Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **4**, July **2015**

Adaptive Sparse Kernel Principal Component Analysis for Computation and Store Space Constrained-based Feature Extraction

Xiao-Dan Xie*

College of Automation Science and Electrical Engineering
Beihang University, Beijing, China
xiexiaodanbeijing@126.com
*Corresponding author

Bo-Hu Li

College of Automation Science and Electrical Engineering
Beihang University, Beijing, China

Xu-Dong Chai

Beijing Simulation Center, Beijing, China

Received January 2015; revised March 2015

Abstract. *On the training samples number and kernel function and its parameter endured by Kernel Principal Component Analysis, this paper presents one-class support vector based Adaptive Sparse Kernel Principal Component Analysis (ASKPCA) 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. ASKPCA is adaptive to the computation and store space constrained applications, such as small size of hardware platform based image retrieval, medical assistant diagnosis system, and so on. Moreover, the few meaningful samples are found with solving the constraint optimization equation, and these training samples are used to compute the kernel matrix which decreases the computing time and saving space, and the algorithm is to increase the recognition accuracy and computing efficiency in the image processing under the hardware computing platform on the limit 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 and Sparse Kernel Principal Component Analysis (SKPCA) 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.

1. **Introduction.** Feature extraction with dimensionality reduction is an important step and essential process in many data analysis [1, 2]. Linear dimensionality reduction aims to develop a meaningful low dimensional subspace in a high-dimensional input space such as PCA and LDA [3]. 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 [1]. Some algorithms using the kernel trick are developed in recent years, e.g., Kernel Principal Component Analysis (KPCA) [6], Kernel Discriminant Analysis (KDA) [7] and Support Vector Machine (SVM) [8]. In the recent years, the multiple kernel learning method was developed for kernel-based learning. Researchers presented Generalized Multiple Kernel Learning (GMKL), and other multiple kernel learning, for example, EMKL(Elastic Multiple Kernel Learnin), GL-MKL(Group Lasso Regularized Multiple Kernel Learning), DTMKL(Domain Transfer Multiple Kernel Learning), NLMKL(nonlinear MKL) [16, 17]. 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 [9], Baudat and Anouar [10], Liang and Shi [11], Yang [12], Wang [13] and Chen [14]). In particular, Kernel Principal Component Analysis (KPCA) took the place of traditional linear PCA as the first feature extraction step in various research and applications. KPCA copes with non-linear 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. So sparse KPCA is very meaningful, and it not only accelerates the evaluation of the test data but also saves the memory of storing the trained data. In this paper, we build the sparse kernel component analysis and prove the feasibility of using the direct method for building sparse kernel principal component analysis from the theoretical derivation. From the analysis on the computation complexity and memory capacity of the algorithm, Sparse KPCA can save the store space and reduce the time consuming.

From the discussion on Kernel Principal Component Analysis and Kernel Discriminant Analysis, we discuss the following points. Firstly, on the saving place of the training samples, in 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 which 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). First, 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. Secondly 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. For the image classification, the dimension of image is $r \times m$. 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 need to store the original sample information owing to computing the kernel matrix, which leads to a huge store and a high computing consuming. On computation and store space constrained-based conditions, in the practical applications, such as small size of hardware platform based image retrieval, medical assistant diagnosis system, and so on. The system is to increase the recognition accuracy and computing efficiency in the image processing under the hardware computing platform on the limit training samples. And the system can decreases the computing time and saving space in the practical applications.

In order to solve the problem, we apply the least squares support vector machine to build the sparse KPCA. Secondly, on the choosing of kernel function and its parameters, kernel function and its parameter has 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 kernel principal component analysis 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.

2. **Problem Statement and Preliminaries.** In this section, we present a novel learning called Adaptive Sparse Kernel Principal Component Analysis (ASKPCA) 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.

Step 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 Analysis, where the structure of the input data is adaptively changed regard to the distribution of input data. Then, the objective function can be described as

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

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

$$\begin{array}{c} \max\limits_{w,e} J(w,e) = -\frac{1}{2} w^T w + \frac{\gamma}{2} \sum\limits_{i=1}^{N} e_i^2 \\ \text{subject to} \quad e_i = w^T \left(\phi(x_i) - u^{\phi}\right) \quad , \quad i = 1, 2, ..., N \end{array} . \tag{2}$$

