A New Supervised Dimensionality Reduction Algorithm Using Linear Discriminant Analysis and Locality Preserving Projection

DI ZHANG*, YUN ZHAO
School of Information Engineering
Guangdong Medical College
Dongguan, Guangdong, China
haihaiwenqi@163.com, zyun@gdmc.edu.cn

MINGHUI DU
School of Electronics and Information
South China University of Technology
Guangzhou, Guangdong, China
ecmhdu@scut.edu.cn

Abstract: Linear discriminant analysis (LDA) is one of the most popular supervised dimensionality reduction (DR) techniques used in computer vision, machine learning, and pattern classification. However, LDA only captures global geometrical structure information of the data and ignores the geometrical variation of local data points of the same class. In this paper, a new supervised DR algorithm called local intraclass geometrical variation preserving LDA (LIPLDA) is proposed. More specifically, LIPLDA first casts LDA as a least squares problem, and then explicitly incorporates the local intraclass geometrical variation into the least squares formulation via regularization technique. We also show that the proposed algorithm can be extended to non-linear DR scenarios by applying the kernel trick. Experimental results on four image databases demonstrate the effectiveness of our algorithm.

Key-Words: dimensionality reduction, locality preserving projection, linear discriminant analysis, pattern classification

1 Introduction

Appearance-based image recognition has attracted considerable interest in computer vision, machine learning, and pattern classification [1-4] in the past two decades. It is well known that the dimension of an image is usually very high. For example, an image with a resolution of 100×100 can be viewed as a 10000-dimensional vector. High dimensionality of feature vector has become a critical problem in practical applications. The data in the high-dimensional space is usually redundant and may degrade the performance of classifiers when the number of training samples is much smaller than the dimensionality of the image data. A common way to resolve this problem is to use either supervised or unsupervised DR techniques. Principal component analysis (PCA) is a popular unsupervised DR algorithm, which performs DR by projecting the original m-dimensional data onto the l-dimensional ($l<<m$) linear subspace spanned by the leading eigenvectors of the data's covariance matrix. LDA searches the projection axes on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other. Since discriminating information is encoded, it is generally believed that LDA is superior to PCA [2]. However, when applying LDA to real-world applications, there are two problems needed to be carefully considered: 1) the singularity of within-class scatter matrix; and 2) the local geometrical variations.

In the past, many LDA extensions have been developed to deal with the singularity of within-class scatter matrix, among which the most representative methods are Fisherface [3], enhanced Fisher linear discriminant models (EFM) [4], regularized discriminant analysis (RDA) [5], LDA/QR [6], maximum margin criterion (MMC) [7] and two-dimensional discriminant analysis(2DLDA) [8]. Although these methods have been shown to be effective in experiments, their generalization capability on testing data cannot be guaranteed. The main reason is that they only capture global geometrical structure information of the data via equally minimizing the distance among data points from the same class and ignore local intraclass geometrical variations. It is just the local intraclass geometrical variation that characterizes important modes of variability of data and helps to alleviate or even avoid the over-fitting problem, which will

improve the generalization ability of the algorithms [9-11].

Recently, a number of graph-based DR methods, which are also called manifold learning based discriminant approaches, have been successfully applied and became important methodologies in computer vision, machine learning and pattern classification. Some well known graph-based algorithms are locally linear embedding (LLE) [12], Isomap [13], Laplacian eigenmap [14], graph embedding [15], and locality preserving projection (LPP) [16]. All these algorithms were developed based on the assumption that the data lie on a manifold which can be modeled by a nearest-neighbor graph that preserves the local geometrical structure of the input space. Different from LLE, Isomap and Laplacian eigenmap, LPP is a linear algorithm which is quite simple and easy to realize, thus has received much attention in the research community [17-26]. As to the problem of local geometrical variations when applying LDA, however, there are only a few articles about using LPP to deal with it have been published so far, such as local LDA (LocLDA) [19], local Fisher discriminant analysis (LFDA) [25], and Graph-based Fisher analysis (GbFA) [26]. Though LocLDA integrates LDA and LPP in an unified framework, it disregards label information in the LPP formulation, which is in contradiction to the supervised nature of LDA. LFDA is still a LDA technique with the redesigned LPP-based local within-class and local between-class scatter matrices. GbFA applies Fisher criteria to the intrinsic graph and penalty graph, i.e., finds projection axes on which the intrinsic graph is minimized while the penalty graph is maximized. Different from generic LDA, both LFDA and GbFA focus only on the local structure and disregard the global structure of the data.

Motivated by the ideas in Refs.[10,16,19,25,26], in this paper, we will develop a new supervised DR algorithm, called local intraclass geometrical variation preserving LDA (LIPLDA), to integrate both global geometrical structure information and local intraclass geometrical variations of the data. More specifically, we cast LDA as a least squares problem based on spectral regression and use a modified locality preserving projection as a regularization term to model the local intraclass geometrical variations. The use of locality preserving projection as regularization term has been studied in [27, 28] in the context of regression and SVM. In [28], a tuning parameter was introduced to balance the tradeoff between global and local structures.

The rest of the paper is organized as follows. In Section 2, we give a brief review of LDA. Section 3 introduces spectral regression discriminant analysis, and our LIPLDA algorithm is presented in Section 4. Section 5 extends LIPLDA to non-linear DR scenarios using kernel tricks. Extensive experiments for object recognition are conducted in Section 6 to verify the efficiency of our methods. Conclusion and discussion are presented in Section 7.

2 A Brief Review of LDA

In classification problems, given a set of n d-dimensional samples $x_1, x_2, \ldots\ldots x_n$, belonging to C known pattern classes, LDA seeks direction v on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other [29], i.e., LDA maximizes the objective function J(v) (also known as the Fisher's criterion) as follows

$$J(\mathbf{v}) = \frac{\mathbf{v}^T \mathbf{S}_B \mathbf{v}}{\mathbf{v}^T \mathbf{S}_W \mathbf{v}} \tag{1}$$

$$\mathbf{S}_B = \sum_{k=1}^{C} m_k (\boldsymbol{\mu}^k - \boldsymbol{\mu})(\boldsymbol{\mu}^k - \boldsymbol{\mu})^T$$

$$\mathbf{S}_W = \sum_{k=1}^{C} (\sum_{i=1}^{m_k} (\mathbf{x}_i^k - \boldsymbol{\mu}^k)(\mathbf{x}_i^k - \boldsymbol{\mu}^k)^T)$$

where $\boldsymbol{\mu}$ is the total sample mean vector, $\boldsymbol{\mu}^k$ is the centroid of the k-th class, m_k is the number of samples in k-th class, and $\mathbf{x}_i^k$ is the i-th sample in the k-th class. The matrices $\mathbf{S}_B$ and $\mathbf{S}_W$ are often called the between-class scatter matrix and within-class scatter matrix, respectively.

By defining the total scatter matrix $\mathbf{S}_T = \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$, it is easy to verify that $\mathbf{S}_T = \mathbf{S}_B + \mathbf{S}_W$. The objective function (1) is then equivalent to

$$J(\mathbf{v}) = \frac{\mathbf{v}^T \mathbf{S}_B \mathbf{v}}{\mathbf{v}^T \mathbf{S}_T \mathbf{v}} \tag{2}$$

Maximizing the above function is equivalent to finding the eigenvectors of the following generalized eigen-problem associated with maximum eigenvalues

$$\mathbf{S}_B \mathbf{v} = \lambda \mathbf{S}_T \mathbf{v} \tag{3}$$

Since the rank of $\mathbf{S}_B$ is bounded by C-1, there are at most C-1 eigenvectors corresponding to non-zero eigenvalues [29].

The solution of Eq.(3) can be obtained by applying an eigen-decomposition on the matrix

$\mathbf{S}_T^{-1}\mathbf{S}_B$, given that $\mathbf{S}_T$ is nonsingular. However, when the number of features is larger than the number of samples, $\mathbf{S}_T$ is singular and $\mathbf{S}_T^{-1}$ doesn't exist. In the past few decades, various approaches have been proposed to solve this singularity problem and all of them can be divided into two categories: 1) applying eigen-value decomposition or singular value decomposition to the data matrix, which is computationally expensive in both time and memory; and 2) casting LDA as a least squares problem based on spectral regression [30], which can be efficiently solved by various iterative algorithms (*e.g.*, LSQR [31], [32]). By casting LDA as a least squares problem, we can also generalize LDA by incorporating various additional information, e.g., local intraclass geometrical variation, into the framework of least squares problem as regularization terms.

3 Spectral Regression Discriminant Analysis

In this section, we use graph embedding to reformulate LDA and show how LDA is connected to least squares problem. We start from analyzing the between-class scatter matrix $\mathbf{S}_B$.

