Neural Comput & Applic
DOI 10.1007/s00521-011-0600-z

ORIGINAL ARTICLE

Sparse data-dependent kernel principal component analysis based on least squares support vector machine for feature extraction and recognition

Jun-Bao Li · Huijun Gao

Received: 4 February 2011 / Accepted: 15 April 2011

Abstract Kernel learning is widely used in many areas, and many methods are developed. As a famous kernel learning method, kernel principal component analysis (KPCA) endures two problems in the practical applications. One is that all training samples need to be stored for the computing the kernel matrix during kernel learning. Second is that the kernel and its parameter have the heavy influence on the performance of kernel learning. In order to solve the above problem, we present a novel kernel learning namely sparse data-dependent kernel principal component analysis through reducing the training samples with sparse learning-based least squares support vector machine and adaptive self-optimizing kernel structure according to the input training samples. Experimental results on UCI datasets, ORL and YALE face databases, and Wisconsin Breast Cancer database show that it is feasible to improve KPCA on saving consuming space and optimizing kernel structure.

Keywords Kernel method · Kernel principal component analysis · Sparse learning · Data-dependent kernel function · Feature extraction · Computation efficiency

J.-B. Li (✉)
Department of Automatic Test and Control,
Harbin Institute of Technology, 150001 Harbin, China
e-mail: junbaolihit@gmail.com

H. Gao
Department of Control Science and Engineering,
Harbin Institute of Technology, 150001 Harbin, China

1 Introduction

Feature extraction with dimensionality reduction is an important step and essential process in many data analysis areas, such as face recognition [1, 2], handwriting recognition [3], human facial expressions analysis [4], speech recognition [5], passive Components Dicing [6], text categorization [7]. Many feature extraction methods were developed in the past research works. Dimensionality reduction method is a most popular method for feature extraction. Linear dimensionality reduction aims to develop a meaningful low-dimensional subspace in a high-dimensional input space such as PCA and LDA [8]. LDA is to find the optimal projection matrix with Fisher criterion through considering the class labels, and PCA seeks to minimize the mean square error criterion. Linear discriminant analysis (LDA) has been widely used in many fields such as face recognition and character recognition. LDA works well in some cases, but it fails to capture a nonlinear relationship with a linear mapping. For this problem, kernel trick is used to represent complicated nonlinear relationships of input data. The kernel-based nonlinear feature extraction techniques have attracted much attention in the areas of pattern recognition and machine learning. Some algorithms using the kernel trick are developed in recent years, e.g., kernel principal component analysis (KPCA), kernel discriminant analysis (KDA) [8], and Support Vector Machine (SVM) [9]. KPCA was originally developed by Scholkopf et al. in 1998, while KDA was firstly proposed by Mika et al. in 1999. KDA has been applied in many real-world applications owing to its excellent performance on feature extraction. Researchers have developed a series of KDA algorithms (Juwei Lu [10], Baudat and Anouar [11], Liang and Shi [12], Yang [13], Wang [14] and Chen [15]). In particular, kernel principal

Published online: 03 May 2011

component analysis (KPCA) took the place of traditional linear PCA as the first feature extraction step in various research and applications. KPCA copes with nonlinear variations well. KPCA algorithm has been applied in pattern recognition areas, but high time-consuming is needed during training KPCA, but in the practical application, processing speed is a crucial problem such as face recognition. However, KPCA is to solve the eigenvalue problem with the number of samples plus the number of samples in the application. Kernel computations with all training samples are required to map a test sample to the subspace obtained by KPCA. In the classification process, KPCA computes kernel functions with all training samples, and the computational cost and memory required are high.

In this paper, we propose sparse data-dependent kernel principal component analysis (SDKPCA) based on least squares support vector machine for feature extraction and recognition. The advantage of SDKPCA compared with KPCA lies in computation efficiency and memory required for classification. The rest of this paper is organized as follows. Section 2 reviews and analyzes KPCA and KDA algorithm, and Sect. 3 presents sparse data-dependent principal component analysis based on least squares support vector machine mainly on theoretical derivations and algorithm procedure. Finally, Sect. 4 implements some experiments to evaluate the proposed algorithms on three datasets compared with the previous work. Conclusion is summarized in Sect. 5.

2 Related work

In this section, we review kernel principal component analysis and kernel discriminant analysis, and then we analyze the kernel principal component analysis.

2.1 Kernel principal component analysis

Kernel principal component analysis (KPCA) is the extension of principal component analysis (PCA) as the linear feature extraction. The main idea of KPCA is to project the input data from the linear space into the nonlinear space, and then implement PCA in the nonlinear feature space for feature extraction. For the clear description, we introduce the PCA as follows. Supposed that the training samples $x_1, x_2, \ldots, x_n$, then

$$C = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T \tag{1}$$

where $\bar{x}$ is the mean sample of all training samples, $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. The eigenvector corresponding to m largest eigenvalue $w_1, w_2, \ldots, w_m$ as the projection axis. Generally, these eigenvectors are calculated in many practical applications such as face recognition. SVD (Singular Value Decomposition) algorithm is applied into solving the singular matrix problem owing to the high-dimensional vector. Let $Q = [x_1 - \bar{x}, \ldots, x_n - \bar{x}]$, then $C = \frac{1}{n}QQ^T$. So $R = Q^TQ$ is the $n \times n$ positive definite matrix. For many applications, such as face recognition, the number of samples is less than the dimension. Accordingly, the dimension of R is less than the C. According to SVD, the eigenvectors $v_1, v_2, \ldots, v_m$ according to the m largest values ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$), the projection vectors are computed as follows.

$$w_j = \frac{1}{\sqrt{\lambda_j}}Qv_j, \quad j = 1, 2, \ldots, m \tag{2}$$

For any sample x, the jth feature is projected into the following feature through projection vector w_j as follows.

$$y_j = w_j^T x = \frac{1}{\sqrt{\lambda_j}}v_j^T Q^T x, \quad j = 1, 2, \ldots, m \tag{3}$$

The sample x, its PCA-based feature is $Y = [y_1, \ldots, y_m]^T$.