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 $Z = [z_1, z_2, ..., z_{N_z}]$, $Z \in R^{N \times N_z}$, composed of N_z expansion vectors, and $\beta_i (i = 1, 2, \cdot, N_z)(N_z < N)$ are expansion coefficients, we modify the optimization problem to the following problem:

$$\begin{array}{c} \max\limits_{w,e} J(w,e) = -\frac{1}{2} w^T w + \frac{\gamma}{2} \sum\limits_{i=1}^{N} e_i^2 \\ \text{subject to} \quad e_i = w^T \left(\phi(x_i) - u^{\phi}\right) \quad , \quad i = 1, 2, ..., N \\ w = \sum\limits_{i=1}^{N_z} \phi(z_i) \beta_i \end{array} , \tag{3}$$

where $\phi(Z) = [\phi(z_1), \phi(z_2), ..., \phi(z_{N_z})]$. 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; second is to find the optimal projection matrix. Firstly we reduce the above optimization problem, then we can obtain

$$\begin{array}{l} \max\limits_{Z,\beta,e} J(Z,\beta,e) = -\frac{1}{2}\left(\sum\limits_{r=1}^{N_z}\phi(z_r)\beta_r\right)^T\left(\sum\limits_{s=1}^{N_z}\phi(z_s)\beta_s\right) + \frac{\gamma}{2}\sum\limits_{i=1}^{N} e_i^2 \\ \text{subject to} \quad e_i = \left(\sum\limits_{r=1}^{N_z}\phi(z_r)\beta_r\right)^T\left(\phi(x_i) - u^{\phi}\right) \quad , \quad i = 1,2,...,N \end{array} \tag{4}$$

where Z is variable. When Z is fixed, then

$$\begin{array}{l} \max\limits_{\beta,e} J(\beta,e) = -\frac{1}{2}\sum\limits_{r=1}^{N_z}\sum\limits_{s=1}^{N_z}\beta_s\beta_r\phi(z_r)^T\phi(z_s) + \frac{\gamma}{2}\sum\limits_{i=1}^{N} e_i^2 \\ \textit{subject to} \quad e_i = \left(\sum\limits_{r=1}^{N_z}\beta_r\phi(z_r)^T\right)\left(\phi(x_i) - u^{\phi}\right) \quad , \quad i = 1,2,...,N \end{array} . \tag{5}$$

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.

$$\begin{array}{l} W(Z) := \max\limits_{\beta,e} -\frac{1}{2}\beta^T K_z\beta + \frac{\gamma}{2}\sum\limits_{i=1}^{N} e_i^2 \\ \textit{subject to} \quad e_i = \beta^T g(x_i) \quad , \quad i = 1,2,...,N. \end{array} \tag{6}$$

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

Step 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 to obtain the optimized sparse training samples with the so-called Lagrangian method. We define the Lagrangian as

$$L(\beta,e,\alpha) = -\frac{1}{2}\beta^T K_z\beta + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N}\alpha_i\left(e_i - \beta^T g(x_i)\right) \tag{7}$$

with the parameter $\alpha_i, i = 1,2,\cdots,N$. The Lagrangian L must be maximized with respect to β, α_i, and $e_i, i = 1,2,\cdots,N$, and the derivatives of L with respect to them must vanish, that is, $\frac{\partial L}{\partial\beta} = 0, \frac{\partial L}{\partial e_i} = 0, \frac{\partial L}{\partial\alpha_i} = 0$, so $K_z\beta = \sum\limits_{i=1}^{N}\alpha_i g(x_i), \alpha_i = \gamma e_i, e_i - \beta^T g(x_i) = 0$. Let $\alpha = [\alpha_1, \alpha_2, ..., \alpha_N]^T$ $(\alpha_{N\times 1})$, and $G = [g(x_1), g(x_2), ..., g(x_N)]$ $(G_{N_z\times N})$ and $E = [e_1, e_2, ..., e_N]^T$ $(E_{N\times 1})$, we can obtain $K_z\beta = G\alpha, \alpha = \gamma E, E = G^T\beta$. 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 $\beta^z = (K_z)^{-1}G\alpha^z$. 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 eigenvector $A = [\alpha_1, \alpha_2, ..., \alpha_m]$ of $G^T(K_z)^{-1}G$ corresponding to the following eigen problem $G^T(K_z)^{-1}G\alpha = \lambda\alpha$, then

$$B = (K_z)^{-1}GA. \tag{8}$$

Step 3. Optimizing kernel structure with the reduced training samples

For given kernel, we introduce the data-dependent kernel with a general geometrical structure 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

$$k'(x, y) = f(x)f(y)k(x, y), \tag{9}$$

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 [18] expanded the spatial resolution in the margin of a SVM by using $f(x) = \sum_{i \in SV} a_i e^{-\delta \|x - \widetilde{x}_i\|^2}$, where $\widetilde{x}_i$ is the ith support vector, and SV is a set of support vector, and α_i is a positive number representing the contribution of $\widetilde{x}_i$, and δ is a free parameter. We generalize Amari and Wu's method as