Let $\overline{\mathbf{x}_i} = \mathbf{x}_i - \boldsymbol{\mu}$ and $\overline{\mathbf{X}^k} = [\overline{\mathbf{x}_1^k}, \overline{\mathbf{x}_1^k}, \ldots \overline{\mathbf{x}_{m_k}^k}]$ denote the centered data sample and the centered data matrix of the *k*-th class, respectively. We see that

$$\begin{aligned}\mathbf{S}_B &= \sum_{k=1}^{C} m_k (\boldsymbol{\mu}^k - \boldsymbol{\mu})(\boldsymbol{\mu}^k - \boldsymbol{\mu})^T \\ &= \sum_{k=1}^{C} m_k \left(\frac{1}{m_k} \sum_{i=1}^{m_k} \left(\mathbf{x}_i^k - \boldsymbol{\mu} \right) \right) \left(\frac{1}{m_k} \sum_{i=1}^{m_k} \left(\mathbf{x}_i^k - \boldsymbol{\mu} \right) \right)^T \qquad (4) \\ &= \sum_{k=1}^{C} \frac{1}{m_k} \sum_{i=1}^{m_k} \overline{\mathbf{x}_i^k} \sum_{i=1}^{m_k} \overline{\mathbf{x}_i^k}^T = \sum_{k=1}^{C} \overline{\mathbf{X}^k} \mathbf{W}^k \overline{\mathbf{X}^k}^T\end{aligned}$$

where $\mathbf{W}^k$ is an $m_k \times m_k$ matrix with all elements equal to $1/m_k$. If we define $\overline{\mathbf{X}} = [\overline{\mathbf{X}^1}, ..., \overline{\mathbf{X}^C}]$ as the centered sample matrix and a matrix W as

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}^1 & 0 & \cdots & 0 \\ 0 & \mathbf{W}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{W}^C \end{bmatrix} \qquad (5)$$

we have

$$\mathbf{S}_B = \overline{\mathbf{X}} \mathbf{W} \overline{\mathbf{X}}^T \qquad (6)$$

Similarly, the total scatter matrix and within-class scatter matrix can be rewritten as

$$\mathbf{S}_T = \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T = \overline{\mathbf{X}} \overline{\mathbf{X}}^T \qquad (7)$$

$$\begin{aligned}\mathbf{S}_W &= \mathbf{S}_T - \mathbf{S}_B = \overline{\mathbf{X}} \overline{\mathbf{X}}^T - \overline{\mathbf{X}} \mathbf{W} \overline{\mathbf{X}}^T \\ &= \overline{\mathbf{X}} (\mathbf{I} - \mathbf{W}) \overline{\mathbf{X}}^T = \overline{\mathbf{X}} \mathbf{L} \overline{\mathbf{X}}^T\end{aligned}$$

If we take $\mathbf{W}$ as the edge weight matrix of a graph $\mathbf{G}$, its entry W_{ij} is the weight of edge joining vertices *i* and *j*. W_{ij} =0 indicates there is no edge between vertices *i* and *j*. Thus $\mathbf{L} = \mathbf{I} - \mathbf{W}$ is called graph Laplacian.

By substituting Eq.(6) and Eq.(7) into Eq.(3), we obtain the following generalized eigen-problem

$$\overline{\mathbf{X}} \mathbf{W} \overline{\mathbf{X}}^T \mathbf{v} = \lambda \overline{\mathbf{X}} \overline{\mathbf{X}}^T \mathbf{v} \qquad (8)$$

In [30],[33], Cai et al. developed an efficient two-stage approach to solve the generalized eigen-problem (8), which is based on the following theorem.

Theorem 1. Let $\overline{\mathbf{y}}$ be the eigenvector of eigen-problem

$$\mathbf{W} \overline{\mathbf{y}} = \lambda \overline{\mathbf{y}} \qquad (9)$$

with eigenvalue λ. If $\overline{\mathbf{X}}^T \mathbf{v} = \overline{\mathbf{y}}$, then $\mathbf{v}$ is the eigenvector of eigen-problem $\overline{\mathbf{X}} \mathbf{W} \overline{\mathbf{X}}^T \mathbf{v} = \lambda \overline{\mathbf{X}} \overline{\mathbf{X}}^T \mathbf{v}$ with the same eigenvalue λ.

Theorem 1 shows that instead of solving the eigen-problem (8) directly, the LDA basis functions can be obtained through the following two steps:

1) Solve the eigen-problem in (9) to get $\overline{\mathbf{y}}$.
2) Find $\mathbf{v}$ which satisfies $\overline{\mathbf{X}}^T \mathbf{v} = \overline{\mathbf{y}}$.

In reality, such $\mathbf{v}$ may not exist. A possible way is to find a $\mathbf{v}$ that fits $\overline{\mathbf{X}}^T \mathbf{v} = \overline{\mathbf{y}}$ in the least squares sense:

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \left\| \overline{\mathbf{X}}^T \mathbf{v} - \overline{\mathbf{y}} \right\|^2 \qquad (10)$$

For the cases that the number of samples is smaller than the number of features, the above minimization problem is ill-posed. The most popular way to deal with the ill-posed problem is to impose a penalty on the norm of $\mathbf{v}$, we have

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \left\{ \left\| \overline{\mathbf{X}}^T \mathbf{v} - \overline{\mathbf{y}} \right\|^2 + \varepsilon \|\mathbf{v}\|^2 \right\} \qquad (11)$$

Since $\mathbf{W}$ is a block-diagonal matrix with *C* blocks, and the rank of each block is 1, so there are exactly *C* eigenvectors, $\overline{\mathbf{y}}_1, \overline{\mathbf{y}}_2, \cdots \overline{\mathbf{y}}_C$, for the eigen-problem $\mathbf{W} \overline{\mathbf{y}} = \lambda \overline{\mathbf{y}}$. As a result, there are *C* optimization problems like Eq.(11) needed to be

solved. For simplicity, all these optimization problems can be written in a single matrix form as

$$\hat{\mathbf{V}} = \arg\min_{\mathbf{V}} \left\{ \left\| \overline{\mathbf{X}}^T \mathbf{V} - \overline{\mathbf{Y}} \right\|_F^2 + \varepsilon \|\mathbf{V}\|_F^2 \right\} \quad (12)$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_C]$, $\overline{\mathbf{Y}} = [\overline{\mathbf{y}}_1, \overline{\mathbf{y}}_2, \cdots \overline{\mathbf{y}}_C]$, and $\|\cdot\|_F$ is the Frobenius norm of a matrix.

4 Local Intraclass Geometrical Variation Preserving LDA

By casting LDA as a least squares problem, additional information of data sets can be incorporated into LDA as regularization terms. In this section, we show how to build a regularization term for the local intraclass geometrical variation and how to solve the final optimization problem. We start from modeling local intraclass geometrical variation.

4.1 Local Intraclass Variation Modeling

LDA aims to capture global geometrical structure information and ignores the geometrical variation of local data points of the same class. However, in many real-world applications, the local intraclass geometrical variation is more important. In this paper, we use a modified LPP to model the local intraclass geometrical variation. The complete derivation and theoretical justifications of LPP can be traced back to [16]. LPP seeks to preserve local structure and intrinsic geometry of the data. The objective function of LPP is as follows

$$\frac{1}{2}\min \sum_{i,j} (y_i - y_j)^2 S_{ij} \quad (13)$$

where y_i is the one-dimensional projection of sample $\mathbf{x}_i$ and the matrix **S** is a similarity matrix whose element S_{ij} representing the similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$. A possible way of defining **S** is

$$S_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / t), & \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \delta \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where δ is sufficiently small , and $\delta > 0$. Here δ defines the radius of the local neighborhood. Or

$$S_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / t), & \mathbf{x}_i \in N_k(\mathbf{x}_j) \\ & \text{or } \mathbf{x}_j \in N_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

where $\mathbf{x}_i \in N_k(\mathbf{x}_j)$ implies that $\mathbf{x}_i$ is among the k nearest neighbors of $\mathbf{x}_j$ or vice versa [14], [17]. With the similarity matrix S defined in Eq.(14) or Eq.(15), the objective function (13) incurs a heavy penalty if neighboring points are mapped far apart in the one-dimensional output space.

From the definition of similarity matrix S, we see that neither Eq.(14) nor Eq.(15) takes sample label into consideration, i.e., the samples in the local neighborhood are considered to be within the same class, while the samples in the nonlocal region are considered to be in different classes. In reality, however, as illustrated in Fig.1, such assumption does not certainly hold. In the figure, the top left circle and the down right circle do not belong to the classes of their local neighbors. If the task at hand is classification, the desired projection axes should be the ones on which the circles are far from their nearest neighbors. However, with the similarity matrix **S** defined in Eq. (14) or Eq. (15), the objective function of LPP, i.e., Eq.(13), tends to push the circles closer to their nearest neighbors.