By introducing the kernel trick, PCA is extended into KPCA algorithm. The detail theoretical derivation is shown as follows.

$$C = \frac{1}{n}\sum_{i=1}^{n}(\Phi(x_i) - \bar{\Phi})(\Phi(x_i) - \bar{\Phi})^T \tag{4}$$

where $\bar{\Phi} = \frac{1}{n}\sum_{i=1}^{n}\Phi(x_i)$, and let

$$\tilde{C} = \frac{1}{n}\sum_{i=1}^{n}\Phi(x_i)\Phi(x_i)^T \tag{5}$$

Let $Q = [\Phi(x_1), \ldots, \Phi(x_n)]$, then $\tilde{C} = \frac{1}{n}QQ^T$. According to $\tilde{R} = Q^TQ$, with the kernel function, then

$$\tilde{R}_{ij} = \Phi(x_i)^T\Phi(x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j) \tag{6}$$

Compute the eigenvectors $u_1, u_2, \ldots, u_m$ according to the mth eigenvalue $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ of R, then $w_1, w_2, \ldots, w_m$ is calculated by

$$w_j = \frac{1}{\sqrt{\lambda_j}}Qu_j, \quad j = 1, 2, \ldots, m \tag{7}$$

Accordingly,

$$R = \hat{R} - 1_n\hat{R} - \hat{R}1_n + 1_n\hat{R}1_n \tag{8}$$

where $(1_n)_{ij} = 1/n$ $(i, j = 1, 2, \ldots, n)$, then

$$y_j = w_j^T x = \frac{1}{\sqrt{\lambda_j}}u_j^T[k(x_1, x), k(x_2, x), \ldots, k(x_n, x)] \tag{9}$$

PCA-based feature extraction needs to store $r \times m$ coefficient matrix W, where r is the number of principal components, and m is the number of training samples. While

KPCA-based feature extraction needs to store the original sample information owing to computing the kernel matrix with all training samples, which leads to a huge store space and a high computing consuming.

2.2 Kernel discriminant analysis

Kernel discriminant analysis is based on a conceptual transformation from the input space into a nonlinear high-dimensional feature space. Supposed that M training samples $\{x_1, x_2, \ldots, x_M\}$ with L class labels take values in an N-dimensional space $\mathbb{R}^N$, the data in $\mathbb{R}^N$ are mapped into a feature space F via the following nonlinear mapping, $\Phi : \mathbb{R}^N \to F, x \mapsto \Phi(x)$. Consequently in the feature space F, Fisher criterion is defined by

$$J(V) = \frac{V^T S_B^{\Phi} V}{V^T S_T^{\Phi} V} \tag{10}$$

where V is the discriminant vector, and S_B^{Φ} and S_T^{Φ} are the between classes scatter matrix and the total population scatter matrix, respectively. According to the Mercer kernel function theory, any solution V belongs to the span of all training pattern in $\mathbb{R}^N$. Hence, there exist coefficients c_p ($p = 1, 2, \ldots, M$) such that

$$V = \sum_{p=1}^{M} c_p \Phi(x_p) = \Psi\alpha \tag{11}$$

where $\Psi = [\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_M)]$ and $\alpha = [c_1, c_2, \ldots, c_M]^T$. Suppose the data are centered, Fisher criterion is transformed into

$$J(\alpha) = \frac{\alpha^T K G K \alpha}{\alpha^T K K \alpha} \tag{12}$$

where $G = \mathrm{diag}(G_1, G_2, \ldots, G_L)$, G_i is an $n_i \times n_i$ matrix whose elements are $\frac{1}{n_i}$, K is the kernel matrix calculated by a basic kernel $k(x, y)$. The criterion given in (12) attains its maximum for the orthonormal vectors. There are numerous algorithms to find this optimal subspace and an orthonormal basis for it.

2.3 Analysis on KPCA and KDA

In this section, we analyze the above kernel learning methods including kernel principal component analysis (KPCA) and kernel discriminant analysis (KDA) as follows.

Firstly, on KPCA, this nonlinearity is firstly mapping the data into another space using a nonlinear map, and then PCA is implemented using the mapped examples. The mapping and the space are determined implicitly by the choice of a kernel function that computes the dot product between two input examples mapped into feature space via kernel function. If kernel function is a positive definite kernel, then there exists a map into a dot product space. The space has the structure of a so-called Reproducing Kernel Hilbert Space (RKHS). The inner products in feature space can be evaluated without computing the nonlinear mapping explicitly. This allows us to work with a very high-dimensional, possibly infinite-dimensional RKHS. If a positive definite kernel is specified, we need to know neither the nonlinear mapping nor feature space explicitly to perform KPCA since only inner products are used in the computations. Commonly used examples of such positive definite kernel functions are the polynomial kernel and Gaussian kernel, each of them implying a different map and RKHS. PCA-based feature extraction needs to store the $r \times m$ coefficient matrix, where r is the number of principal components, and m is the number of training samples. While KPCA-based feature extraction needs to store the original sample information owing to computing the kernel matrix, which leads to a huge store and a high computing consuming. In order to solve the problem, we apply the least squares support vector machine to build the sparse KPCA.

