

## Research Article

# Supervised Kernel Optimized Locality Preserving Projection with Its Application to Face Recognition and Palm Biometrics

Chuang Lin,<sup>1</sup> Jifeng Jiang,<sup>1</sup> Xuefeng Zhao,<sup>2</sup> Meng Pang,<sup>1</sup> and Yanchun Ma<sup>3</sup>

<sup>1</sup>School of Software, Dalian University of Technology, Dalian 116600, China

<sup>2</sup>School of Civil Engineering, Dalian University of Technology, Dalian 116024, China

<sup>3</sup>Heilongjiang University of Traditional Chinese Medicine, Harbin 150040, China

Correspondence should be addressed to Xuefeng Zhao; zhaoxf@dlut.edu.cn

Received 24 January 2015; Revised 12 May 2015; Accepted 15 June 2015

Academic Editor: Bo Shen

Copyright © 2015 Chuang Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Kernel Locality Preserving Projection (KLPP) algorithm can effectively preserve the neighborhood structure of the database using the kernel trick. We have known that supervised KLPP (SKLPP) can preserve within-class geometric structures by using label information. However, the conventional SKLPP algorithm endures the kernel selection which has significant impact on the performances of SKLPP. In order to overcome this limitation, a method named supervised kernel optimized LPP (SKOLPP) is proposed in this paper, which can maximize the class separability in kernel learning. The proposed method maps the data from the original space to a higher dimensional kernel space using a data-dependent kernel. The adaptive parameters of the data-dependent kernel are automatically calculated through optimizing an objective function. Consequently, the nonlinear features extracted by SKOLPP have larger discriminative ability compared with SKLPP and are more adaptive to the input data. Experimental results on ORL, Yale, AR, and Palmprint databases showed the effectiveness of the proposed method.

## 1. Introduction

In recent years, the kernel methods have been widely studied for feature extraction and pattern recognition. They map the input data onto a kernel space and assume the nonlinear problem can be transferred to linear problem, which can be conveniently solved by linear algorithms. However, different kernel geometrical structures give different class discriminations, and the inappropriate selection of kernel function will induce disastrous effects, because the kernel matrix determines the geometrical structure of the mapped data in the kernel space. Thus it is necessary to use an adaptively optimized kernel function to improve the classification performance. We know that optimizing kernel parameters cannot change the geometrical structures of kernel in the feature space [1, 2], so, Schölkopf et al. [3] proposed an empirical kernel map which maps original input data onto a subspace of the empirical feature space, since the training data have the same geometrical structure in both the empirical feature space and the kernel space, and the former is easier to access than the latter. Cristianini et al. [4] and Lanckriet et al. [5], respectively,

employed the alignment and margin as the measure of data separation to evaluate the adaptability of a kernel to input data. He and Niyogi [6] pointed out that locality preserving projection (LPP) is capable of discovering nonlinear method with kernel trick. To utilize the merit of LPP, Wang and Lin [7] proposed supervised kernel LPP (SKLPP) to improve kernel LPP (KLPP), by using classification information in kernel feature extraction process. Li et al. [8] extended LPP with nonparametric similarity measure and then optimized the kernel with maximum margin criterion for feature extraction and recognition. Sun and Zhao [9] proposed a normalized Laplacian based optimal LPP method. Lu et al. [10] proposed a regularized generalized discriminant LPP approach. Lu and Tan [11] proposed a parametric regularized LPP. Pang and Yuan [12] proposed to substitute L2-norm with L1-norm to improve the robustness of LPP against outliers. Although SKLPP showed good performance in [7], the selection of kernel function has a significant influence on the kernel feature extraction and the problem was widely studied in the previous works [7, 13–15]. In [15], we proposed a Kernel Optimized PCA (KOPCA) with sparse representation-based

classifier (SRC). Although both KOPCA and SKOLPP aim to improve the recognition rate through optimizing the kernel function, they employed different feature extraction methods where KOPCA extracted features by PCA and SKOLPP extracted features by LPP. In [16], we proposed a Supervised Gabor-wavelet-based Kernel Locality Preserving Projections (SGKLPP) method, which integrated the Gabor-wavelet representation of face images and the SKLPP method to improve the recognition rate. Gabor wavelets extracted the features brought by illumination and facial expression changes, and SKLPP solved the nonlinear feature extraction and classification problem.

In [14], Pan et al. applied the optimizing kernel [17] to kernel discriminant analysis (KDA) called adaptive quasiconformal kernel discriminant analysis (AQKDA). Different from optimizing kernel based on Fisher Criterion [18], the maximum margin criterion (MMC) was chosen to extract the feature by maximizing the average margin between different classes of data in the quasiconformal kernel mapping space.

Li et al. proposed class-wise locality preserving projection (CLPP) which utilized the class information for feature extraction [19]. In CLPP, a nonparametric similarity measure of LPP was proposed by Li et al. and then the optimized kernel with maximum margin criterion was used for feature extraction. According to the nonparametric similarity measure, the local structure of the original data was constructed which took consideration of both the local information and the class label information. Moreover, Li et al. applied the kernel trick to CLPP to increase its performance on nonlinear feature extraction.

In [8], Li et al. proposed the Kernel Self-optimized Locality Preserving Discriminant Analysis (KSLPDA). In the paper, the authors integrated CLPP [19] and data-dependent kernel based MMC [14] to form a constraint optimization equation for KSLPDA.

In [20], Li et al. proposed the Quasiconformal Kernel Common Locality Discriminant Analysis (QKCLDA). In QKCLDA, the quasiconformal kernel based on Fisher Criterion is used for breast cancer diagnoses. Li et al. divided the procedure of QKCLDA into two steps. First, the original data was mapped to a low-dimensional space via quasiconformal kernel locality projection. Secondly, the low-dimensional data was mapped to a common space.

In SKOLPP, we first construct a data-dependent kernel [18] to maximize the class separability based Fisher Criterion. Then, we use gradient descent method to optimize the object function where the combination coefficients can be obtained. Last, integrating the supervised kernel locality preserving projections [21], optimized kernel LPP can be used to extract features. In this paper, SKOLPP aims to optimize the kernel function. By retaining the local information and optimizing the kernel function by maximizing the between-class distance, SKOLPP surpassed the above methods.

The paper is organized as follows. In Section 2, we optimize the kernel in the empirical feature space by seeking the optimal combination of coefficients with data-dependent kernel based on Fisher Criterion. In Section 3, we employ the optimized kernel function mentioned above to construct the supervised kernel optimized LPP (SKOLPP). Finally, in

Section 4, experiments are executed on ORL, Yale, AR, and Palmprint databases to demonstrate the effectiveness of the optimized kernel in classification.

## 2. Kernel Optimization in the Empirical Feature Space

*2.1. Data-Dependent Kernel.* The geometrical structure of the data in the feature space is determined by the kernel functions, which means that choosing different kernels may induce different class discrimination performance [4]. Because there is no general kernel function that can be suitable to all databases, it is necessary to choose a data-dependent kernel to solve this problem. In this paper, a data-dependent kernel which is similar to that used in [13] is employed as the objective kernel to be optimized.