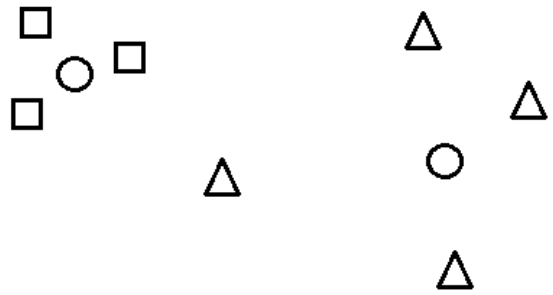

Fig.1 Illustration of local intraclass geometrical variation

In order to model the local intraclass geometrical variation more effectively, we redefine the similarity matrix S whose element is given by

$$S_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / t), & \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \delta \\ & \text{and } C_i = C_j \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

or

$$S_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / t), & \mathbf{x}_i \in N_k(\mathbf{x}_j) \\ & \text{or } \mathbf{x}_j \in N_k(\mathbf{x}_i) \text{ and } C_i = C_j \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

where C_i and C_j denote the class label of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. Formulas (16) and (17) indicate that, even if two points $\mathbf{x}_i$ and $\mathbf{x}_j$ from different classes are close to each other, the objective function doesn't incur a heavy penalty if they are mapped far apart in the one-dimensional output space because the corresponding S_{ij} is zero.

Supposing there are C one-dimensional projections of the form $y = \mathbf{v}_i^T \mathbf{x}, i = 1, \ldots C$, by substituting $y = \mathbf{v}_i^T \mathbf{x}$ into Eq.(13) and combining all these functions together into a single matrix form, following some simple algebraic steps, we see that

$$\begin{aligned} &\frac{1}{2}\sum_{i,j}\left\|\mathbf{V}^T\mathbf{x}_i - \mathbf{V}^T\mathbf{x}_j\right\|^2 S_{ij} \\ &= \frac{1}{2}\sum_{i,j}\mathrm{tr}\{\mathbf{V}^T(\mathbf{x}_i - \mathbf{x}_j)S_{ij}(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{V}\} \end{aligned} \quad (18)$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_C]$. Since the operation of trace is linear and S_{ij} is a scalar, Eq. (18) can be easily simplified as

$$\begin{aligned} &\frac{1}{2}\sum_{i,j}\mathrm{tr}\{\mathbf{V}^T(\mathbf{x}_i - \mathbf{x}_j)S_{ij}(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{V}\} \\ &= \frac{1}{2}\mathrm{tr}\left\{\mathbf{V}^T\left(\sum_{i,j}(\mathbf{x}_i - \mathbf{x}_j)S_{ij}(\mathbf{x}_i - \mathbf{x}_j)^T\right)\mathbf{V}\right\} \\ &= \frac{1}{2}\mathrm{tr}\left\{\mathbf{V}^T\left(2\sum_{i,j}\mathbf{x}_i S_{ij}\mathbf{x}_i^T - 2\sum_{i,j}\mathbf{x}_i S_{ij}\mathbf{x}_j^T\right)\mathbf{V}\right\} \\ &= \mathrm{tr}\left\{\mathbf{V}^T\left(\mathbf{X}\mathbf{D}\mathbf{X}^T - \mathbf{X}\mathbf{S}\mathbf{X}^T\right)\mathbf{V}\right\} \\ &= \mathrm{tr}\left\{\mathbf{V}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{V}\right\} \end{aligned} \quad (19)$$

where $\mathbf{D} = \mathrm{diag}(D_{11}, \cdots, D_{nn})$, $D_{ii} = \sum_{j=1}^{n} S_{ij} (i = 1, \cdots, n)$ and $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is the Laplacian matrix.

4.2 The LIPLDA algorithm

The local intraclass geometrical variation can be incorporated into the least squares formulation of LDA as a regularization term defined in Eq.(19). Given a matrix $\overline{\mathbf{Y}} = [\overline{\mathbf{y}}_1, \overline{\mathbf{y}}_2, \cdots \overline{\mathbf{y}}_C]$, whose column vector $\overline{\mathbf{y}}_i$ is the eigenvector with eigenvalue λ_i for the eigen-problem $\mathbf{W}\overline{\mathbf{y}} = \lambda\overline{\mathbf{y}}$, our LIPLDA algorithm calculates an optimal projection matrix $\mathbf{V}$ from the following optimization problem:

$$\hat{\mathbf{V}} = \arg\min_{\mathbf{V}}\left\{\left\|\overline{\mathbf{X}}^T\mathbf{V} - \overline{\mathbf{Y}}\right\|_F^2 + (1-\varepsilon)\mathrm{tr}\left\{\mathbf{V}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{V}\right\} + \varepsilon\|\mathbf{V}\|_F^2\right\} \quad (20)$$

where $\varepsilon \in (0,1)$ is a tuning parameter that controls the tradeoff between global geometrical structure and local intraclass geometrical variation.

By differentiating the right part of Eq.(20) with respect to $\mathbf{V}$, setting the derivative equal to zero, after some manipulation, we get

$$\overline{\mathbf{X}}\overline{\mathbf{X}}^T\mathbf{V} + (1-\varepsilon)\overline{\mathbf{X}}\mathbf{L}\overline{\mathbf{X}}^T\mathbf{V} + \varepsilon\mathbf{V} = \overline{\mathbf{X}}\overline{\mathbf{Y}} \quad (21)$$

Because matrix $\overline{\mathbf{X}}\overline{\mathbf{X}}^T + (1-\varepsilon)\overline{\mathbf{X}}\mathbf{L}\overline{\mathbf{X}}^T + \varepsilon\mathbf{I}$ is nonsingular, the optimal projection matrix $\mathbf{V}$ can be computed as

$$\hat{\mathbf{V}} = \left(\overline{\mathbf{X}}\overline{\mathbf{X}}^T + (1-\varepsilon)\overline{\mathbf{X}}\mathbf{L}\overline{\mathbf{X}}^T + \varepsilon\mathbf{I}\right)^{-1}\overline{\mathbf{X}}\overline{\mathbf{Y}} \quad (22)$$

Algorithm: LIPLDA

Summarizing the previous sections, the LIPLDA algorithm is as follows

Training:

1) Construct similarity matrix $\mathbf{S}$ using either Eq.(16) or Eq.(17).
2) Solve the eigen-problem Eq.(9) to get $\overline{\mathbf{Y}}$.
3) Use Eq.(22) to compute $\mathbf{V}$.
4) Obtain a feature matrix $\mathbf{Z}$ of the training data by $\mathbf{Z} = \mathbf{V}^T\overline{\mathbf{X}}$.

Test:

1) For a test sample $\mathbf{x}$, center it by $\bar{\mathbf{x}} = \mathbf{x} - \boldsymbol{\mu}$, where $\boldsymbol{\mu}$ is the centroid of training data.
2) Obtain a feature vector of the test sample by $\mathbf{z} = \mathbf{V}^T\bar{\mathbf{x}}$.

5 Kernel LIPLDA for non-linear DR

The first kernel-based DR method, kernel principal component analysis (KPCA) was originally developed by Scholkopf et al. in 1998 [34], and kernel Fisher discriminant analysis (KDA) was introduced by Mika et al. in 1999 [35]. Subsequent research saw the development of a series of KDA algorithms (see Baudat and Anouar [36], Lu et al. [37], Yang et al. [38], Cortes et al. [39], and Lin et al. [40]). Because of its ability to extract the most discriminatory nonlinear features, KDA has been found to be very effective in many real-world applications. Compared to other methods for nonlinear feature extraction, kernel-based DR methods have the advantage that they do not require nonlinear optimization. Here we show how LIPLDA can be extended to non-linear DR scenarios.

5.1 A Brief Review of KDA

The idea of KDA is to extend LDA to a nonlinear version by using the so-called kernel trick [36]. Assume that we have a nonlinear mapping $\phi(\cdot)$ that maps a point in a *d*-dimensional input space into a *r*-dimensional feature space, i.e.,

$$\phi: \mathrm{R}^d \to \mathrm{R}^r \quad (23)$$

Here, the dimension of the feature space *r* can either be finite or infinite. Let $\boldsymbol{\mu}_\phi^k = (1/m_k)\sum_{i=1}^{m_k}\phi(\mathbf{x}_i^k)$, $\boldsymbol{\mu}_\phi = (1/n)\sum_{i=1}^{n}\phi(\mathbf{x}_i)$ and