Secondly, KDA is successfully to solve the nonlinear problem endured by linear discriminant analysis (LDA) as a traditional dimensionality reduction technique for feature extraction. In order to overcome this weakness of LDA, the kernel trick is used to represent the complicated nonlinear relationships of input data to develop kernel discriminant analysis (KDA) algorithm. Kernel-based nonlinear feature extraction techniques have attracted much attention in the areas of pattern recognition and machine learning. KDA has been applied in many real-world applications owing to its excellent performance on feature extraction.

Thirdly, both KDA and KPCA endure the kernel function and its parameters. Kernel function and its parameter have significant influence on feature extraction owing to the fact that the geometrical structure of the data in the kernel mapping space is determined totally by the kernel function. If an inappropriate kernel is used, the data points in the feature space may become worse. However, choosing the kernel parameters from a set of discrete values will not change the geometrical structures of the data in the kernel mapping space.

So, it is feasible to improve the performance of KPCA with sparse analysis and kernel optimization. In this paper, we reduce the training samples with sparse analysis and then optimize kernel structure with the reduced training samples.

3 Sparse data-dependent kernel principal component analysis

In this section, we present a novel learning called sparse data-dependent kernel principal component analysis

(SDKPCA) with the viewpoint of least squares support vector machine to solve the following problem. That is, the first is that all training samples need to be stored for the computing the kernel matrix during kernel learning, and the second is that the kernel and its parameter have the heavy influence on performance of kernel learning. We reduce the training samples with sparse analysis and then optimize kernel structure with the reduced training samples.

3.1 Reducing the training samples with sparse analysis

Firstly, we apply a least squares support vector machine formulation to KPCA which is interpreted as one class modeling problem with a target value equal to zero around which one maximizes the variance. Secondly, we introduce data-dependent kernel into sparse kernel principal component analysis, where the structure of the input data is adaptively changed with regard to the distribution of input data. Then, the objective function can be described as

$$\max_{w} \sum_{i=1}^{N} \left[0 - w^T\left(\phi(x_i) - u^{\phi}\right)\right]^2 \tag{13}$$

where $\phi : \mathbb{R}^N \to \mathbb{R}^l$ denotes the mapping to a high-dimensional feature space and $u^{\phi} = \left({}^{1}\!/\!_{N}\right) \sum_{i=1}^{N} \phi(x_i)$. The interpretation of the problem leads to the following optimization problem:

$$\max_{w,e} J(w,e) = -\frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2 \\ \text{subject to} \quad e_i = w^T\left(\phi(x_i) - u^{\phi}\right), \quad i = 1, 2, \ldots, N \tag{14}$$

We also apply the direct sparse kernel learning method to KPCA. Here, we also use the phase “expansion coefficients” and “expansion vectors”. Supposed a matrix $Z = \left[z_1, z_2, \ldots z_{N_z}\right]$, $Z \in \mathbb{R}^{N \times N_z}$, composed of N_z expansion vectors, and β_i (i = 1, 2, …, N_z) ($N_z < N$) are expansion coefficients, we modify the optimization problem to the following problem:

$$\max_{w,e} J(w,e) = -\frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2$$

$$\text{subject to} \quad e_i = w^T\left(\phi(x_i) - u^{\phi}\right), \quad i = 1, 2, \ldots, N \tag{15}$$

$$w = \sum_{i=1}^{N_z} \phi(z_i)\beta_i$$

where $\phi(Z) = \left[\phi(z_1), \phi(z_2), \ldots, \phi\left(z_{N_z}\right)\right]$. Now our goal is to solve the above optimization problem. We divide the above optimization problem into two steps: one is to find the optimal expansion vectors and expansion coefficients and second is to find the optimal projection matrix. Firstly, we reduce the above optimization problem, then we can obtain

$$\max_{Z,\beta,e} J(Z,\beta,e) = -\frac{1}{2} \left(\sum_{r=1}^{N_z} \phi(z_r)\beta_r\right)^T \left(\sum_{s=1}^{N_z} \phi(z_s)\beta_s\right) + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2 \\ \text{subject to} \quad e_i = \left(\sum_{r=1}^{N_z} \phi(z_r)\beta_r\right)^T \left(\phi(x_i) - u^{\phi}\right), \\ i = 1, 2, \ldots, N \tag{16}$$

where Z is variable. When Z is fixed, then

$$\max_{\beta,e} J(\beta,e) = -\frac{1}{2} \sum_{r=1}^{N_z} \sum_{s=1}^{N_z} \beta_s \beta_r \phi(z_r)^T \phi(z_s) + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2 \\ \text{subject to} \quad e_i = \left(\sum_{r=1}^{N_z} \beta_r \phi(z_r)^T\right) \left(\phi(x_i) - u^{\phi}\right), \\ i = 1, 2, \ldots, N \tag{17}$$

We apply the kernel function, that is, $k(x, y) = \langle \Phi(x), \Phi(y) \rangle$, given a random Z, then the above problem is same to the following problem.

$$W(Z) := \max_{\beta,e} -\frac{1}{2} \beta^T K_z \beta + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2 \\ \text{subject to} \quad e_i = \beta^T g(x_i), \quad i = 1, 2, \ldots, N \tag{18}$$

where $\beta = \left[\beta_1, \beta_2, \ldots \beta_{N_z}\right]^T$, and $g(x_i) = [k(z_1, x_i) - \frac{1}{N} \sum_{q=1}^{N} k(z_1, x_q) \cdots k(z_{N_z}, x_i) - \frac{1}{N} \sum_{q=1}^{N} k(z_{N_z}, x_q)]^T$, and $K_{ij}^{z} = k(z_i, z_j)$.