$$f(x) = b_0 + \sum_{n=1}^{N_{XV}} b_n e(x, \widetilde{x}_n), \tag{10}$$

where $e(x, \widetilde{x}_n) = e^{-\delta \|x - \widetilde{x}_n\|^2}$, and δ is a free parameter, and $\widetilde{x_n}$ are called the "expansion vectors (XVs)", and N_{XV} is the number of XVs, and $b_n (n = 0, 1, 2, \cdots, N_{XV_s})$ are the "expansion coefficients" associated with $\widetilde{x_n} (n = 0, 1, 2, \cdots, N_{XV_s})$. 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 $\{\widetilde{x}_i\}_{i=1,2,\ldots,N_{XVs}}$, we create the matrix

$$E = \begin{bmatrix} 1 & e(x_1, \widetilde{x}_1) & \cdots & e(x_1, \widetilde{x}_{N_{XVs}}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e(x_M, \widetilde{x}_1) & \cdots & e(x_M, \widetilde{x}_{N_{XVs}}) \end{bmatrix}. \tag{11}$$

Let $\beta = [b_0, b_1, b_2, \ldots, b_{N_{XVs}}]^T (i = 0, 1, 2, \cdots, N_{XV_s})$ and $\Lambda = diag(f(x_1), f(x_2), \ldots, f(x_M))$, the following equation is obtained

$$\Lambda 1_M = E\beta, \tag{12}$$

where 1_M is a M-dimensional vector whose entries equal to unity. The expansion coefficient vector β 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 [15].

Step 4. Feature exaction

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 sparse KPCA, That is, $W = B'$, $g(z_i, x) = k'(z_i, x) - \frac{1}{N} \sum_{q=1}^{N} k'(z_i, x_q)$, $V_{zx} = \begin{bmatrix} g(z_1, x) & g(z_2, x) & \ldots & g(z_{N_z}, x) \end{bmatrix}^T$, we can obtain,

$$y = WV_{zx} \tag{13}$$

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

Table 1. Recognition Performance of KPCA

Datasets	Number of Training samples		Recognition Error(%)		
	KPCA	SKPCA/ASKPCA	KPCA	SKPCA	ASKPCA
Banana	400	120	13.6	14.2	13.9
Image	1300	180	4.8	5.4	5.1
F.Solar	666	50	31.4	34.2	32.8
Splice	1000	280	8.6	9.4	9.0
Thyroid	140	30	2.1	2.2	2.2
Titanic	150	30	22.8	23.2	24.4

$$y = \sum_{i=1}^{N_z} \beta_i^z \left[\phi(z_i)^T \left(\phi(x) - u^\phi \right) \right]. \tag{14}$$

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

$$z = BK_{zx}. \tag{15}$$

As above discussion from the theoretical viewpoints, Adaptive Sparse Kernel Principal Component Analysis (ASKPCA) chooses adaptively a few of samples from the training sample set but 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, ASKPCA 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, ASKPCA is adaptive to the applications with the demand of the strict computation efficiency but not strict on recognition.

3. **Experimental results.** In this section, we implement some experiments to testify the feasibility and performance of Adaptive Sparse Kernel Principal Component Analysis (ASKPCA) on UCI datasets, ORL, Yale, Wisconsin Breast Cancer database. For comparison purpose, we implement Sparse Kernel Principal Component Analysis (SKPCA) with the basic kernel under the same conditions. The number of training samples used in the experiments is equal to the computing stress on computation and store space constrained-based conditions in the practical applications. The system is to increase the recognition accuracy and computing efficiency in the image processing under the hardware computing platform on the limit training samples, the limited training sample number is to decrease the computing time and saving space in the practical applications.

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 the 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 Table 1. The first column and second column are the number of training samples for KPCA, SKPCA, and ASKPCA respectively. 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% training samples are used but only error rate 2.8% higher than the common methods. Since only small size of training

Table 2. Performance comparison on ORL face database

Algorithms	Error rate (%)	Training samples
KPCA	15.3	200
SKPCA	18.4	120(60%)
ASKPCA	17.5	120(60%)

Table 3. Performance comparison on Yale face database

Algorithms	Error rate (%)	Training samples
KPCA	17.8	75
SKPCA	20.4	45(60%)
ASKPCA	18.7	45(60%)

samples are applied in the proposed algorithm, so it will save some place for storing and increase the computation efficiency for KPCA.

Secondly, we implement the algorithms on ORL database. To quantitatively assess and fairly compare the methods, we evaluate the proposed scheme on ORL [5] and Yale [3] 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. So, the dimensions of the samples are 48×48 and 100×100 for Yale and ORL databases. The experimental results are shown in Table 2, ASKPCA performs better than SKPCA under the same number of training samples.