Considering a set of training data  $x_1, \dots, x_N \in \mathbf{R}^d$ , we apply the conformal transformation kernel [13] as our data-dependent kernel function:

$$k(x, y) = q(x)q(y)k_0(x, y), \quad (1)$$

where  $x, y \in \mathbf{R}^d$  and  $k_0(x, y)$  is an ordinary kernel function, called the basic kernel.  $q(\cdot)$  is the factor function which is determinative for the properties of data-dependent kernel:

$$q(x) = \alpha_0 + \sum_{i=1}^n \alpha_i k_1(x, a_i), \quad (2)$$

where  $k_1(x, a_i) = \exp(-\gamma\|x - a_i\|)$ ,  $a_i \in \mathbf{R}^d$ ,  $\gamma$  is a free parameter, and  $\alpha_i \{i = 0, 1, \dots, n\}$  are the combination coefficients. The set  $\{a_i, i = 1, 2, \dots, n\}$  is called the ‘‘empirical cores’’ which can be chosen from the training data. In [13], in order to enlarge the spatial resolution around the class boundary, the support vectors are chosen as the empirical cores. In this paper, we choose the mean value of each class as empirical core. Apparently, the data-dependent kernel satisfies the Mercer condition [3].

Supposing that  $Q = \text{diag}(q(x_1), q(x_2), \dots, q(x_N))$ , then we have

$$K = QK_0Q, \quad (3)$$

where  $K_0 = [k_0(x_i, x_j)]_{N \times N}$  is the basic kernel and  $K$  is the data-dependent kernel. Letting  $\alpha = [\alpha_0, \alpha_1, \dots, \alpha_n]^T$ , then we have

$$Q1_n = E\alpha, \quad (4)$$

where  $E = [1 \ k_1(x_1, a_1) \ \dots \ k_1(x_1, a_n)]_{N \times (n+1)}$ .

*2.2. Fisher Criterion Based Kernel Optimization.* In [18], we note that the geometrical structure of data in the kernel feature space and empirical feature space is the same. That is, the optimized kernel parameters cannot change the geometrical structures of kernel in the feature space. It is better to measure class separability in the empirical feature space, because it is easier to access the empirical feature space than the kernel

feature space. Specifically, we use the Fisher Criteria for measuring the class separability:

$$J = \frac{\text{tr } S_b}{\text{tr } S_w}, \quad (5)$$

where  $J$  is the well-known Fisher scalar,  $\text{tr}$  denotes the trace of given matrix,  $S_b$  is the between-class scatter matrix, and  $S_w$  is the within-class scatter matrix.  $J$  measures the class separability in the feature space rather than in the projection subspace.  $J$  is a good choice for the task of kernel optimization as it is independent of the projections. So optimizing the data-dependent kernel means maximizing Fisher scalar  $J$ .

We can call the matrices  $B$  and  $W$  between-class and within-class kernel scatter matrices, respectively. Then, they can be written as

$$B = \text{diag} \left( \frac{1}{N_1} K_{11}, \dots, \frac{1}{N_C} K_{CC} \right) - \frac{1}{N} K, \quad (6)$$

$$W = \text{diag} (k_{11}, k_{22}, \dots, k_{NN}) - \text{diag} \left( \frac{1}{N_1} K_{11}, \dots, \frac{1}{N_C} K_{CC} \right), \quad (7)$$

where  $K$  is the data-dependent kernel and  $K_{ij}$  denotes the submatrix of the kernel matrix  $K$ . Apparently,  $K_{ii}$  is the kernel matrix corresponding to the samples in class  $i$ .

For the basic kernel  $K_0$ , the matrices  $B_0$  and  $W_0$  are similar to formulae (6) and (7). Now, the relationship between Fisher scalar  $J$  and the kernel scatter matrices can be established:

$$J(\alpha) = \frac{\mathbf{q}^T B_0 \mathbf{q}}{\mathbf{q}^T W_0 \mathbf{q}}. \quad (8)$$

The proof is given in the Appendix.

We use the standard gradient approach to maximize  $J(\alpha)$  and let  $J_1 = \mathbf{q}^T B_0 \mathbf{q} = \alpha^T E^T B_0 E \alpha$  and  $J_2 = \mathbf{q}^T W_0 \mathbf{q} = \alpha^T E^T W_0 E \alpha$ . Then, we have  $\partial J_1 / \partial \alpha = 2K_1^T B_0 K_1 \alpha$  and  $\partial J_2 / \partial \alpha = 2K_1^T W_0 K_1 \alpha$ .

Thus, the iteration algorithm is as follows:

$$\frac{\partial J}{\partial \alpha} = \frac{2}{J_2} (E^T B_0 E - E^T W_0 E) \alpha. \quad (9)$$

In order to maximize  $J$ , we let  $\partial J / \partial \alpha = 0$ , and then  $E^T B_0 E \alpha = E^T W_0 E \alpha$ .

However, the number of training samples is not enough in real-world applications. Thus, it is hard to get the invertible matrix of  $W_0$ . Additionally, we use the general gradient descent method to get  $\alpha$  which is an approximate value. Then, the updating equation for maximizing the class separability  $J$  is given by

$$\alpha^{(n+1)} = \alpha^{(n)} + \eta \left( \frac{1}{J_2} E^T B_0 E - \frac{1}{J_2} E^T W_0 E \right) \alpha^{(n)}, \quad (10)$$

where  $\eta$  is the learning rate and  $\eta(n) = \eta_0 (1 - n/N)$ , where  $N$  is the number of iterations,  $n$  denotes the current iteration number, and  $\eta_0$  is the initial learning rate.

When we get  $\alpha$ ,  $Q$  can be calculated by  $Q1_n = E\alpha$  and the data-dependent kernel  $K$  is easy to achieve.

### 3. Locality Preserving Projections and Supervised Kernel Optimized LPP

In this section, first, LPP algorithm is reviewed briefly, and then we use the optimized kernel function mentioned above to construct the supervised optimizing kernel LPP.

**3.1. Locality Preserving Projections.** Locality Preserving Projections (LPP) [6] is a linear manifold learning method which seeks an embedding that retains local information and obtains a face subspace that best perceives the crucial face manifold structure [22].

Given a matrix  $X = [x_1, x_2, \dots, x_N]^T \in R^{n \times N}$  with each point  $x_i \in R^{n \times 1}$ . Similar to other subspace learning algorithms, LPP uses the obtained transformation matrix  $W = [w_1, w_2, \dots, w_r] \in R^{n \times r}$  with the basis vector  $w_i \in R^{n \times 1}$  to map the high-dimensional points  $x \in R^{n \times 1}$  to low-dimensional points  $y \in R^{r \times 1}$ :  $y = W^T x$ . The objective function of LPP is defined as follows to compute the optimal basis vector  $w \in R^{n \times 1}$ :

$$\min \sum_{ij} \|w^T x_i - w^T x_j\|^2 S_{ij}, \quad (11)$$

where  $S_{ij}$  measures the similarity of  $x_i$  and  $x_j$ . Heat kernel is frequently used to define  $S_{ij}$ :

$$S_{ij} = \exp \left( -\frac{\|x_i - x_j\|^2}{t} \right), \quad (12)$$

where parameter  $t \in R$  is predefined. In (12) the similarity  $S_{ij}$  monotonously increases with the decrease of the distance between  $x_i$  and  $x_j$ . It is worth noting that if  $x_i$  and  $x_j$  do not belong to the same class, the value of  $S_{ij}$  will be zero.