$\overline{\phi(\mathbf{x}_i)} = \phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi$ denote the centroid of the *k*-th class, the global centroid and the centered data sample, respectively, in the feature space. For the new between-class scatter matrix in the feature space, following some simple algebraic steps, we see that

$$\begin{aligned}\mathbf{S}_B^\phi &= \sum_{k=1}^{C} m_k(\boldsymbol{\mu}_\phi^k - \boldsymbol{\mu}_\phi)(\boldsymbol{\mu}_\phi^k - \boldsymbol{\mu}_\phi)^T \\ &= \sum_{k=1}^{C} m_k\left(\frac{1}{m_k}\sum_{i=1}^{m_k}\left(\phi(\mathbf{x}_i^k) - \boldsymbol{\mu}_\phi\right)\right)\left(\frac{1}{m_k}\sum_{i=1}^{m_k}\left(\phi(\mathbf{x}_i^k) - \boldsymbol{\mu}_\phi\right)\right)^T \\ &= \sum_{k=1}^{C}\frac{1}{m_k}\sum_{i=1}^{m_k}\overline{\phi(\mathbf{x}_i^k)}\sum_{i=1}^{m_k}\overline{\phi(\mathbf{x}_i^k)}^T = \sum_{k=1}^{C}\overline{\phi(\mathbf{X}^k)}\mathbf{W}^k\overline{\phi(\mathbf{X}^k)}^T\end{aligned}$$

where $\overline{\phi(\mathbf{X}^k)} = [\overline{\phi(\mathbf{x}_1^k)}, \ldots\ldots, \overline{\phi(\mathbf{x}_{m_k}^k)}]$ is the centered data matrix of the *k*-th class in the feature space. If we define $\overline{\phi(\mathbf{X})} = [\overline{\phi(\mathbf{X}^1)}, \ldots\ldots, \overline{\phi(\mathbf{X}^C)}]$ as the centered sample matrix in the feature space, we have

$$\mathbf{S}_B^\phi = \overline{\phi(\mathbf{X})}\mathbf{W}\overline{\phi(\mathbf{X})}^T \tag{24}$$

Similarly, the new total scatter matrix and within-class scatter matrix in the feature space can be rewritten as

$$\begin{aligned}\mathbf{S}_T^\phi &= \sum_{i=1}^{n}(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi)(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi)^T \\ &= \overline{\phi(\mathbf{X})}\overline{\phi(\mathbf{X})}^T \\ \mathbf{S}_W^\phi &= \mathbf{S}_T^\phi - \mathbf{S}_B^\phi = \overline{\phi(\mathbf{X})}\overline{\phi(\mathbf{X})}^T - \overline{\phi(\mathbf{X})}\mathbf{W}\overline{\phi(\mathbf{X})}^T \\ &= \overline{\phi(\mathbf{X})}(\mathbf{I} - \mathbf{W})\overline{\phi(\mathbf{X})}^T\end{aligned} \tag{25}$$

By replacing $\mathbf{S}_B$ and $\mathbf{S}_T$ in Eq.(2) with $\mathbf{S}_B^\phi$ and $\mathbf{S}_T^\phi$, respectively, we obtain the corresponding objective function in the feature space as follows

$$J(\mathbf{v}) = \frac{\mathbf{v}^T\mathbf{S}_B^\phi\mathbf{v}}{\mathbf{v}^T\mathbf{S}_T^\phi\mathbf{v}} \tag{26}$$

However, direct calculation of **v** by solving the corresponding GED problem of Eq.(26) is difficult because the dimension of **v** is not known and furthermore it could be infinite. To resolve this problem, instead of mapping the data explicitly, an alternative way is using dot-products of the training samples to reformulate the objective function [35,36].

Clearly, the optimal projection vector **v** is a linear combination of the centered training samples in the feature space, i.e.,

$$\mathbf{v} = \sum_{i=1}^{n}\alpha_i\,\overline{\phi(\mathbf{x}_i)} = \overline{\phi(\mathbf{X})}\boldsymbol{\alpha} \tag{27}$$

for some $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots\alpha_n]^T \in \mathrm{R}^n$.

Considering that the projection of a centered sample $\overline{\phi(\mathbf{x}_i)}$ onto the vector v in the feature space is obtained by the inner product of v and the centered sample itself, the projection of the entire training data is obtained by

$$\mathbf{v}^T\overline{\phi(\mathbf{X})} = \boldsymbol{\alpha}^T\overline{\phi(\mathbf{X})}^T\overline{\phi(\mathbf{X})} = \boldsymbol{\alpha}^T\overline{\mathbf{K}} \tag{28}$$

where $\overline{\mathbf{K}} = \overline{\phi(\mathbf{X})}^T\overline{\phi(\mathbf{X})}$ is a centered symmetric kernel matrix whose (*i*,*j*) element is $\overline{k(\mathbf{x}_i, \mathbf{x}_j)} = \overline{\phi(\mathbf{x}_i)}^T\overline{\phi(\mathbf{x}_j)}$. Then, for the objective function (26), following some simple algebraic steps, we see that

$$J(\mathbf{v}) = \frac{\mathbf{v}^T\mathbf{S}_B^\phi\mathbf{v}}{\mathbf{v}^T\mathbf{S}_T^\phi\mathbf{v}} = \frac{\mathbf{v}^T\overline{\phi(\mathbf{X})}\mathbf{W}\overline{\phi(\mathbf{X})}^T\mathbf{v}}{\mathbf{v}^T\overline{\phi(\mathbf{X})}\overline{\phi(\mathbf{X})}^T\mathbf{v}} = \frac{\boldsymbol{\alpha}^T\overline{\mathbf{K}}\mathbf{W}\overline{\mathbf{K}}\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T\overline{\mathbf{K}}\overline{\mathbf{K}}\boldsymbol{\alpha}}$$

The optimal $\boldsymbol{\alpha}$'s can be obtained by solving the following GED problem:

$$\overline{\mathbf{K}}\mathbf{W}\overline{\mathbf{K}}\boldsymbol{\alpha} = \lambda\overline{\mathbf{K}}\overline{\mathbf{K}}\boldsymbol{\alpha} \tag{29}$$

By generalizing the idea of **Theorem 1** to KDA, we have the following theorem

Theorem 2. Let $\overline{\mathbf{y}}$ be the eigenvector of eigen-problem $\mathbf{W}\overline{\mathbf{y}} = \lambda\overline{\mathbf{y}}$ with eigenvalue λ. If $\overline{\mathbf{K}}\boldsymbol{\alpha} = \overline{\mathbf{y}}$, then $\boldsymbol{\alpha}$ is the eigenvector of eigen-problem in Eq.(29) with the same eigenvalue λ.

Proof: With $\overline{\mathbf{K}}\boldsymbol{\alpha} = \overline{\mathbf{y}}$ and $\mathbf{W}\overline{\mathbf{y}} = \lambda\overline{\mathbf{y}}$, following some algebraic steps, the left side of Eq.(29) can be rewritten as

$$\overline{\mathbf{K}}\mathbf{W}\overline{\mathbf{K}}\boldsymbol{\alpha} = \overline{\mathbf{K}}\mathbf{W}\overline{\mathbf{y}} = \overline{\mathbf{K}}\lambda\overline{\mathbf{y}} = \lambda\overline{\mathbf{K}}\overline{\mathbf{y}} = \lambda\overline{\mathbf{K}}\overline{\mathbf{K}}\boldsymbol{\alpha}$$

Thus, $\boldsymbol{\alpha}$ is the eigenvector of eigen-problem in Eq.(29) with the same eigenvalue λ.

□

Following the same two-stage approach as mentioned in Section 3, the KDA solution $\boldsymbol{\alpha}$ can be obtained by solving the following regularized least squares problem

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}}\left\{\left\|\overline{\mathbf{K}}\boldsymbol{\alpha} - \overline{\mathbf{y}}\right\|^2 + \varepsilon\|\boldsymbol{\alpha}\|^2\right\} \tag{30}$$

Again, since there are total *C* optimization problems like Eq.(30) needed to be solved, we can combine them into a single matrix form as

$$\hat{\mathbf{A}} = \arg\min_{\mathbf{A}}\left\{\left\|\overline{\mathbf{K}}\mathbf{A} - \overline{\mathbf{Y}}\right\|_F^2 + \varepsilon\|\mathbf{A}\|_F^2\right\} \tag{31}$$

where $\mathbf{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots\boldsymbol{\alpha}_C]$.