3.2 Solving the optimal projection matrix

After the optimal solution of data-dependent kernel is solved, the optimal kernel structure is achieved which is robust to the changing of the input data. After this step, the next step is to solve the equation of (18) to obtain the optimized sparse training samples with the so-called Lagrangian method. We define the Lagrangian as

$$L(\beta, e, \alpha) = -\frac{1}{2} \beta^T K_z \beta + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N} \alpha_i \left(e_i - \beta^T g(x_i)\right) \tag{19}$$

with the parameter α_i, i = 1, 2, …, N. The Lagrangian L must be maximized with respect to β, α_i, and e_i i = 1, 2, …, N, and the derivatives of L with respect to them must vanish, that is,

$$\begin{cases} \frac{\partial L}{\partial \beta} = 0 \to K_z \beta = \sum_{i=1}^{N} \alpha_i g(x_i) \\ \frac{\partial L}{\partial e_i} = 0 \to \alpha_i = \gamma e_i \\ \frac{\partial L}{\partial \alpha_i} = 0 \to e_i - \beta^T g(x_i) = 0 \end{cases} \tag{20}$$

Let $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$ $(\alpha_{N\times 1})$, and $G = [g(x_1), g(x_2), \ldots, g(x_N)]$ $\left(G_{N_z\times N}\right)$ and $E = [e_1, e_2, \ldots, e_N]^T$ $(E_{N\times 1})$, we can obtain

$$\begin{cases} K_z\beta = G\alpha \\ \alpha = \gamma E \\ E = G^T\beta \end{cases} \tag{21}$$

So, we can obtain $\beta = (K_z)^{-1}G\alpha$, then $E = G^T(K_z)^{-1}G\alpha$. It is easy to obtain the optimal solution α^z, which is an eigenvector of the $G^T(K_z)^{-1}G$ corresponding to the largest eigenvalue $\beta^z = (K_z)^{-1}G\alpha^z$. $W(Z)$ reaches the largest value when α^z is the eigenvector of $G^T(K_z)^{-1}G$ corresponding to the largest value, and $\beta^z = (K_z)^{-1}G\alpha^z$. The proof is described as follows.

Theorem 1 *$W(Z)$ reaches the largest value when α^z is the eigenvector of $G^T(K_z)^{-1}G$ corresponding to the largest value, and $\beta^z = (K_z)^{-1}G\alpha^z$.*

Proof. Firstly, let us reconsider the (24) as follows.

When $\lambda > 0$, then

$$\begin{aligned} -\frac{1}{2}\beta^T K_z\beta &= -\frac{1}{2}\left[\alpha^T G^T\left(\left(K_z\right)^{-1}\right)^T\right]K_z\left[\left(K_z\right)^{-1}G\alpha\right] \\ &= -\frac{1}{2}\left[\alpha^T G^T\left(K_z\right)^{-1}G\alpha\right] \\ &= -\frac{1}{2}\lambda\alpha^T\alpha \end{aligned} \tag{22}$$

Moreover, since $E = G^T(K_z)^{-1}G\alpha$, $G^T(K_z)^{-1}G\alpha = \lambda\alpha$, and $E = \lambda\alpha$, we obtain

$$\frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 = \frac{\gamma}{2}E^T E = \frac{\gamma}{2}\lambda^2\alpha^T\alpha \tag{23}$$

Since $\alpha^T\alpha = 1$, we can obtain

$$\begin{aligned} J(\beta, e) &= -\frac{1}{2}\beta^T K_z\beta + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 = -\frac{1}{2}\lambda\alpha^T\alpha + \frac{\gamma}{2}\lambda^2\alpha^T\alpha \\ &= \frac{1}{2}\lambda^2\left(\gamma - \frac{1}{\lambda}\right) \end{aligned} \tag{24}$$

From above the equation, we can see that $J(\beta, e)$ reaches the largest value when λ reaches the largest value. □

From above the equation, we can see that $J(\beta, e)$ reaches the largest value when λ reaches the largest value. Now our goal is to find the optimal Z that maximizes $W(Z) = -\frac{1}{2}(\beta^z)^T K_z(\beta^z) + \frac{\gamma}{2}(\beta^z)^T GG^T(\beta^z)$.

After we obtain Z*, and then compute the eigenvectorA $= [\alpha_1, \alpha_2, \ldots, \alpha_m]$ of $G^T(K_z)^{-1}G$ corresponding to the following eigen problem $G^T(K_z)^{-1}G\alpha = \lambda\alpha$, then

$$B = (K_z)^{-1}GA \tag{25}$$

3.3 Optimizing kernel structure with the reduced training samples

For given kernel, we introduce the data-dependent kernel with a general geometrical structure that can obtain the different kernel structure with different combination parameters, and the parameters are self-optimized under the criterions. Data-dependent kernel $k'(x, y)$ is described as

$$k'(x, y) = f(x)f(y)k(x, y) \tag{26}$$

where $f(x)$ is a positive real-valued function x, and $k(x, y)$ is a basic kernel, e.g., polynomial kernel and Gaussian kernel. Amari and Wu [16] expanded the spatial resolution in the margin of a SVM by using $f(x) = \sum_{i\in SV} a_i e^{-\delta\|x-\tilde{x}_i\|^2}$, where $\tilde{x}_i$ is the ith support vector, SV is a set of support vector, a_i is a positive number representing the contribution of $\tilde{x}_i$, and δ is a free parameter. We generalize Amari and Wu's method as