The minimization problem of (11) can be reduced to the eigendecomposition problem [6]:

$$XLX^T w = \lambda DX^T w, \quad (13)$$

where  $D$  is a diagonal matrix,  $D_{ii} = \sum S_{ij}$ , and  $L = D - S$  is the Laplacian matrix [23]. There is a constraint as follows:

$$w^T XDX^T w = 1. \quad (14)$$

**3.2. Supervised Kernel Optimized Locality Preserving Projections.** We utilize the nonlinear projection  $\Phi$  to map the input data  $R^n$  onto a Hilbert space; that is,  $x \rightarrow \Phi$ . We expanded LPP into a new space  $\Phi(x) = [\Phi(x_1), \Phi(x_2), \dots, \Phi(x_n)]$ . The objective function is

$$\min \sum_{i,j=1}^n \|y_i^\Phi - y_j^\Phi\|^2 S_{ij}^\Phi, \quad (15)$$

where  $y_i^\Phi = (w^\Phi)^\top \Phi(x_i)$ . The optimal transformation matrix  $W^\Phi = [w_1^\Phi, w_2^\Phi, \dots, w_r^\Phi]$  can be obtained through (13). The eigenvector can be expressed as follows:

$$w^\Phi = \sum_{p=1} \delta_p \Phi(x_p) = Q\delta, \quad (16)$$

where  $Q = [\Phi(x_1), \Phi(x_2), \dots, \Phi(x_n)]$ ,  $\delta = [\delta_1, \delta_2, \dots, \delta_n]^T$ , and  $S^\Phi$  is the similarity matrix in the Hilbert space. Then, SLPP is generalized to the nonlinear case with kernel function. The basic kernel in (1) is  $k_0 = Q^T Q$ . Gaussian kernel  $k(x_i, x_j) = \exp(-\|x_i - x_j\|^2/2\sigma^2)$  and polynomial kernel  $k(x_i, x_j) = [\langle x_i, x_j \rangle + \theta]^d$ ,  $d \in N$ , are the most popular kernel functions in kernel trick.

We can simplify the objective function (15) as

$$\frac{1}{2} \sum_{i,j}^n \|y_i^\Phi - y_j^\Phi\|^2 S_{ij}^\Phi = \delta^T K (D^\Phi - S^\Phi) K \delta, \quad (17)$$

where  $K$  is the data-dependent kernel matrix defined in (3) and  $D^\Phi$  is a diagonal matrix where  $D_{ii}^\Phi = \sum_j S_{ij}^\Phi$ . The local structure information of data in the original space can be presented by matrix  $D^\Phi$ . That is, if  $y_i^\Phi$  is more important, the value of  $D_{ii}^\Phi$  is bigger. Consider the constraint  $(Y^\Phi)^T D^\Phi Y^\Phi = 1$ ; that is,  $\delta^T K D^\Phi K \delta = 1$ . Then minimization problem can be transformed as

$$\begin{aligned} \min_{\alpha} \quad & \delta^T K L^\Phi K \delta, \\ \text{s.t.} \quad & \delta^T K D^\Phi K \delta = 1. \end{aligned} \quad (18)$$

We can obtain the optimal  $\delta$  by solving (18). Therefore, the essence of SKOLPP is clear. That is, we first use Fisher Criterion to maximize the class separability and a data-dependent kernel can be obtained. Then, we seek the optimal projection matrix of kernel optimized SLPP to extract feature. Last, a classifier can be adopted for classification. In this paper, we use nearest neighbor classifier for recognition.

More information about LPP can be obtained from [6, 19, 21].

## 4. Experimental Results

In this section, we first verify the assumption that the classification performance is probably worse in the feature space than without using kernel tricks in some cases. And we demonstrate that our proposed kernel optimization algorithm can obtain a better performance of classification. Then we test the proposed SKOLPP and other methods on the ORL, Yale, AR, and Palmprint database.

**4.1. Kernel Optimization on Synthetic Gaussian Distributed Database.** In this part, we generated two simple datasets with Gaussian distribution by computer. Figure 1(a) shows a two-dimensional dataset with 600 samples and the coordinates are uncorrelated. The samples are separated into two classes. Each class has 300 samples with Gaussian distributions where  $\mu_x = -2$ ,  $\mu_y = 0$ ,  $\sigma_x = 4$ , and  $\sigma_y = 4$  and  $\mu_x = 2$ ,  $\mu_y = 0$ ,  $\sigma_x = 1$ , and  $\sigma_y = 1$ , respectively.

From this figure, we can see that some samples of the two classes are overlapped. We use polynomial kernel function  $k_0(x, y) = [\langle x, y \rangle + \theta]^d$  where  $d = 3$  to project the data into the empirical feature space and Figure 1(b) shows the projection of the data in the empirical feature space onto the first three

significant dimensions corresponding to the first three largest eigenvalues of  $K$ . From Figure 1(b), we observe that the class separability is worse in the feature space than that in the input space. Figure 1(c) shows the corresponding results when we use Gaussian kernel function  $k_0(x, y) = \exp(-(1/\sigma)\|x - y\|^2)$  with  $\sigma = 1.0 \times 10^5$  to project the data into the empirical feature space. Similarly, the class separability cannot be endured. As a consequence, the kernel optimization algorithm is necessary to overcome this trouble. In order to show the effectiveness of the optimization algorithm, we carried out another experiment. In this experiment, we use the polynomial kernel  $k_0(x, y) = (x, y)^3$  and Gaussian kernel  $k_0(x, y) = \exp(-1.0 \times 10^{-5}\|x - y\|^2)$  as basic kernels. We select one-third of the samples randomly to form the empirical core set  $\{a_i\}$ .

In the polynomial kernel and Gaussian kernel, the initial learning rate  $\eta_0$  of the algorithm is 0.1 while the number of iterations is 200. Figure 2(a) shows the projections of the data in the empirical feature space when the third-order polynomial kernel is used as the basic kernel. The corresponding results, when the Gaussian kernel is used, are shown in Figure 2(b). From Figure 2, we can see that the class separability of the data in the feature space is improved significantly while our kernel optimization algorithm is used.

**4.2. SKOLPP on ORL and Yale Databases.** This experiment is conducted on the well-known face image databases (ORL and Yale).

The ORL database contains 40 individuals. Each of them includes 10 different images, which show variations in facial expressions (smiling or not smiling), facial details (glasses or no glasses), and poses. Yale database is more challenging than ORL, which contains 165 grayscale images of 15 individuals. The images demonstrate variations in lighting condition (left-light, center-light, and right-light), facial expression (normal, happy, sad, sleepy, and surprised), and facial details (glasses or no glasses). Some sample images from the same individual on ORL and Yale dataset are shown in Figures 3 and 4.