5.2 Kernel Local Intraclass Geometrical Variation Modeling

Since the projection of a centered sample $\overline{\phi(\mathbf{x}_i)}$ onto the vector **v** in the feature space is

obtained by the inner product of **v** and the centered sample itself, we can similarly define an objective function of LPP in the feature space as follows

$$\frac{1}{2}\min \sum_{i,j}\left\|\mathbf{v}^T\overline{\phi(\mathbf{x}_i)}-\mathbf{v}^T\overline{\phi(\mathbf{x}_j)}\right\|^2 S_{ij} \tag{32}$$

where S_{ij} is the same as defined in Eq.(16) or Eq.(17). Following similar procedure described in section 4.1, we have

$$\begin{aligned}&\frac{1}{2}\sum_{i,j}\left\|\mathbf{V}^T\mathbf{x}_i-\mathbf{V}^T\mathbf{x}_j\right\|^2 S_{ij}\\&=\mathrm{tr}\left\{\mathbf{V}^T\overline{\phi(\mathbf{X})}\mathbf{L}\overline{\phi(\mathbf{X})}^T\mathbf{V}\right\}\end{aligned} \tag{33}$$

where $\mathbf{V}=[\mathbf{v}_1,\mathbf{v}_2,\ldots\mathbf{v}_C]$, $\mathbf{D}=\mathrm{diag}(D_{11},\cdots,D_{nn})$, $D_{ii}=\sum_{j=1}^{n}S_{ij}(i=1,\cdots,n)$ and $\mathbf{L}=\mathbf{D}-\mathbf{S}$. Substituting $\mathbf{V}^T\overline{\phi(\mathbf{X})}=\mathbf{A}^T\overline{\mathbf{K}}$ into Eq.(33), we have the final form of the objective function of LPP in the kernel space

$$\min \mathrm{tr}\left\{\mathbf{A}^T\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}\mathbf{A}\right\} \tag{34}$$

5.3 Kernel LIPLDA

Given a matrix $\overline{\mathbf{Y}}=[\overline{\mathbf{y}}_1,\overline{\mathbf{y}}_2,\cdots\overline{\mathbf{y}}_C]$, whose column vector $\overline{\mathbf{y}}_i$ is the eigenvector with eigenvalue λ_i for the eigen-problem $\mathbf{W}\overline{\mathbf{y}}=\lambda\overline{\mathbf{y}}$, our kernel LIPLDA (LIPKDA) algorithm calculates the matrix **A,** whose entries are the expansion coefficients of the optimal transformation matrix **V**, from the following optimization problem:

$$\hat{\mathbf{A}}=\arg\min_{\mathbf{A}}\left\{\left\|\overline{\mathbf{K}}\mathbf{A}-\overline{\mathbf{Y}}\right\|_F^2+(1-\varepsilon)\mathrm{tr}\left\{\mathbf{A}^T\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}\mathbf{A}\right\}+\varepsilon\|\mathbf{A}\|_F^2\right\} \tag{35}$$

where $\varepsilon\in(0,1)$ is a tuning parameter that controls the tradeoff between global geometrical structure and local intraclass geometrical variation in the feature space.

By differentiating the right part of Eq.(35) with respect to **A**, setting the derivative equal to zero, after some manipulation, we get

$$\overline{\mathbf{K}}^2\mathbf{A}+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}\mathbf{A}+\varepsilon\mathbf{A}=\overline{\mathbf{K}}\overline{\mathbf{Y}} \tag{36}$$

To solve Eq.(36), we need the following theorem

Theorem 3. Matrix $\overline{\mathbf{K}}^2+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}+\varepsilon\mathbf{I}$ is nonsingular.

Proof: Let $\mathbf{F}=\overline{\mathbf{K}}^2+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}$. By the definition of Laplacian matrix **L**, it is easy to verify that **L** is a symmetric positive semi-definite matrix. With Schur decomposition, we get

$$\mathbf{L}=\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \tag{37}$$

where $\mathbf{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots\lambda_n)$ is a diagonal matrix. Let $\mathbf{P}=\mathbf{Q}\mathbf{\Lambda}^{1/2}$, we have $\mathbf{L}=\mathbf{P}\mathbf{P}^T$. Thus **F** can be rewritten as

$$\begin{aligned}\mathbf{F}&=\overline{\mathbf{K}}^2+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{P}\mathbf{P}^T\overline{\mathbf{K}}\\&=\overline{\mathbf{K}}^2+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{P}\left(\overline{\mathbf{K}}\mathbf{P}\right)^T\end{aligned} \tag{38}$$

It follows that **F** is symmetric positive definite. By Cholesky decomposition, **F** can further be simplified as

$$\mathbf{F}=\mathbf{G}\mathbf{G}^T \tag{39}$$

Let $\mathbf{G}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the singular value decomposition of **G**, we have

$$\begin{aligned}\mathbf{F}+\varepsilon\mathbf{I}&=\mathbf{G}\mathbf{G}^T+\varepsilon\mathbf{I}=\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T+\varepsilon\mathbf{I}\\&=\mathbf{U}(\mathbf{\Sigma}^2+\varepsilon\mathbf{I})\mathbf{U}^T\end{aligned} \tag{40}$$

Thus

$$\left|\overline{\mathbf{K}}^2+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}+\varepsilon\mathbf{I}\right|=\left|\mathbf{U}(\mathbf{\Sigma}^2+\varepsilon\mathbf{I})\mathbf{U}^T\right|=\left|\mathbf{\Sigma}^2+\varepsilon\mathbf{I}\right|$$

which is nonsingular because $\varepsilon>0$.

□

With **Theorem 3**, the optimal solution can be computed as

$$\hat{\mathbf{A}}=\left(\overline{\mathbf{K}}^2+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}+\varepsilon\mathbf{I}\right)^{-1}\overline{\mathbf{K}}\overline{\mathbf{Y}} \tag{41}$$

Algorithm: LIPKDA

Summarizing the previous sections, the LIPKDA algorithm is as follows

Training:

1) Generate a centered kernel matrix $\overline{\mathbf{K}}=\overline{\phi(\mathbf{X})}^T\overline{\phi(\mathbf{X})}$ from the training samples.
2) Solve the eigen-problem Eq.(9) to get $\overline{\mathbf{Y}}$.
3) Use Eq.(41) to compute **A**.
4) Obtain a nonlinear feature matrix **Z** of the training data by $\mathbf{Z}=\mathbf{A}^T\overline{\mathbf{K}}$.

Test:

1) For a test sample **x**, generate a centered kernel vector $\overline{\mathbf{k}(\mathbf{x})}=\left[\overline{k(\mathbf{x},\mathbf{x}_1)},\overline{k(\mathbf{x},\mathbf{x}_2)},\ldots,\overline{k(\mathbf{x},\mathbf{x}_n)}\right]^T$, where $\overline{k(\mathbf{x},\mathbf{x}_i)}=\overline{\varphi(\mathbf{x})}^T\overline{\varphi(\mathbf{x}_i)}$.
2) Obtain a nonlinear feature vector of the test sample by $\mathbf{z}=\mathbf{A}^T\overline{\mathbf{k}(\mathbf{x})}$.

In LIPKDA, the kernel function $k(\cdot,\cdot)$ plays an important role and the essential property of the kernel function is that it should be decomposed into an inner product of a mapping $\varphi(\cdot)$ to itself, i.e., $k(\mathbf{x}_i,\mathbf{x}_j)=\varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$. However, it is obviously that not all the functions meet this property. To be a proper kernel function, a function should meet

the so-called *Mercer's* condition [41] and the two most popular kernels are the polynomial kernel $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i{}^T \mathbf{x}_j + c)^d$ and the Gaussian RBF kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma)$ in which *c*, *d*, and σ are the kernel parameters.

In the training of LIPKDA algorithm, the most time consuming part is Step 3 where the matrix inverse problem should be solved. Because the matrices $\overline{\mathbf{K}}$ and $\mathbf{L}$ in Eq.(41) are $\mathbf{R}^{n\times n}$, the computational complexity of Step 3 is normally $O(n^3)$. Nevertheless, it is unnecessary to compute the matrix inverse involved in Eq.(41) directly. The detailed efficient procedure is discussed as follows.

Since $\mathbf{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots \boldsymbol{\alpha}_C]$, $\overline{\mathbf{Y}} = [\overline{\mathbf{y}}_1, \overline{\mathbf{y}}_2, \cdots \overline{\mathbf{y}}_C]$, let $\mathbf{H} = \overline{\mathbf{K}}^2 + (1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}} + \varepsilon \mathbf{I}$ and $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_C] = [\overline{\mathbf{X}}\overline{\mathbf{y}}_1, \overline{\mathbf{X}}\overline{\mathbf{y}}_2, \ldots \overline{\mathbf{X}}\overline{\mathbf{y}}_C]$, Eq.(41) can be decomposed into the following *C* linear equations:

$$\mathbf{H}\boldsymbol{\alpha}_i = \mathbf{p}_i, i = 1,2,\ldots C \quad (42)$$

There are many efficient iterative algorithms have been proposed to solve Eq.(42). In this paper, we use LSQR algorithm, an iterative algorithm designed to solve large scale sparse linear equations and lest squares problems [31]. In each iteration, LSQR needs to compute two matrix-vector products [32]. The computational complexity of LSQR for solving Eq.(42) is normally $O(n^2+n)$. If the sample number is large and parallel computation is applicable, using LSQR algorithm will be more efficient than performing matrix inverse directly.

6 Experimental results

In this section, two experiments are designed to evaluate the performance of the proposed algorithms. The first experiment is on face recognition and the second is on artificial object recognition. Face recognition is performed on three face databases (Yale, ORL, and PIE) and artificial object recognition is performed on COIL20 image database [42]. In all the experiments, we use Euclidean metric and nearest neighbor classifier for classification due to the simplicity. In order to get a fair result, for all experiments, we adopt a two-phase scheme: 1) perform model selection, i.e., to determine the proper parameters for all the involved algorithms; and 2) reevaluate all the methods with the parameters got in the phase of model selection. Both the two phases are carried on the same data sets but under different partitions. The implementation environment is the personal computer with Intel(R) Core(TM)2 Duo CPU P8700 @ 2.53GHz, 4 GB memory.