Thirdly, we evaluate the algorithm on Yale face database. Also, we evaluate the proposed scheme on Yale [3] 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), and 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 pixels. 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 9 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 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. As shown in Table 3, the experimental results show that ASKPCA performs better than SKPCA under the same number of training samples.

Finally, we elevate the performance on Wisconsin Breast Cancer database [4] 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. As shown in Table 4, the recognition accuracy 5.4% and 3.8% are achieved by the common training

Table 4. Performance comparison on Yale face database

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

method and the proposed training method. But only 37% training samples are applied in the training procedure. As shown in the Table 4, only 37% training samples are used but only error rate 1.6% higher than the common methods. Some storing space is saved and high computation efficiency is achieved for the practical applications.

As shown the above experimental results, on KPCA, Sparse KPCA (SKPCA) and Adaptive Sparse Kernel Principal Component Analysis (ASKPCA) algorithm, 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. In order to increase the recognition accuracy under the same training samples with SKPCA, ASKPCA achieves a higher recognition accuracy than SKPCA because it uses kernel optimization procedure combined with SKPCA. So, ASKPCA is adaptive to the applications with the demand of the strict computation efficiency but not strict on recognition.

4. **Conclusions.** In this paper, we present a novel kernel learning namely Adaptive Sparse Kernel Principal Component Analysis (ASKPCA) 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. ASKPCA 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 ASKPCA algorithm has the potential applications in image classification, face recognition, and speech recognition.

REFERENCES

[1] C. Chen, J. Zhang, and R. Fleischer, Distance approximating dimension reduction of Riemannian manifolds, *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 40, No. 1, pp. 208-217, 2010.

[2] T. Zhang, K. Huang, X. Li, J. Yang, and D. Tao, Discriminative orthogonal neighborhood-preserving projections for classification, *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 40, No. 1, pp. 253-263, 2010.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 711-720, 1997.

[4] W. H. Wolberg, W. N. Street, D. M. Heisey, and O. L. Mangasarian , Computer-derived nuclear features distinguish malignant from benign breast cytology, *Human Pathology*, Vol.26, pp.792-796, 1995.

[5] F. Samaria and A. Harter, Parameterisation of a Stochastic Model for Human Face Identification, *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision, Sarasota FL*, December 1994.

[6] J. Yang, A. F. Frangi, J. V. Yang, D. Zhang, and Z. Jin, KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 230-244, Feb.2005.

[7] Q. S. Liu, H. Q. Lu, and S. D. Ma, Improving kernel Fisher discriminant analysis for face recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 1, pp. 42-49, Jan. 2004.

[8] V. Vapnik, *The Nature of Statistical Learning Theory*, New York, Springer, 1995.

[9] J. W. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, Face recognition using kernel direct discriminant analysis algorithms, *IEEE Trans. on Neural Networks*, vol. 14, no. 1, pp.117-226, 2003.

[10] G. Baudat and F. Anouar, Generalized Discriminant Analysis Using a Kernel Approach, *Neural Computation*, vol. 12, no. 10, pp. 2385-2404, 2000.

[11] Z. Z. Liang and P. F. Shi, Uncorrelated discriminant vectors using a kernel method, *Pattern Recognition*, vol. 38, pp. 307-310, 2005.

[12] M. H. Yang, Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods, *Proc.Fifth IEEE Intl Conf. Automatic Face and Gesture Recognition*, pp. 215-220, May 2002.

[13] L. Wang, K. L. Chan, and P. Xue, A Criterion for Optimizing Kernel Parameters in KBDA for Image Retrieval, *IEEE Trans. Systems, Man and Cybernetics-Part B: Cybernetics*, vol. 35, no. 3, pp. 556-562, June 2005.

[14] W. S. Chen, P. C. Yuen, J. Huang, and D. Q. Dai, Kernel Machine-Based One-Parameter Regularized Fisher Discriminant Method for Face Recognition, *IEEE Trans. Systems, Man and Cybernetics-Part B: Cybernetics*, vol. 35, no. 4, pp. 658-669, August 2005.

[15] J. B. Li, J. S. Pan and Z. M. Lu, Kernel Optimization-Based Discriminant Analysis for Face Recognition. *Neural Computing and Applications*. Vol. 18, No. 6, pp. 603-612, 2009.

[16] Y. Y. Lin, T. L. Liu, C. S. Fuh, Multiple Kernel Learning for Dimensionality Reduction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1147-1160, 2011.

[17] Y. R. Yeh, A Novel Multiple Kernel Learning Framework for Heterogeneous Feature Fusion and Variable Selection. *Multimedia*, vol.14, pp.563-574, 2012.