**Experiment 1.** In this part, we compare the proposed method with PCA, KPCA, KOPCA, KFD, SVM, KMSVM, SLPP, and SKLPP on ORL and Yale databases. Gaussian function is used with  $\sigma^2 = 10^6$  for SVM, SKLPP, and KFD, whereas  $\sigma^2 = 10^4$  for KPCA and  $\sigma^2 = 10^8$  for KOPCA, KMSVM, and SKOLPP. The dimension of eigenvectors is 60 here.

In the experiment,  $l$  images ( $l = 2, 3, 4, 5$ ) are randomly selected from the image gallery of each individual to form the training sample set  $G_m$ , respectively, and the corresponding remaining images are taken to form the testing set  $P_n$ . The results are averaged by 5 random replicates. Table 1 presents the top recognition accuracy of PCA, KPCA, KOPCA, KFD, SVM, KMSVM, SLPP, SKLPP, and SKOLPP along different number of training samples on ORL database.

Table 1 shows the top recognition rate of all the methods on ORL database. It is clear to see that SKOLPP performs the best. Moreover, SKOLPP still performs well when the number of training samples is small. It is worth noting that SKOLPP works better than KOPCA and KMSVM while all of them used optimizing kernel. One of the reasons is that SKOLPP

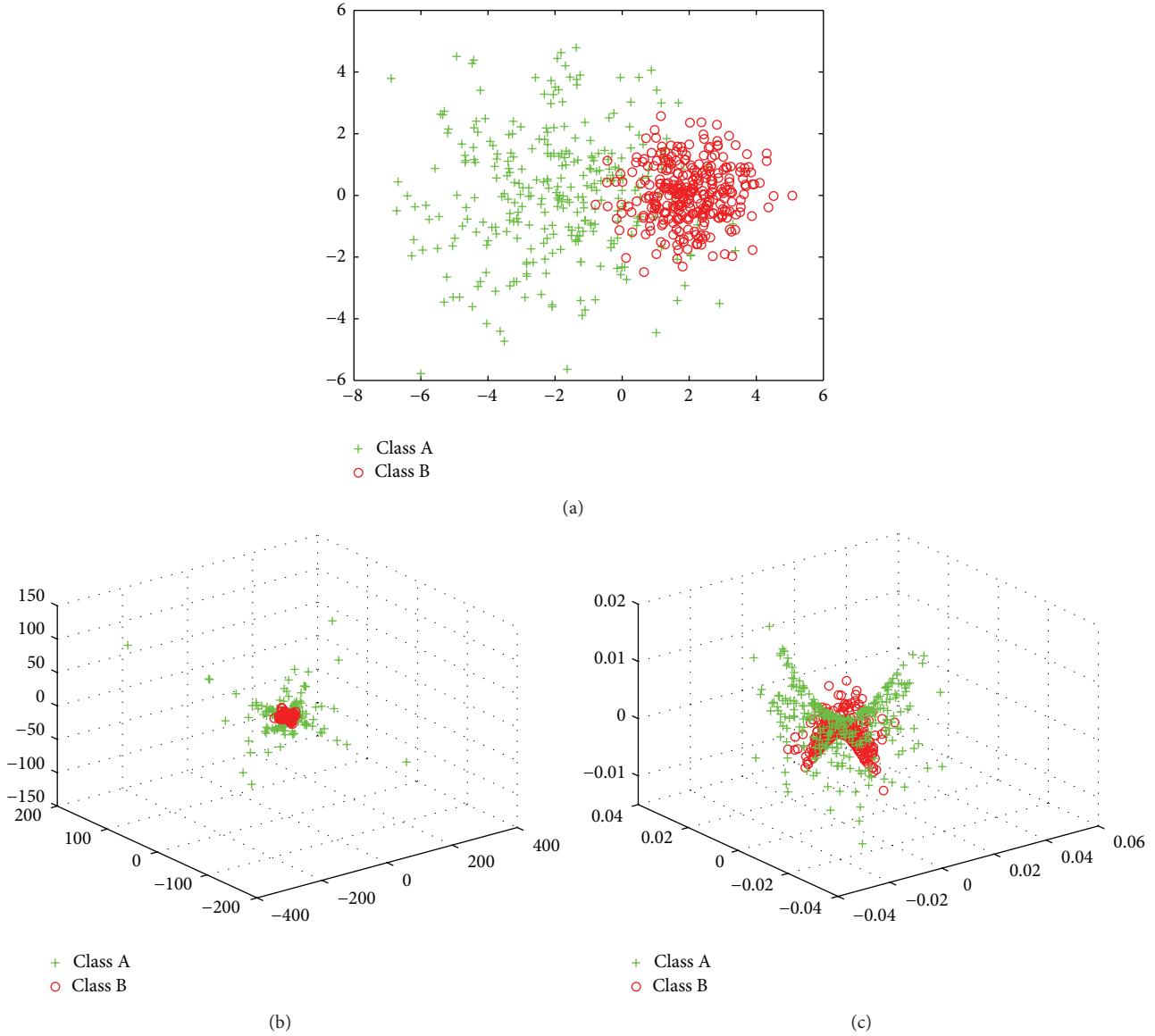


FIGURE 1: 2D database and its projections in the feature space onto the first three significant dimensions. (a) Two classes of data samples with two Gaussian distributions. (b) 3D projection in the feature space for polynomial kernel with  $d = 3$ . (c) 3D projection in the empirical feature space for Gaussian kernel with  $\sigma = 1.0 \times 10^5$ .

TABLE 1: Algorithms comparison using ORL (mean  $\pm$  std-dev%).

Method	G2/P8	G3/P7	G4/P6	G5/P5
PCA	67.5 $\pm$ 3.4	74.3 $\pm$ 2.3	78.3 $\pm$ 2.2	82.0 $\pm$ 2.4
KPCA	69.4 $\pm$ 3.2	76.1 $\pm$ 2.1	81.7 $\pm$ 2.1	86.5 $\pm$ 2.2
KOPCA	69.3 $\pm$ 3.1	80.5 $\pm$ 2.0	89.7 $\pm$ 2.2	91.0 $\pm$ 1.5
KFD	71.3 $\pm$ 3.5	77.9 $\pm$ 2.2	86.3 $\pm$ 2.1	86.0 $\pm$ 1.7
SVM	67.7 $\pm$ 3.2	78.2 $\pm$ 2.1	86.7 $\pm$ 2.4	88.3 $\pm$ 2.0
KMSVM	70.1 $\pm$ 3.4	81.7 $\pm$ 2.3	89.2 $\pm$ 2.0	90.0 $\pm$ 1.9
SLPP	62.8 $\pm$ 3.2	75.42 $\pm$ 2.3	87.51 $\pm$ 1.7	89.0 $\pm$ 1.6
SKLPP	69.4 $\pm$ 3.0	82.1 $\pm$ 2.1	89.6 $\pm$ 2.0	90.5 $\pm$ 1.7
SKOLPP	69.8 $\pm$ 3.1	83.9 $\pm$ 2.2	92.5 $\pm$ 1.3	<b>93.8 <math>\pm</math> 1.5</b>

retains the local information and obtains a face subspace that best perceives the crucial face manifold structure.