Eight DR algorithms, namely, LDA, LPP [16], LocLDA [19], KPCA [43], KDA [43], complete kernel Fisher discriminant analysis (CKFD) [38], the proposed LIPLDA and LIPKDA are tested and compared. To perform a fair comparison, we split these eight methods into two groups: linear group (including LDA, LPP, LocLDA, and LIPLDA) and non-linear group (including KPCA, KDA, CKFD, and LIPKDA). For non-linear DR methods, in this paper, the Gaussian RBF kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / \sigma)$ is used.

6.1 Experiment on Face Recognition

The Yale face database [44] contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expressions or lighting conditions. The images demonstrate variations in lighting conditions (left-light, center-light, right-light), facial expressions (normal, happy, sad, sleep, surprised, and wink), and with/without glasses.

The ORL face database [45] has a total number of 400 images of 40 people. There are ten different images per subject. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken with a tolerance for some tilting and rotation.

The CMU PIE database [46] contains 68 subjects with 41,368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. We choose the five near frontal poses (C05, C07, C09, C27, C29) and use all the 11,544 images under different illuminations and expressions where each person has 170 images except a few bad images.

In our experiments, all the images are manually aligned, cropped and resized to have a resolution of 32×32 pixels. Fig.2 shows some examples where three sample images of one subject are randomly chosen from each database. For each database, we randomly partition the images into a training set (*n* images per subject for training) and a test set (the remaining images are used for testing). The detailed description of partition for the phases of model selection and performance evaluation is listed in Table 1. The partition procedure is repeated 20 times and we obtain 20 different training and testing sample sets. The first 10 are used for the phase of

model selection and the others for the phase of performance evaluation.

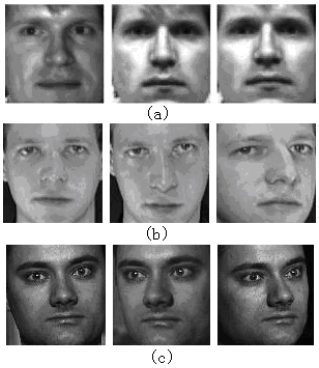

Fig.2 Samples from (a) Yale, (b) ORL, (c) PIE

In the phase of model selection, our goal is to determine proper kernel parameters (i.e., the width σ of the Gaussian RBF kernel), the dimension of the projection subspace for each method, the fusion coefficient that determines the weight ratio between regular and irregular discriminant information for CKFD [38], and the tuning parameter ε that controls the tradeoff between global geometrical structure and local intraclass geometrical variation in our proposed algorithms. Since it is very difficult to determine these parameters at the same time, a stepwise selection strategy is more feasible and thus is adopted here [37,38]. Specifically, we fix the subspace dimension and the tuning parameter ε or the fusion coefficient (for LIPKDA or CKFD) in advance and try to find the optimal kernel parameter for the Gaussian RBF kernel function. To get the proper kernel parameter, we use the global-to-local search strategy [47]. Then, based on the chosen kernel parameter, we can choose the optimal subspace dimension for each method. Finally, the tuning parameter ε or the fusion coefficient is determined with respect to the other chosen parameters.

The error rates of the random 10 different splits on three face databases with all the tested DR algorithms are presented in Fig.3. The training size used in Fig.3 is 5, 5, and 30 per subject for Yale, ORL, and PIE, respectively. From Fig.3, we can see some obvious conclusions as follows:

1. KPCA has the lowest performance among all the tested methods. This is because unlike other methods, KPCA yields projection directions which have minimal reconstruction error by describing as much variance of the data as possible, thus the yielded directions are meant for reconstruction, not for classification.
2. Except for KPCA, kernel-based methods always achieve lower error rates than their corresponding linear counterparts, which demonstrates that non-linear features play an important role in face recognition.
3. For either linear or non-linear group, our proposed LIPLDA and LIPKDA outperform other DR methods. This demonstrates that either global geometrical structure or local intraclass geometrical variation contains important discriminant information for classification, the fusion of these two kinds of information can achieve better results. Moreover, further improvement can be achieved if class label is taken into consideration when constructing local discriminant information.
4. LPP is slightly better than LDA on Yale database, while LDA outperforms LPP on ORL and PIE database. This implies that the relative importance of local and global structures in object recognition depends on specific data sets. For example, the local structure may contain less effective discriminative information in ORL and PIE database than in Yale database.

We then provide detailed performance comparison of the eight methods in Tables 2-4, where the mean error rates and standard deviations of the 10 different partitions on each data set with different training numbers are reported. Except for the case that the training data size n is 2 when dealing with Yale database, it is clear that the proposed LIPLDA and LIPKDA achieves the best performance in linear and non-linear groups, respectively. From Table 2, we can observe that the error rates of LocLDA, LIPLDA and LIPKDA are almost the same and are higher than that of LPP when the training data size n is 2. This implies that for some applications, when the number of training sample per subject is extremely low, it is difficult for the joint global and local information based methods to capture more useful discriminant information, thus fusing both local and global discriminant information does not help. For the results on PIE database listed in Table 4, it is interesting to note that the methods in the same group (except for KPCA in the non-linear group) all achieve comparably low error rates when the training data size is large, e.g., n=120. Considering the large variance of images in PIE database, this may be due to the fact that in some cases when the training data size and data variance is large, the useful discriminant information of local intraclass geometrical variation is corrupted by the densely and randomly distributed sample points, causing LPP-based techniques to capture no more new discriminant information other than global geometrical structure information, hence integrating

both local and global information makes little help in improving performance.

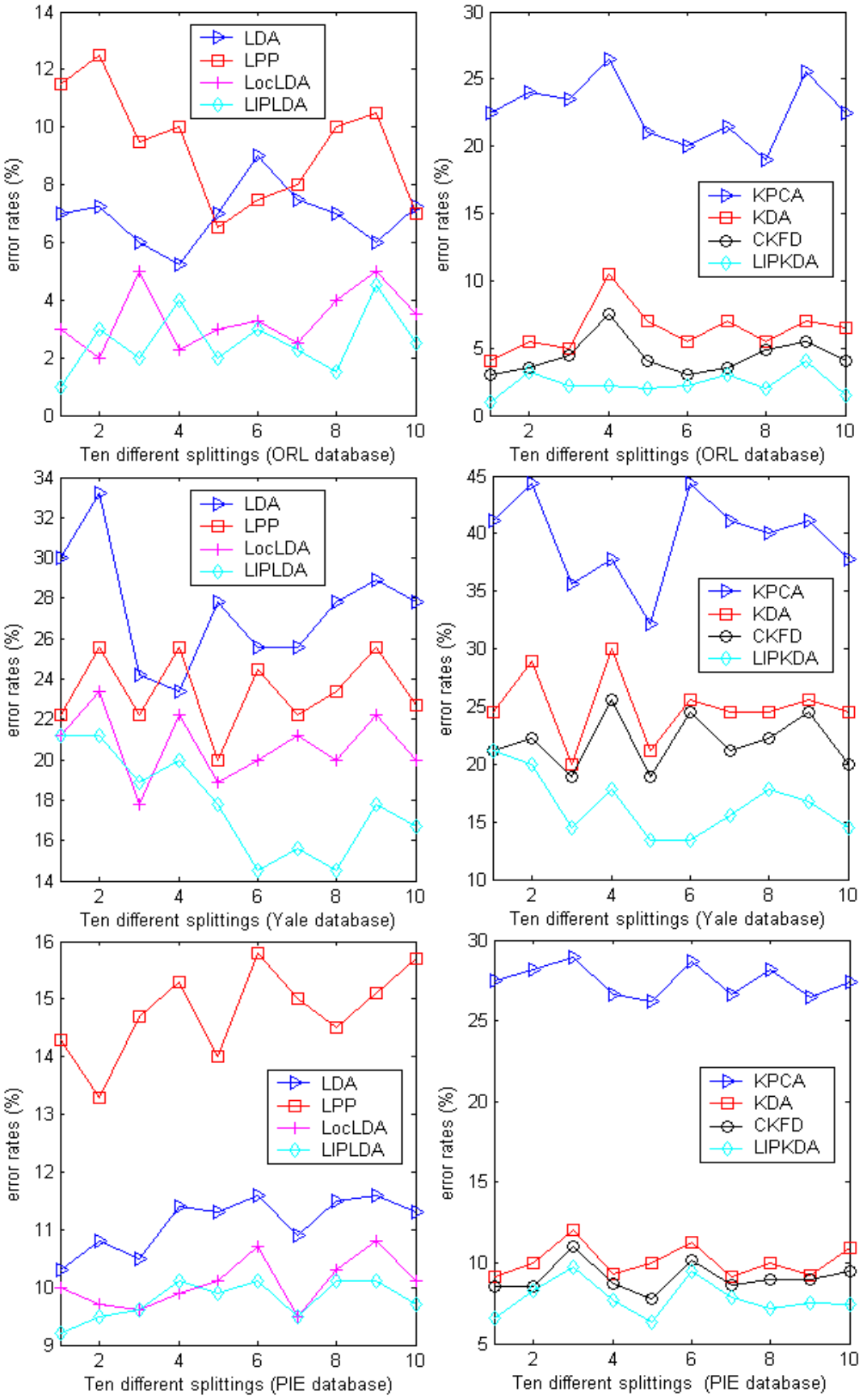

Fig.3 Comparison of eight DR methods in error rates on three face databases.