$$f(x) = b_0 + \sum_{n=1}^{N_z} b_n e(x, \tilde{x}_n) \tag{27}$$

where $e(x, \tilde{x}_n) = e^{-\delta\|x-\tilde{x}_n\|^2}$, δ is a free parameter, $\tilde{x}_n$ are called the "expansion vectors (XVs)," N_z is the number of XVs, and b_n $(n = 0, 1, 2, \ldots, N_z)$ are the "expansion coefficients" associated with $\tilde{x}_n$ $(n = 0, 1, 2, \ldots, N_z)$. The definition of the data-dependent kernel shows that the geometrical structure of the data in the kernel mapping space is determined by the expansion coefficients with the determinative XVs and free parameter. The objective function to find the adaptive expansion coefficients varied with the input data for the quasiconformal kernel. Given the free parameter δ and the expansion vectors $\{\tilde{x}_i\}_{i=1,2,\ldots,N_Z}$, we create the matrix

$$E = \begin{bmatrix} 1 & e(x_1, \tilde{x}_1) & \ldots & e(x_1, \tilde{x}_{N_z}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e(x_M, \tilde{x}_1) & \ldots & e(x_M, \tilde{x}_{N_z}) \end{bmatrix} \tag{28}$$

Let $\xi = \left[b_0, b_1, b_2, \ldots, b_{N_z}\right]^T (i = 0, 1, 2, \ldots, N_z)$ and $\Lambda = \mathrm{diag}(f(x_1), f(x_2), \ldots, f(x_m))$, the following equation is obtained

$$\Lambda 1_M = E\xi \tag{29}$$

where 1_M is a M-dimensional vector whose entries equal to unity. The expansion coefficient vector ξ is solved through optimizing an objective function designed for measuring the class separability of data in feature space with Fisher criterion and maximum margin criterion in our previous work [17]. In this paper, we apply maximum margin criterion to optimized kernel as follows. The main idea of optimizing kernel structure with the reduced training samples is to find the optimal data-dependent kernel parameter vector ξ through optimizing an objective function. The kernel optimization algorithm procedure is described as follows.

$$\begin{aligned} &\max \quad J_{\text{Fisher}} \\ &\text{subject to} \quad \xi^T\xi - 1 = 0 \end{aligned} \tag{30}$$

where $J_{\text{Fisher}} = \frac{\xi^T E^T B_0 E\xi}{\xi^T E^T W_0 E\xi}$, and let $J_1(\xi) = \xi^T E^T B_0 E\xi$ and $J_2(\xi) = \xi^T E^T W_0 E\xi$, then

$$\frac{\partial J_{\text{Fisher}}(\xi)}{\partial \xi} = \frac{2}{J_2^2}\left(J_2 E^T B_0 E - J_1 E^T W_0 E\right)\xi \tag{31}$$

The optimal solution is achieved with the iteration method, and then the optimal ξ is solved as follows.

$$\xi^{(n+1)} = \xi^{(n)} + \varepsilon\left(\frac{1}{J_2}E^T B_0 E - \frac{J_{\text{Fisher}}}{J_2}E^T W_0 E\right)\xi^{(n)} \tag{32}$$

where ε is the learning rate with its definition of $\varepsilon(n) = \varepsilon_0\left(1 - \frac{n}{N}\right)$, where ε_0 is the initialized learning rate, n and N are the current iteration number and the total iteration number, respectively.

3.4 Algorithm procedure

For a set of training sample set, first we optimize the kernel function $k'(x, y)$ with the given the basic kernel function $k(x, y)$, and then implement SKPCA.

$$y = B^T V_{zx} \tag{33}$$

where $g(z_i, x) = k'(z_i, x) - \frac{1}{N}\sum_{q=1}^{N} k'(z_i, x_q)$, $V_{zx} = [g(z_1, x)\ g(z_2, x)\ldots g(z_{N_z}, x)]^T$. Since $w = \sum_{i=1}^{N_z} \phi(z_i)\beta_i^z$, so

$$y = \sum_{i=1}^{N_z} \beta_i^z\left[\phi(z_i)^T(\phi(x) - u^{\phi})\right] \tag{34}$$

Let $\beta_z = \left[\beta_1^z\beta_2^z\ldots\beta_{N_z}^z\right]^T$. For we choose m eigenvector α corresponding to m largest eigenvalue. Let $P = \left[\left(\beta_z^T\right)_1\left(\beta_z^T\right)_2\ldots\left(\beta_z^T\right)_m\right]^T$, the feature can be obtained as follows.

$$z = \text{PK}_{zx} \tag{35}$$

As above discussed, from the theoretical viewpoints, sparse data-dependent kernel principal component analysis (SDKPCA) chooses adaptively a few of samples from the training sample set but a little influence on recognition performance, which saves much space of storing training samples on computing the kernel matrix with the lower time-consuming. So in the practical applications, SDKPCA can solve the limitation from KPCA owing to its high store space and time-consuming its ability on feature extraction. So from the theory viewpoint, SDKPCA is adaptive to the applications with the demand of the strict computation efficiency but not strict on recognition.

4 Experimental results

In this section, we implement some experiments to testify the feasibility and performance of sparse data-dependent kernel principal component analysis (SDKPCA) with the framework of experiment system as shown in Fig. 1. For comparison purpose, we implement sparse kernel principal component analysis (SKPCA) with the basic kernel under the same conditions.