Table 2 shows the results of all the algorithms on Yale database. Obviously, SKOLPP performs always better than other methods along different number of training samples.

*Experiment 2.* We design this experiment in order to test the performance of all the algorithms under different value of  $\sigma^2$  in Gaussian kernel function. The value of  $\sigma^2$  ranges from  $10^4$  to  $10^8$ . Similarly, we select five images of each class as training samples and the results are shown in Figures 5 and 6.

From Figure 5, we can see that SKOLPP performs best compared with other methods when the value of  $\sigma^2$  is 7

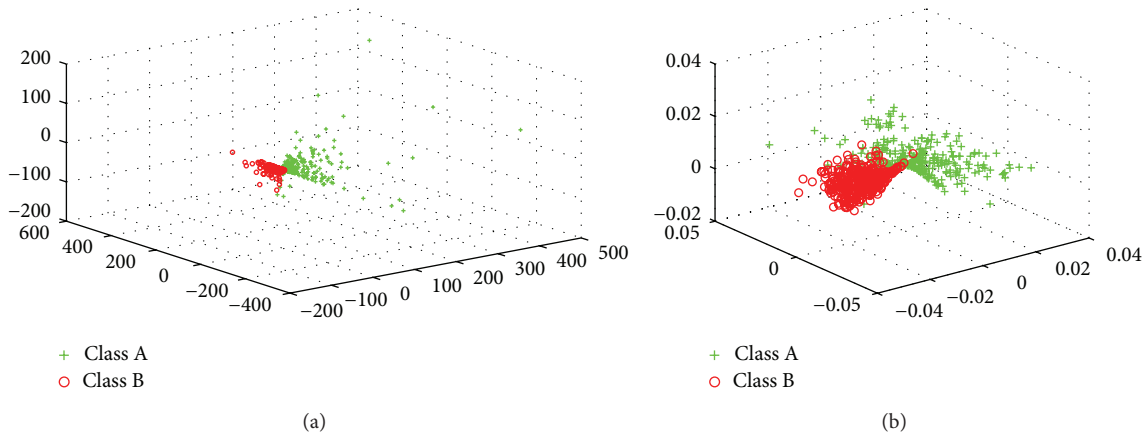


FIGURE 2: Improvement of class separability via kernel optimization algorithm. (a) 3D projection in the empirical feature space for polynomial basic kernel  $d = 3$ . (b) 3D projection in the empirical feature space for Gaussian basic kernel with  $\sigma = 1.0 \times 10^5$ .

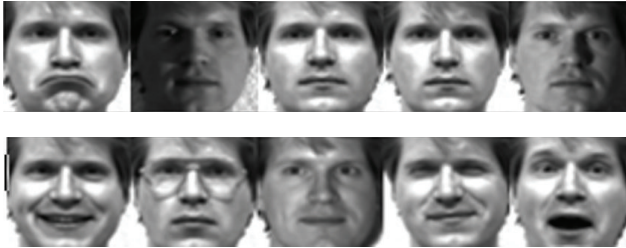


FIGURE 3: Ten sample images from Yale database.



FIGURE 4: Twenty sample images from ORL database.

TABLE 2: Algorithms comparison using Yale (mean  $\pm$  std-dev%).

Method	G2/P9	G3/P8	G4/P7	G5/P6
PCA	63.4 $\pm$ 3.7	70.5 $\pm$ 3.9	79.3 $\pm$ 4.1	79.5 $\pm$ 4.1
KPCA	64.4 $\pm$ 3.7	72.1 $\pm$ 4.1	79.5 $\pm$ 4.1	80.3 $\pm$ 3.9
KOPCA	65.0 $\pm$ 3.2	78.7 $\pm$ 3.5	82.3 $\pm$ 3.7	84.5 $\pm$ 3.2
KFD	62.9 $\pm$ 3.9	67.5 $\pm$ 3.7	75.2 $\pm$ 3.7	77.6 $\pm$ 3.4
SVM	63.5 $\pm$ 3.8	74.3 $\pm$ 3.5	81.7 $\pm$ 3.8	81.2 $\pm$ 3.7
KMSVM	65.4 $\pm$ 4.0	78.9 $\pm$ 3.7	82.6 $\pm$ 3.8	83.3 $\pm$ 3.9
SLPP	60.8 $\pm$ 4.1	64.1 $\pm$ 4.1	70.4 $\pm$ 3.9	76.8 $\pm$ 3.7
SKLPP	57.0 $\pm$ 4.3	66.8 $\pm$ 3.9	72.3 $\pm$ 3.5	85.7 $\pm$ 3.9
SKOLPP	65.1 $\pm$ 4.1	80.9 $\pm$ 3.5	86.5 $\pm$ 3.9	<b>89.6 <math>\pm</math> 3.3</b>

whereas the result is not perfect when  $\sigma^2 = 10^4 \sim 10^5$ . The small value of  $\sigma^2$  is more suitable for KPCA and KFD on

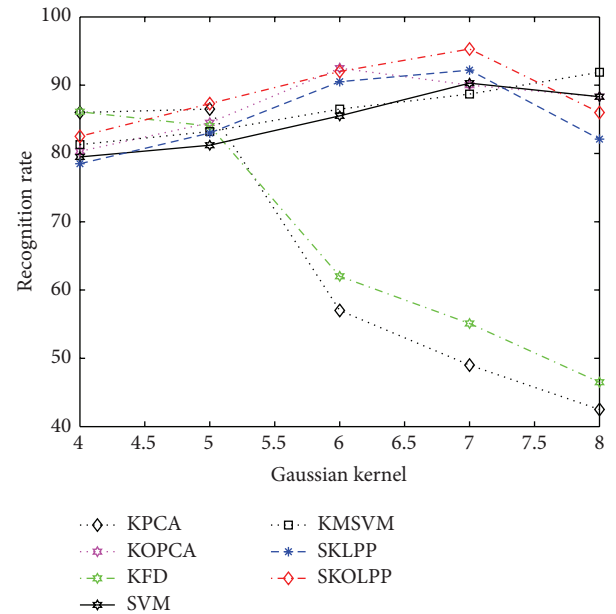


FIGURE 5: Optimal average recognition accuracy (%) among different parameters of Gaussian kernel function on ORL database.

ORL database. It is worthwhile to note that SKOLPP performs always better than other methods including KMSVM on ORL database (Figure 5). Figure 6 shows the corresponding results on Yale database. Although, in the case where the values of  $\sigma^2$  are small in Figure 6, the SKOLPP method works slightly worse than KMSVM, when the value of  $\sigma^2$  becomes larger, SKOLPP surpasses KMSVM and achieves the highest recognition result. The recognition rate of SKOLPP reaches 95.5% and 92.9% on ORL and Yale databases, respectively.