Table 1 Random partition on three databases for the phases of model selection and performance evaluation

Database	Classes	Different numbers for training (n per subject)	
	(C)	Model selection	Performance evaluation
Yale	40	5	2/3/5/6
ORL	15	5	2/3/5/6
PIE	68	60	30/60/90/120

Table 2 The average error rates (%) across 10 tests and their standard deviations (std) on Yale database

Training/Testing numbers		2/9	3/8	5/6	6/5
Linear methods	LDA	59.91 ± 3.19	39.51 ± 3.03	27.43 ± 2.90	22.82 ± 2.91
	LPP	44.50 ± 2.68	35.30 ± 2.68	23.40 ± 1.89	20.41 ± 2.17
	LocLDA	45.92 ± 2.50	32.40 ± 2.32	20.69 ± 1.68	18.32 ± 2.05
	LIPLDA	**45.94 ± 2.61**	**31.78 ± 2.94**	**17.82 ± 2.51**	**16.77 ± 2.38**
Non-linear methods	KPCA	63.72 ± 4.26	50.10 ± 4.04	39.55 ± 3.80	36.44 ± 3.25
	KDA	55.61 ± 3.15	37.51 ± 2.95	24.93 ± 3.00	21.83 ± 2.60
	CKFD	50.31 ± 2.68	33.10 ± 2.77	21.92 ± 2.36	18.22 ± 2.33
	LIPKDA	**45.32 ± 2.81**	**31.12 ± 3.15**	**16.49 ± 2.70**	**15.90 ± 2.41**

Table 3 The average error rates (%) across 10 tests and their standard deviations (std) on ORL database

Training/Testing numbers		2/8	3/7	5/5	6/4
Linear methods	LDA	27.51 ± 2.40	13.77 ± 2.61	6.93 ± 1.02	4.80 ± 1.88
	LPP	26.33 ± 2.95	16.80 ± 2.84	9.30 ± 1.99	7.52 ± 2.04
	LocLDA	17.44 ± 2.31	9.28 ± 1.52	3.35 ± 1.05	2.89 ± 1.46
	LIPLDA	**17.42 ± 2.16**	**9.08 ± 1.43**	**2.58 ± 1.08**	**2.43 ± 1.21**
Non-linear methods	KPCA	40.03 ± 3.05	29.05 ± 2.88	22.60 ± 2.35	21.94 ± 2.57
	KDA	26.61 ± 2.22	12.20 ± 2.04	6.35 ± 1.76	4.42 ± 1.78
	CKFD	17.80 ± 2.79	11.12 ± 2.15	4.34 ± 1.37	3.82 ± 1.93
	LIPKDA	**16.13 ± 1.90**	**7.20 ± 1.44**	**2.35 ± 0.87**	**2.11 ± 1.09**

Table 4 The average error rates (%) across 10 tests and their standard deviations (std) on PIE database

Training/Testing numbers		30/140	60/110	90/80	120/50
Linear methods	LDA	11.12 ± 0.47	5.43 ± 0.87	4.23 ± 0.81	3.70 ± 0.79
	LPP	14.77 ± 0.78	6.71 ± 0.69	4.10 ± 0.55	3.26 ± 0.54
	LocLDA	10.07 ± 0.43	5.22 ± 0.83	4.15 ± 0.78	3.61 ± 0.58
	LIPLDA	**9.78 ± 0.33**	**5.09 ± 0.93**	**3.99 ± 1.23**	**3.57 ± 0.84**
Non-linear methods	KPCA	27.49 ± 0.98	23.8 ± 0.88	22.30 ± 0.88	22.05 ± 0.69
	KDA	10.09 ± 1.01	5.50 ± 1.05	3.92 ± 0.83	3.22 ± 0.91
	CKFD	9.08 ± 0.93	5.04 ± 0.93	3.31 ± 0.82	2.90 ± 0.78
	LIPKDA	**7.81 ± 1.11**	**4.42 ± 1.02**	**3.16 ± 0.90**	**2.85 ± 0.83**

6.2 Experiment on Artificial Object Recognition

The COIL20 image database [42] contains 1440 images of 20 objects (72 images per subject). The images of each subject were taken every 5 degree

apart as the object was rotated on a turntable. Each image is of size 128×128. Fig.4 shows some examples from the database.

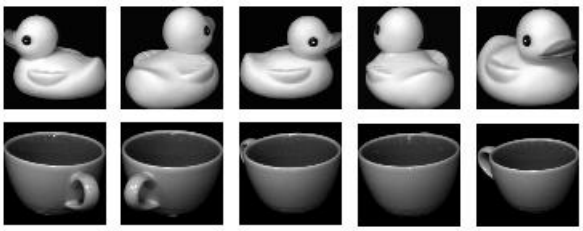

Fig.4 Sample images from COIL20 database

In our experiments, each image is resized to have a resolution of 64×64 and 36 samples are randomly chosen from each class for training, while the remaining 36 samples are used for testing. In this way, we run the system 20 times and obtain 10 different training and testing sample sets for both the phases of model selection and performance evaluation. The same methods described in Section 6.1 are used here for parameter selection.

The error rates of the random 10 different splits on COIL20 database with the tested eight methods are presented in Fig.5. The mean error rates and standard deviations of the 10 different partitions are reported in Table 5. From Fig.5 and Table 5, it can be seen that 1) KPCA has the lowest performance among all the tested methods and our proposed LIPLDA and LIPKDA algorithms consistently outperform other methods in linear and non-linear group, respectively. 2) Both the global and local geometrical information are effective for class classification, and fusing both of them can further improve recognition accuracy. Moreover, the results in Table 5 also prove that local intraclass geometrical variation contains more useful discriminant information than pure local geometrical information.

Table 5 The average error rates (%) across 10 tests and their standard deviations (std) on COIL20 database

	Linear methods				Non-linear methods			
	LDA	LPP	LocLDA	LIPLDA	KPCA	KDA	CKFD	LIPKDA
Error	8.96	8.97	6.00	**5.12**	25.63	7.85	5.86	**4.32**
rates	±1.41	±2.09	±1.65	**±1.19**	±2.22	±1.89	±1.63	**±1.64**

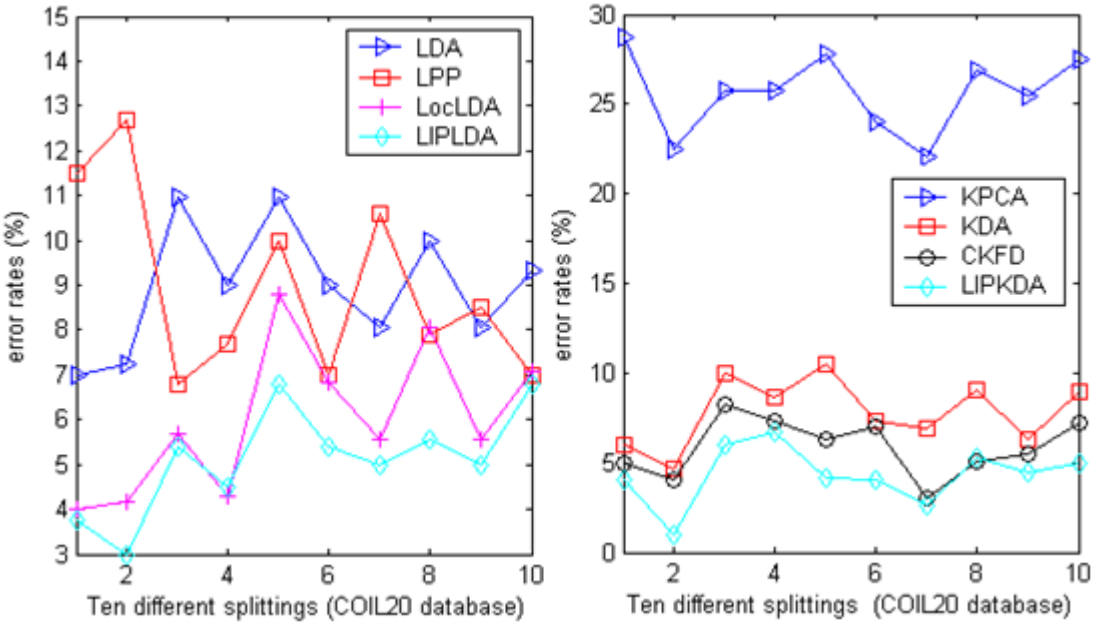

Fig.5 Comparison of eight DR methods in error rates on COIL20 database.