4.1 Performance on UCI dataset

Firstly, we use the six UCI datasets popular widely in pattern recognition area to testify the performance of the proposed algorithm compared with the KPCA algorithm using the part of training samples and the whole size of samples. In the experiments, we randomly choose one hundred of training samples on each training sample set, especially 20 parts on Image and Splice dataset. In the experiments, we choose the Gaussian kernel with its parameters determined by the training samples. The experimental results are shown in Tables 1, 2 and 3, and the second column shows the error rate of each algorithm on the corresponding dataset. The third column shows the number of training samples in Table 1 and the number of training samples in the proposed algorithm in Table 2. And in the brackets denote the ratio between the number of training samples of KPCA with the common training method and the proposed training samples. The results show that the proposed algorithm achieves the similar recognition performance, but the proposed algorithm only use the less size of training set. For example, only 8% of training samples are used but only error rate of 2.8% is higher than the common methods. Since only small size of training samples are applied in the proposed algorithm, so it will save some place for storing and increase the computation efficiency for KPCA. As the experimental results in Tables 2 and 3 compared with Table 1, KPCA, sparse KPCA (SKPCA) and sparse data-dependent kernel principal component analysis (SDKPCA), have the similar recognition accuracy but with different number of training samples. SKPCA saves much space of storing training samples on computing the kernel matrix with the lower time-consuming, but achieves the similar recognition accuracy compared with KPCA. The comparison of the results in Tables 2 and 3 shows that SDKPCA achieves the

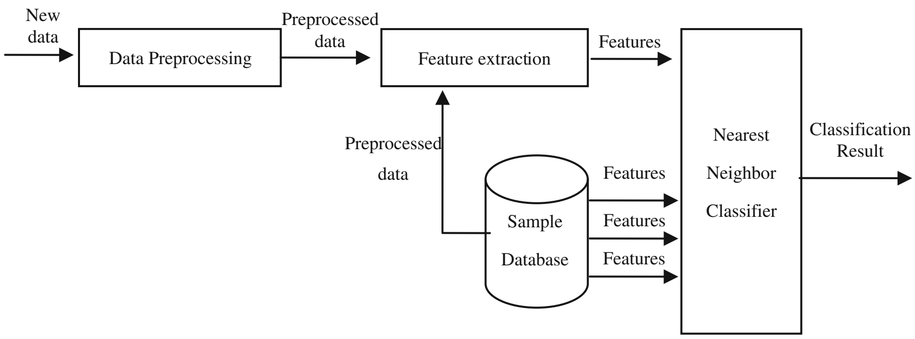

Fig. 1 A framework of a real image classification system

Table 1 Recognition performance of KPCA

Datasets	Error rate (%)	Training samples
Banana	13.6 ± 0.1	400
Image	4.8 ± 0.4	1,300
F.Solar	31.4 ± 2.1	666
Splice	8.6 ± 0.8	1,000
Thyroid	2.1 ± 1.0	140
Titanic	22.8 ± 0.3	150

Table 2 Recognition performance of SKPCA

Datasets	Error rate (%)	Training samples
Banana	14.2 ± 0.1	120 (30%)
Image	5.4 ± 0.3	180 (14%)
F.Solar	34.2 ± 2.3	50 (8%)
Splice	9.4 ± 0.9	280 (28%)
Thyroid	2.2 ± 1.3	30 (21%)
Titanic	23.2 ± 0.5	30 (20%)

Table 3 Recognition performance of SDKPCA

Datasets	Error rate (%)	Training samples
Banana	13.9 ± 0.2	120 (30%)
Image	5.1 ± 0.2	180 (14%)
F.Solar	32.8 ± 2.1	50 (8%)
Splice	9.0 ± 0.7	280 (28%)
Thyroid	2.2 ± 1.3	30 (21%)
Titanic	24.4 ± 0.4	30 (20%)

higher recognition accuracy than SKPCA owing to its kernel optimization combined with SKPCA. SDKPCA is adaptive to the applications with the demand of the strict computation efficiency but not strict on recognition. This set of experiments show that SDKPCA performs better than SKPCA with the same number of training samples, while SKPCA achieves the similar recognition accuracy but less number of training samples compared with traditional KPCA. The results testify the feasibility of SDKPCA and SKPCA.

4.2 Performance on ORL database

To quantitatively assess and fairly compare the methods, we evaluate the proposed scheme on ORL [18] and Yale [19] databases under the variable illumination conditions according to a standard testing procedure. ORL face database, developed at the Olivetti Research Laboratory, Cambridge, U.K., is composed of 400 grayscale images with 10 images for each of 40 individuals. The variations of the images are across pose, time, and facial expression. To reduce computation complexity, we resize the original ORL face images sized 112 × 92 pixels with a 256 gray scale to 48 × 48 pixels, and some examples are shown in Fig. 2. The experimental results are shown in Table 4, SDKPCA performs better than SKPCA under the same number of training samples.

4.3 Performance on Yale database

Also, we evaluate the proposed scheme on Yale [19] databases under the variable illumination conditions according to a standard testing procedure to quantitatively assess and fairly compare the methods. The Yale face database was constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. These images are taken under different lighting condition (left-light, center-light, and right-light), different facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. Similarly, the images from Yale databases are cropped to the size of 100 × 100 pixels, and some examples are shown in Fig. 2b.

We randomly choose one face image per person as the training sample, and the rest face images are to test the performance of proposed scheme. That is, the rest nine test samples are to test on ORL face database, while 10 test samples per person are to test the performance on Yale face database. The average recognition accuracy rate is to

Fig. 2 Example face images of ORL face databases used in our experiments

Table 4 Performance comparison on ORL face database

Algorithms	Error rate (%)	Training samples
KPCA	15.3 ± 0.8	200
SKPCA	18.4 ± 0.9	120 (60%)
SDKPCA	17.5 ± 0.7	120 (60%)

evaluate the performance of the recognition accuracy, and we implement the experiments for 10 times and 11 times for ORL and Yale face database, respectively. Some examples are shown in Fig. 3. As shown in Table 5, the experimental results show that SDKPCA performs better than SKPCA under the same number of training samples.