*Experiment 3.* The polynomial function is used in this part to test the performance of the proposed method. Three kinds

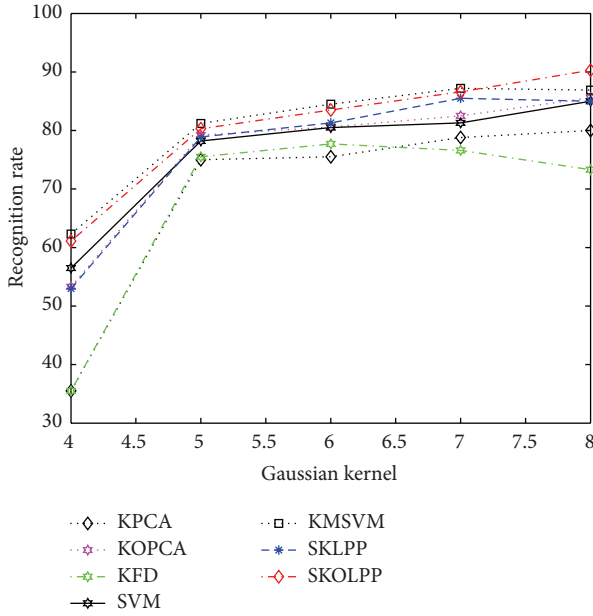


FIGURE 6: Optimal average recognition accuracy (%) among different parameters of Gaussian kernel function on Yale database.

TABLE 3: Optimal average recognition accuracy (%) among different polynomial kernel functions on ORL database.

Polynomial function	$k_1(x_i, x_j)$	$k_2(x_i, x_j)$	$k_3(x_i, x_j)$
KPCA	82.1	84.0	84.5
KOPCA	90.0	86.7	87.3
KFD	38.0	43.1	41.0
SVM	78.5	80.7	82.3
KMSVM	89.2	85.6	87.4
SKLPP	91.1	89.0	89.5
SKOLPP	<b>94.4</b>	89.5	89.0

of polynomial functions are adopted in this part,  $k_1(x_i, x_j) = \langle x_i, x_j \rangle$ ,  $k_2(x_i, x_j) = [\langle x_i, x_j \rangle]^2$ , and  $k_3(x_i, x_j) = [\langle x_i, x_j \rangle + 1]^2$ , to compare the different kernel methods, such as KPCA, KOPCA, KFD, SVM, KMSVM, and KLPP.

Five images of each class are selected as training samples and the rest of the images are used for testing. Table 3 shows the performance of different methods under different polynomial functions mentioned above.

From Table 3, we can see that the performance of KFD is unsatisfactory whereas SKLPP and SKOLPP achieve better results than other methods. Not surprisingly, SKOLPP achieves a better result as well and reaches the highest recognition rate of 94.4%. The corresponding results on Yale database are presented in Table 4.

**4.3. SKOLPP on AR Database.** This experiment is conducted on the AR face database. The AR database [24] contains over 4000 color images corresponding to 126 people's face (70 men and 56 women). Images feature frontal view faces

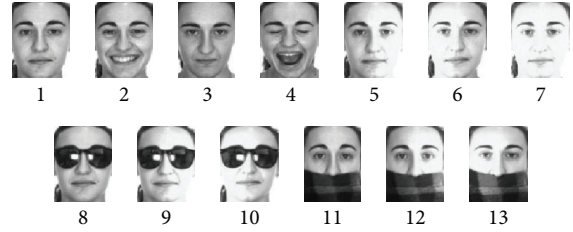


FIGURE 7: Image sample from AR database under variations in facial expressions, lighting conditions, and occlusions.

TABLE 4: Optimal average recognition accuracy (%) among different polynomial kernel functions on Yale database.

Polynomial function	$k_1(x_i, x_j)$	$k_2(x_i, x_j)$	$k_3(x_i, x_j)$
KPCA	80.0	81.1	81.1
KOPCA	87.7	85.3	83.2
KFD	68.8	72.2	74.4
SVM	74.3	78.0	78.5
KMSVM	87.3	82.4	85.6
SKLPP	89.5	75.5	71.1
SKOLPP	<b>92.9</b>	83.3	81.1

TABLE 5: Top recognition rates of different methods on AR database under varying facial expressions.

Method	$d$	Recognition rate (%)
PCA	80	72.5
2DPCA	10	78.5
LDA	39	77.5
NPE	70	75.0
SLPP	90	76.5
SKLPP	75	78.0
SKOLPP	50	<b>84.5</b>

with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). The images of each person were taken in two sessions, separated by two weeks' time. The same pictures were taken in both sessions. In this experiment, we take 100 individuals (50 men and 50 women) and use the first 13 images of each person to test the performance of all the algorithms. Thus, the total number of images used in this experiment is 1300. All images are gray with 256 levels and are of size  $13 \times 100$  pixels. To simplify the computation of the experiments, we cropped each image manually and resized each image to  $48 \times 48$  pixels. Figure 5 shows the samples of one person. To fully evaluate the performance of SKOLPP, we make three tests based on variations in facial expressions, lighting conditions, and occlusions. Gaussian function is performed with  $\sigma^2 = 10^7$  for SKOLPP.

**4.3.1. Facial Expressions.** In this test, we randomly select two images from Figure 7 (1–4) as training samples; then the remaining two images are used for testing. Therefore, the total number of training samples is 200. These images have different facial expressions. Table 5 shows the top recognition

TABLE 6: Top recognition rate of different methods on AR database under varying lighting expressions.

Method	$d$	Recognition rate (%)
PCA	160	74.4
2DPCA	38	76.2
LDA	39	81.1
NPE	70	69.5
SLPP	100	73.5
SKLPP	75	78.9
SKOLPP	50	<b>82.5</b>

TABLE 7: Top recognition rate of different methods on AR database under varying occlusions.

Method	$d$	Recognition rate (%)
PCA	70	35.2
2DPCA	24	39.3
LDA	39	56.2
NPE	65	54.7
SLPP	90	52.0
SKLPP	75	58.9
SKOLPP	50	<b>78.5</b>

rate of different algorithms. From Table 5, we can see SKOLPP achieves better results than other methods (PCA, 2DPCA [25], LDA [26], NPE [27], SLPP, and SKLPP). Particularly, SKOLPP performs better than SKLPP by 6.5% recognition rate.

**4.3.2. Lighting Conditions.** To test SKOLPP together with other methods under varying lighting conditions, we selected images from Figure 7 (1, 3, and 6) as training samples. Images 2, 4, 5, and 7 in Figure 7 were considered as testing samples. Thus the number of training samples is 300, while that of the testing samples is 400. The recognition rates are summarized in Table 6. It is obvious that SKOLPP is the most effective technique dealing with illumination variation among the listed methods. SKOLPP exceeds 6.6% rates compared with SKLPP.

**4.3.3. Occlusions.** In this part, we test the recognition rate under varying occlusions. We took images 1–7 in Figure 7 as training samples. Then the number of training samples is 700. Meanwhile, we took the rest of the images (8–13) (Figure 7) as test samples. Table 7 shows the top recognition rates of all the involved methods. Apparently, SKOLPP delivers the best result of all the algorithms while SKLPP is worse than SKOLPP by nearly 20% recognition rates. SKLPP, SLPP, and NPE also achieve good result.

