Brain-Machine Interfaces (with Arvaneh, IET, UK), 2018. He served as an associate editor and a guest editor of special issues in journals including Neurocomputing and IEICE Transactions on Fundamentals and Computational Intelligence and Neuroscience (Hindawi). Currently he serves as an associate editor of IEEE Transactions on Neural Networks and Learning Systems, Applied Sciences (MDPI), and Advances in Data Science and Adaptive Analysis (World Scientific). Furthermore, he serves as a member-at-large, board of governors (BoG) of Asia-Pacific Signal and Information Processing Association (APSIPA). He was a chair of the Technical Committee on Biomedical Signal Processing, APSIPA. He is a senior member of IEEE, and a member of IEICE, APSIPA, and Society for Neuroscience.