4.4 Performance on Wisconsin Breast Cancer database

We elevate the performance on Wisconsin Breast Cancer database [20] consisting of 569 instances including 357 benign samples and 212 malignant samples. And each one represents FNA test measurements for one diagnosis case. For this dataset, each instance has 32 attributes, where the first two attributes correspond to a unique identification number and the diagnosis status (benign or malignant). The rest 30 features are computations for ten real-valued features, along with their mean, standard error and the mean of the three largest values (“worst” value) for each cell nucleus, respectively. Figure 4 depicts two images, which were taken from fine needle biopsies of breast tumors. As shown in Table 6, the recognition accuracy 5.4 and 3.8% are achieved by the common training method and the proposed training method. But only 37% training samples are applied in the training procedure. As shown in the Table 6, only 37% training samples are used but only error rate 1.6% is higher than the common methods. Some storing space is saved, and high computation efficiency is achieved for the practical applications.

4.5 Discussion

The feasibility and performance are evaluated on UCI dataset, ORL, Yale, and Wisconsin Breast Cancer databases. The recognition performance of the proposed SDKPCA is enough testified and evaluated on the simulated and real databases. The comprehensive analysis on the experimental results of the proposed SDKPCA, SKPCA, and KPCA is shown as follows.

Firstly, KPCA in pattern recognition areas has some advantage of recognition performance but high time-consuming in training procedure in the practical application. For the high-dimensional data such as face recognition in ORL and Yale databases, the computation efficiency is a crucial problem owing to the eigenvalue problem endured by KPCA with a large number of training samples together with the kernel computations with all training samples. So

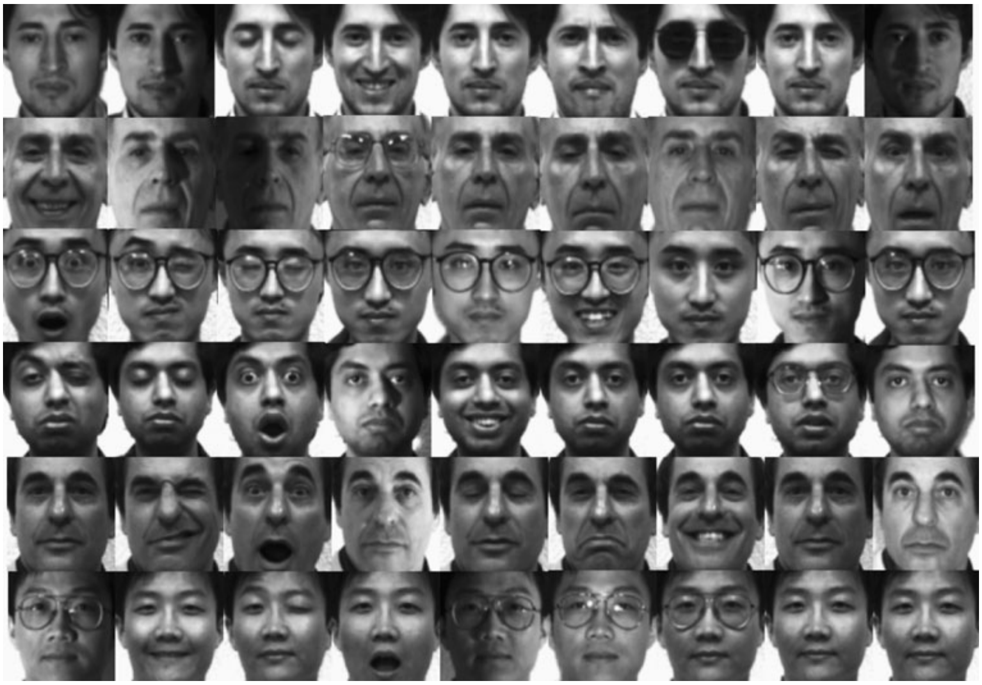

Fig. 3 Example face images of Yale face databases used in our experiments

Table 5 Performance comparison on Yale face database

Algorithms	Error rate (%)	Training samples
KPCA	17.8 ± 0.7	75
SKPCA	20.4 ± 0.8	45 (60%)
SDKPCA	18.7 ± 0.6	45 (60%)

it is feasible to improve KPCA with direct sparse analysis to develop sparse kernel principal component analysis (SKPCA) with the less number of training samples, which denotes that SKPCA is effective in the practical applications, where the store space and time-consuming are limited and only few training samples are used. SKPCA is very meaningful on not only accelerating the evaluation of the test data but also saving the memory of storing the trained data.

Secondly, under the condition of the limited training samples stored in the databases, in sparse data-dependent kernel principal component analysis (SDKPCA) algorithm, data-dependent kernel is applied to increase the recognition accuracy with the same training samples compared with SKPCA, where the kernel structure is adaptive to the input training samples. So it is feasible to improve the feasibility of SDKPCA on practical applications under the few training samples.