**4.4. SKOLPP on Palmprint Database.** The PolyU Palmprint database contains 7752 grayscale images corresponding to 386 different palms in BMP image format (<http://www4.comp.polyu.edu.hk/~biometrics/>). Around twenty samples from each of these palms were collected in two sessions, where ten

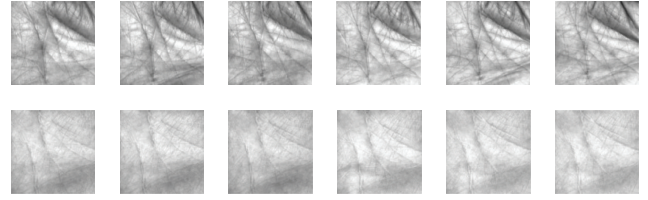


FIGURE 8: Samples of the cropped images in the PolyU Palmprint database.

TABLE 8: Top recognition rates of different methods on the PolyU Palmprint database and the corresponding dimensions.

Methods	$G3/P7$	$G4/P6$	$G5/P5$
PCA	76.0 (120)	79.5 (100)	80.1 (100)
2DPCA	76.2 (50)	80.8 (50)	82.6 (50)
LDA	78.4 (60)	82.0 (80)	83.3 (80)
NPE	78.2 (180)	80.5 (200)	80.8 (200)
SLPP	78.2 (180)	79.4 (180)	81.5 (160)
SKLPP	82.2 (120)	84.5 (100)	85.0 (120)
SKOLPP	87.4 (100)	90.3 (80)	<b>90.4 (80)</b>

samples were captured in the first session and the other ten in the second session. The average interval between the first and the second collection was two months. In this experiment, we took 200 different palms and used the first 5 images of the first session and second session, respectively. Thus, the total number of images used in this experiment is 2000. All images are gray with 256 levels and are of size  $384 \times 284$  pixels. To simplify the computation of the experiments, we cropped each image manually and resized each image to  $64 \times 64$  pixels. Figure 8 shows the samples of one person.

The maximal recognition rates of each method and the corresponding dimension are given in Table 8. A random subset with  $G_m$  ( $= G3, G4, G5$ ) is taken with labels to form the training set and the remaining part  $P_n$  ( $= P7, P6, P5$ ) to form the testing set. From Table 8, we notice that SKOLPP consistently outperforms other methods in all the cases. Particularly in the cases of  $G4/P6$  and  $G5/P5$ , SKOLPP boosts over 5% recognition rates compared with SKLPP and even delivers nearly 10% of improvement recognition rates compared to PCA. In addition, SKLPP gets the second best results while the performance of SLPP is slightly better than NPE. PCA performs the worst among all the methods.

Through the experimental results, we can come to the conclusion that SKOLPP indeed improves the class discrimination in the empirical feature space compared with SKLPP, and it is robust to the influence of illumination, facial expression, and occlusions.

## 5. Conclusion

In this paper, we proposed an efficient classification method by maximizing a measure of the class separability in the feature space named supervised kernel optimized LPP (SKOLPP). Based on the Fisher Criterion, our method achieved satisfactory classification performance by preserving the geometrical



structure of the data in the kernel feature space. SKOLPP integrates the merit of kernel optimization and SKLPP to increase the performance of nonlinear feature extraction and classification, and it is robust to the influence of illumination, facial expression, and occlusions. Several experiments were conducted to demonstrate the effectiveness of SKOLPP.

## Appendix

*Proof.* Note the empirical feature mapping  $\Phi_r : X \rightarrow \mathbf{R}^r$ , and  $y_i = \Phi_r(x_i)$ ; we know the dot product matrix  $K$  has exactly  $r$  positive eigenvalues.

Let  $Y^T = [y_1, y_2, \dots, y_N]$ ,  $Y_i^T = [y_1^i, y_2^i, \dots, y_{N_i}^i]$ ,  $i = 1, \dots, C$ . Then, we have

$$m_i = \frac{1}{N_i} \sum_{j=1}^{N_i} y_j = \frac{1}{N_i} Y_i^T \mathbf{1}_{N_i}, \quad (\text{A.1})$$

$$m_0 = \frac{1}{N} \sum_{k=1}^N y_k = \frac{1}{N} Y^T \mathbf{1}_N.$$

The empirical feature space preserves the dot product, that is,

$$\begin{aligned} & \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_C \end{bmatrix} \begin{bmatrix} Y_1^T & Y_2^T & \dots & Y_C^T \end{bmatrix} \\ &= \begin{bmatrix} Y_1 Y_1^T & Y_1 Y_2^T & \dots & Y_1 Y_C^T \\ Y_2 Y_1^T & Y_2 Y_2^T & \dots & Y_2 Y_C^T \\ \vdots & \vdots & \ddots & \vdots \\ Y_C Y_1^T & Y_C Y_2^T & \dots & Y_C Y_C^T \end{bmatrix} = Y Y^T = K \quad (\text{A.2}) \\ &= \begin{bmatrix} K_{11} & K_{12} & \dots & K_{1C} \\ K_{21} & K_{22} & \dots & K_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ K_{C1} & K_{C2} & \dots & K_{CC} \end{bmatrix}. \end{aligned}$$

Therefore,

$$\begin{aligned} \text{tr } S_b &= \frac{1}{N} \sum_{i=1}^C N_i (m_i - m_0)^T (m_i - m_0) = \frac{1}{N} \sum_{i=1}^C N_i m_i^T m_i - m_0^T m_0 \\ &= \frac{1}{N} \sum_{i=1}^C N_i \frac{1}{N_i} \mathbf{1}_{N_i}^T Y_i \frac{1}{N_i} Y_i^T \mathbf{1}_{N_i} - \frac{1}{N} \mathbf{1}_N^T Y \frac{1}{N} Y^T \mathbf{1}_N = \frac{1}{N} \\ &\cdot \sum_{i=1}^C \frac{1}{N_i} \mathbf{1}_{N_i}^T Y_i Y_i^T \mathbf{1}_{N_i} - \frac{1}{N^2} \mathbf{1}_N^T Y Y^T \mathbf{1}_N = \frac{1}{N} \sum_{i=1}^C \frac{1}{N_i} \mathbf{1}_{N_i}^T K_{ii} \mathbf{1}_{N_i} - \frac{1}{N^2} \\ &\cdot \mathbf{1}_N^T K \mathbf{1}_N = \frac{1}{N} \end{aligned}$$

$$\cdot \begin{bmatrix} \mathbf{1}_{N_1}^T & \mathbf{1}_{N_2}^T & \dots & \mathbf{1}_{N_C}^T \end{bmatrix} \begin{bmatrix} \frac{1}{N_1} K_{11} & 0 & \dots & 0 \\ 0 & \frac{1}{N_2} K_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{N_C} K_{CC} \end{bmatrix} \begin{bmatrix} \mathbf{1}_{N_1} \\ \mathbf{1}_{N_2} \\ \vdots \\ \mathbf{1}_{N_C} \end{bmatrix}$$

$$- \frac{1}{N} \mathbf{1}_N^T \begin{bmatrix} \frac{1}{N} K_{11} & \frac{1}{N} K_{12} & \dots & \frac{1}{N} K_{1C} \\ \frac{1}{N} K_{21} & \frac{1}{N} K_{22} & \dots & \frac{1}{N} K_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{N} K_{C1} & \frac{1}{N} K_{C2} & \dots & \frac{1}{N} K_{CC} \end{bmatrix} \mathbf{1}_N = \frac{1}{N} \mathbf{1}_N^T B \mathbf{1}_N,$$

$$\text{tr } S_w = \frac{1}{N} \sum_{i=1}^C \sum_{j=1}^{N_i} (y_j^i - m_i)^T (y_j^i - m_i) = \frac{1}{N}$$

$$\cdot \sum_{i=1}^C \left( \sum_{j=1}^{N_i} (y_j^i)^T y_j^i - N_i m_i^T m_i \right) = \frac{1}{N} \left( \sum_{k=1}^N y_k^T y_k - \sum_{i=1}^C N_i m_i^T m_i \right)$$

$$= \frac{1}{N} \sum_{k=1}^N y_k^T y_k - \frac{1}{N} \sum_{i=1}^C \frac{1}{N_i} \mathbf{1}_{N_i}^T K_{ii} \mathbf{1}_{N_i} = \frac{1}{N}$$

$$\cdot \mathbf{1}_N^T \begin{bmatrix} \text{diag}(k_{11}, k_{22}, \dots, k_{NN}) \end{bmatrix}$$

$$- \begin{bmatrix} \frac{1}{N_1} K_{11} & 0 & \dots & 0 \\ 0 & \frac{1}{N_2} K_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{N_C} K_{CC} \end{bmatrix} \mathbf{1}_N = \frac{1}{N} \mathbf{1}_N^T W \mathbf{1}_N.$$

(A.3)

Note formula (3); we easily get  $B = Q B_0 Q$ ,  $W = Q W_0 Q$ ; simultaneously,  $\mathbf{1}_N^T Q = \mathbf{q}^T$  and  $Q \mathbf{1}_N = \mathbf{q}$ . Hence,

$$\begin{aligned} J &= \frac{(1/N) \mathbf{1}_N^T B \mathbf{1}_N}{(1/N) \mathbf{1}_N^T W \mathbf{1}_N} = \frac{\mathbf{1}_N^T B \mathbf{1}_N}{\mathbf{1}_N^T W \mathbf{1}_N} = \frac{\mathbf{1}_N^T Q B_0 Q \mathbf{1}_N}{\mathbf{1}_N^T Q W_0 Q \mathbf{1}_N} \\ &= \frac{\mathbf{q}^T B_0 \mathbf{q}}{\mathbf{q}^T W_0 \mathbf{q}}. \end{aligned} \quad (\text{A.4})$$

□

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] J. Huang, P. C. Yuen, W.-S. Chen, and J. H. Lai, "Kernel subspace LDA with optimized kernel parameters on face recognition," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 327–332, May 2004.
- [2] L. Wang, K. L. Chan, and P. Xue, "A criterion for optimizing kernel parameters in KBDA for image retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 3, pp. 556–562, 2005.
- [3] B. Schölkopf, S. Mika, C. J. C. Burges et al., "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.
- [4] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On kernel-target alignment," in *Proceedings of the 15th Annual Neural Information Processing Systems Conference (NIPS '01)*, pp. 367–373, December 2001.
- [5] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, 2004.
- [6] X. He and P. Niyogi, "Locality preserving projections," in *Proceedings of the Conference on Advances in Neural Information Processing Systems*, 2003.
- [7] D. Wang and H. Lin, "Feature extraction based on supervised kernel locality preserving projection algorithm and application to fault diagnosis," in *Proceedings of the 9th International Conference on Electronic Measurement & Instruments (ICEMI '09)*, pp. 1-553–1-557, IEEE, Beijing, China, August 2009.
- [8] J.-B. Li, J.-S. Pan, and S.-M. Chen, "Kernel self-optimized locality preserving discriminant analysis for feature extraction and recognition," *Neurocomputing*, vol. 74, no. 17, pp. 3019–3027, 2011.
- [9] S. Sun and H. Zhao, "Normalized laplacian based optimal locality preserving projection," in *Proceedings of the International Conference on Audio, Language and Image Processing (ICALIP '10)*, pp. 478–483, November 2010.
- [10] G.-F. Lu, Z. Lin, and Z. Jin, "Face recognition using regularised generalised discriminant locality preserving projections," *IET Computer Vision*, vol. 5, no. 2, pp. 107–116, 2011.
- [11] J. Lu and Y.-P. Tan, "Regularized locality preserving projections and its extensions for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 3, pp. 958–963, 2010.
- [12] Y. Pang and Y. Yuan, "Outlier-resisting graph embedding," *Neurocomputing*, vol. 73, no. 4–6, pp. 968–974, 2010.
- [13] S. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999.
- [14] J.-S. Pan, J.-B. Li, and Z.-M. Lu, "Adaptive quasiconformal kernel discriminant analysis," *Neurocomputing*, vol. 71, no. 13–15, pp. 2754–2760, 2008.
- [15] C. Lin, B. Wang, X. Zhao, and M. Pang, "Optimizing kernel PCA using sparse representation-based classifier for mstar SAR image target recognition," *Mathematical Problems in Engineering*, vol. 2013, Article ID 847062, 10 pages, 2013.
- [16] C. Lin, X. Sun, K. Zhou, and F. Xia, "Supervised gabor-based kernel locality preserving projections for face recognition," in *Proceedings of the International Conference on and 4th International Conference on Cyber, Physical and Social Computing Internet of Things (iThings/CPSCoM '11)*, pp. 252–257, IEEE, October 2011.
- [17] J. Peng, D. R. Heisterkamp, and H. K. Dai, "Adaptive quasiconformal kernel nearest neighbor classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 656–661, 2004.
- [18] H. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 460–474, 2005.
- [19] J.-B. Li, J.-S. Pan, and S.-C. Chu, "Kernel class-wise locality preserving projection," *Information Sciences*, vol. 178, no. 7, pp. 1825–1835, 2008.
- [20] J.-B. Li, Y. Peng, and D. Liu, "Quasiconformal kernel common locality discriminant analysis with application to breast cancer diagnosis," *Information Sciences*, vol. 223, pp. 256–269, 2013.
- [21] J. Cheng, Q. Liu, H. Lu, and Y.-W. Chen, "Supervised kernel locality preserving projections for face recognition," *Neurocomputing*, vol. 67, no. 1–4, pp. 443–449, 2005.
- [22] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.
- [23] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proceedings of the 15th Annual Neural Information Processing Systems Conference (NIPS '01)*, Vancouver, Canada, December 2001.
- [24] A. Martinez and R. Benavente, "The AR face database," *Tech. Rep. 24*, 1998.
- [25] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.
- [26] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [27] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1208–1213, IEEE, Beijing, China, October 2005.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