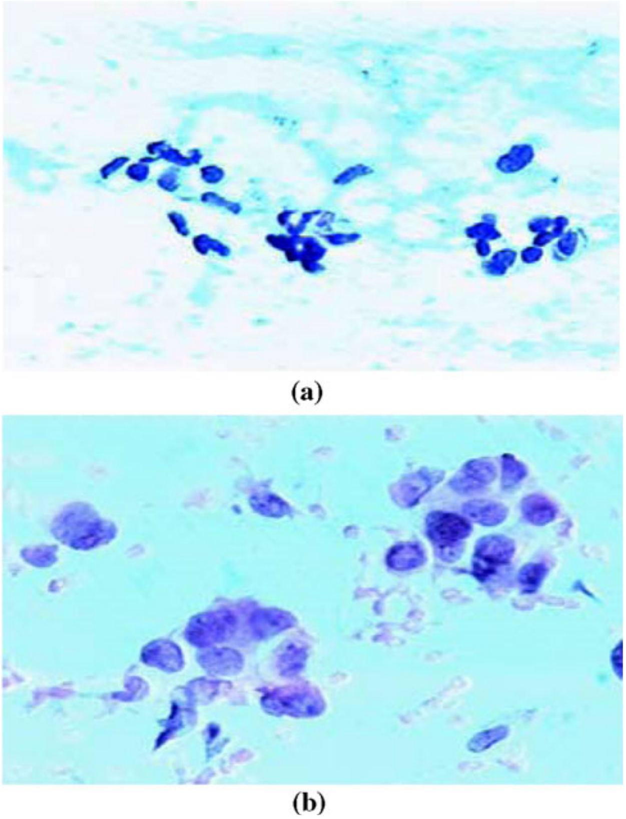

Fig. 4 Example images (**a**) Benign, (**b**) Malignant [7]

Finally, the choosing of kernel function and its parameter is a key impact factor on recognition performance on kernel learning. The adaptively parameter choosing of data-dependent kernel function can improve the recognition performance of kernel learning under the same condition of the same number of training and test samples. The feasibility

Table 6 Performance comparison on KPCA and RKPCA

Algorithms	Error rate (%)	Training samples
KPCA	3.8 ± 0.4	300
SKPCA	5.4 ± 0.3	110 (37%)
SDKPCA	4.9 ± 0.4	110 (37%)

of data-dependent kernel optimization is testified on the experimental results of comparing SDKPCA and SKPCA. So, kernel optimization is an effective way of improving recognition performance of kernel learning in the practical applications.

5 Conclusion

In this paper, we present a novel kernel learning namely sparse data-dependent kernel principal component analysis (SDKPCA) through reducing the training samples with sparse learning-based least squares support vector machine and adaptive self-optimizing kernel structure according to the input training samples. SDKPCA has solved two problems widely endured by kernel learning: one is that all training samples need to be stored for the computing the kernel matrix during kernel learning and second is that the kernel and its parameter have the heavy influence on performance of kernel learning. The experimental results testify the feasibility and effectiveness of the proposed algorithm on saving consuming space and optimizing kernel structure. The proposed SDKPCA algorithm has the potential applications in image classification, face recognition, and speech recognition.

Acknowledgments This work is supported by National Science Foundation of China under Grant No. 61001165, and Heilongjiang Provincial Natural Science Foundation of China (Grant No. QC2010066), and HIT Young Scholar Foundation of 985 Project.

References

1. Lee J-S, Lin S-F (2010) A hierarchical face recognition scheme. Int J Innov Comput Inf Control 6(12):5439–5450
2. Wei X, Zhou C, Zhang Q (2010) ICA-based features fusion for face recognition. Int J Innov Comput Inf Control 6(10): 4651–4661
3. Arora S, Bhattacharjee D, Nasipuri M, Basu DK, Kundu M (2011) Complementary features combined in a MLP-based system to recognize handwritten devnagari character. J Inf Hiding Multimed Signal Process 2(1):71–77
4. Krinidis S, Pitas I (2010) Statistical analysis of human facial expressions. J Inf Hiding Multimed Signal Process 1(3):241–260
5. Sayoud H, Ouamour S (2010) Speaker clustering of stereo audio documents based on sequential gathering process. J Inf Hiding Multimed Signal Process 1(4):344–360
6. Lin H-D, Peter Chiu Y-S (2010) RBF network and EPC method applied to automated process regulations for passive components dicing. Int J Innov Comput Inf Control 6(12):5077–5091
7. Tang H, Wu J, Lin Z, Lu M (2010) An enhanced AdaBoost algorithm with naive Bayesian text categorization based on a novel re-weighting strategy. Int J Innov Comput Inf Control 6(11):5299–5310
8. Yang J, Frangi AF, Yang J-Y, Zhang D, Jin Z (2005) KPCA plus LDA: a complete kernel fisher Discriminant framework for feature extraction and recognition. IEEE Trans Pattern Anal Mach Intell 27(2):230–244
9. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
10. Lu J, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans Neural Netw 14(1):117–226
11. Baudat G, Anouar F (2000) Generalized discriminant analysis using a kernel approach. Neural Comput 12(10):2385–2404
12. Liang Z, Shi P (2005) Uncorrelated discriminant vectors using a kernel method. Pattern Recognit 38:307–310
13. MH Yang (2002) Kernel Eigenfaces vs. Kernel Fisherfaces: face recognition using kernel methods. In: Proceedings of the fifth IEEE international conference automatic face and gesture recognition, pp 215–220
14. Wang L, Chan KL, Xue P (2005) A criterion for optimizing kernel parameters in KBDA for image retrieval. IEEE Trans Syst Man Cybern B Cybern 35(3):556–562
15. Chen W-S, Yuen PC, Huang J, Dai D-Q (2005) Kernel machine-based one-parameter regularized fisher discriminant method for face recognition. IEEE Trans Syst Man Cybern B Cybern 35(4):658–669
16. Amari S, Wu S (1999) Improving support vector machine classifiers by modifying kernel functions. Neural Network 12(6):783–789
17. J-B Li, Pan J-S, Lu Z-M (2009) Kernel optimization-based discriminant analysis for face recognition. Neural Comput Appl 18(6):603–612
18. Samaria F, Harter A (1994) Parameterisation of a stochastic model for human face identification. In: Proceedings of 2nd IEEE workshop on applications of computer vision, Sarasota, FL
19. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
20. Wolberg WH, Street WN, Heisey DM, Mangasarian OL (1995) Computer-derived nuclear features distinguish malignant from benign breast cytology. Human Pathol 26:792–796