7 Conclusion, Discussion and Future Work

In this paper, we have proposed a new DR algorithm, called local intraclass geometrical variation preserving LDA, which integrates both global geometrical structure and local intraclass

geometrical variation for feature extraction and classification. We also show that the proposed algorithm can be extended to non-linear DR scenarios by applying the kernel trick. The new algorithm first casts LDA as a least squares problem and then uses a modified locality preserving projection as a regularization term to model the local intraclass geometrical variation. Extensive experimental results on Yale, ORL, PIE, and COIL20 image databases demonstrate the effectiveness of our approach.

Considering the results listed in Table 4 which show that in some cases when the training data size and data variance is large, the useful local structure information for class classification is corrupted by the densely and randomly distributed sample points, it is interesting to think about the possibility of the existence of "support" samples by which useful local structure information for class classification can be fully determined (hereinafter we call these samples the local-structure-supported vectors, or simply LSS vectors) and how to locate them. If LSS vectors exist, then by finding them in the training stage, two benefits can be expected: 1) LPP-related operation can be efficiently executed since only the LSS vectors are involved in the calculation and most of the "noisy" samples are neglected; 2) only using the useful local structure information for classification and disregarding the noisy information, the system performance can be further improved.

One of the tested methods, the CKFD algorithm, also achieves relatively good performance in our tests. Since CKFD makes full use of two kinds of discriminant information (regular and irregular, which extracted from the range space and null space of the within-class scatter matrix, respectively) while LDA and KDA only use regular discriminant information, it is also worth to explore the possibility of improving system performance by combing the idea of CKFD and local intraclass variation preserving.

Acknowledgement

This work was supported by the China Postdoctoral Science Foundation (Grant No. 2012M511804).

References:

[1] Jian-Bing Xia-Hou, Kun-Hong Liu, H. Murase, S.K. Nayar, "A GA Based Approach to Improving the ICA Based Classification Models for Tumor Classification", WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, vol.8, no.1, pp.28-38, 2011.

[2] A.M. Martinez, A.C. Kak, "PCA versus LDA", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no.2, pp.228-233, 2001.

[3] P.N. Belhumeur, J.P. Hepanha, 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] C. Liu, H. Wechsler, "Enhanced fisher linear discriminant models for face recognition", in: Proceedings of the International Conference on Pattern Recognition (ICPR), vol.2, pp.1368 – 1372, 1998.

[5] J.H. Friedman, "Regularized discriminant analysis", Journal of the American Statistical Association, vol.84, no.405, pp. 165–175 , 1989.

[6] J. Ye, Q. Li, "A two-stage linear discriminant analysis via QR decomposition", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.27, no.6, pp.929-941, 2005.

[7] H. Li, T. Jiang, K. Zhang, "Efficient and robust feature extraction by maximum margin criterion", IEEE Trans. Neural Network, vol.17, no.1, pp.157–165, 2006.

[8] J. Ye, R. Janardan, Q. Li, "Two-dimensional linear discriminant analysis", in: Proceedings of the Eighteenth Annual Conference on Neural Information Processing Systems, pp.1569–1576, 2004.

[9] K.Q. Weinberger, B.D. Packer, L.K. Saul, "Nonlinear dimensionality reduction by semi-definite programming and kernel matrix factorization", in: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, pp.381–388, 2005.

[10] Q. Gao, H. Xu, Y. Li, D. Xie, "Two-dimensional supervised local similarity and diversity projection", Pattern Recognition, vol.43, no.10, pp.3359–3363, 2010.

[11] C. Hou, C. Zhang, Y. Wu, Y. Jiao, "Stable local dimensionality reduction approaches", Pattern Recognition, vol.42, no.9, pp.2054–2066, 2009.

[12] S.T. Roweis, L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding", Science, vol.290, no.5500, pp.2323-2326, 2000.

[13] J.B. Tenenbaum, V. de Silva, J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction", Science, vol.290, no.5500, pp.2319-2323, 2000.

[14] M. Belkin, P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering", in Advances in Neural Information Processing Systems, vol.1, pp.585–592, 2002.
[15] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.29, no.1, pp.40-51, 2007.
[16] X. He and P. Niyogi, "Locality preserving projections", in Advances in Neural Information Processing Systems , 2003.
[17] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang, "Face Recognition Using Laplacianfaces", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.27, no.3, pp.328-340, 2005.
[18] W. Yu, X. Teng, C. Liu, "Face recognition using discriminant locality preserving projections", Image and Vision Computing, vol.24, pp. 239–248, 2006.
[19] Xin Shu, Yao Gao, Hongtao Lu, "Efficient linear discriminant analysis with locality preserving for face recognition", pattern recognition, vol.45, no.5, pp. 1892-1898, 2012.
[20] L. Zhu, S. Zhu, "Face recognition based on orthogonal discriminant locality preserving projections", Neurocomputing, vol.70, pp.1543–1546, 2007.
[21] J. Yang, D. Zhang, J. Yang, B. Niu, "Globally maximizing, locally minimizing: unsupervised discriminant projection with applications to face and palm biometrics", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.29, no.4, pp. 650–664, 2007.
[22] Wankou Yang, ChangyinSun, LeiZhang, "A multi-manifold discriminant analysis method for image feature extraction", Pattern Recognition, vol.44, no.8, pp. 1649–1657, 2011.
[23] W.K. Wong, H.T. Zhao, "Supervised optimal locality preserving projection", Pattern Recognition, vol.45, no.1, pp. 186–197, 2012.
[24] L. Yang, W. Gong, X. Gu, W. Li, Y. Liang, "Null space discriminant locality preserving projections for face recognition", Neurocomputing, vol.71, pp.3644–3649, 2008.
[25] S. Masashi, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis", Journal of Machine Learning Research, vol.8, pp.1027–1061, 2007.
[26] Yan Cui, Liya Fan, "A novel supervised dimensionality reduction algorithm: Graph-based Fisher analysis", Pattern Recognition, vol.45, no.4, pp. 1471–1481, 2012.
[27] M. Belkin, P. Niyogi, V. Sindhwani, "Manifold regularization: a geometric framework for learning from labeled and unlabeled examples", The Journal of Machine Learning Research, vol.7, pp. 2399–2434, 2006.
[28] J. Chen, J. Ye, Q. Li, "Integrating global and local structures: a least squares framework for dimensionality reduction", in: IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8.
[29] K. Fukunaga, "Introduction to Statistical Pattern Recognition", Academic Press, 2nd edition, 1990.
[30] D. Cai, X. He, J. Han, "Spectral regression: a unified approach for sparse subspace learning", in: Proceedings of the International Conference on Data Mining, 2007.
[31] C. Paige, M. Saunders, "LSQR: an algorithm for sparse linear equations and sparse least squares", ACM Transactions on Mathematical Software, vol.8, pp.43-71, 1982.
[32] C. Paige, M. Saunders, "Algorithm 583 LSQR: sparse linear equations and least squares problems", ACM Transactions on Mathematical Software, vol.8, pp.195-209, 1982.
[33] D. Cai, X. He, J. Han, "SRDA: an efficient algorithm for large-scale discriminant analysis", IEEE Trans. Knowledge and Data Engineering, vol.20, pp.1-12, 2008.
[34] B. Scholkopf, A. Smola, and K.R. Muller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem", Neural Computation, vol. 10, no. 5, pp.1299-1319, 1998.
[35] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K.-R.Muller, "Fisher Discriminant Analysis with Kernels", in Proc. IEEE Int'l Workshop Neural Networks for Signal Processing IX, pp. 41-48, Aug, 1999.
[36] G. Baudat and F. Anouar, "Generalized Discriminant Analysis Using a Kernel Approach", Neural Computation, vol.12, no.10, pp.2385-2404, 2000.
[37] J. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, "Face Recognition Using Kernel Direct Discriminant Analysis Algorithms", IEEE Trans. Neural Networks, vol.14, no.1, pp. 117-126, 2003.
[38] Jian Yang, Alejandro F. Frangi, Jing-yu Yang, David Zhang, and Zhong 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, 2005.

[39] C. Cortes, M. Mohri, A. Rostamizadeh, "Two-stage learning kernel algorithms", in: Proceedings of the 27th International Conference on Machine Learning, 2010.
[40] Yen-Yu Lin, Tyng-Luh Liu, and Chiou-Shann Fuh, "Multiple Kernel Learning for Dimensionality Reduction", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.33, no.6, pp.1147-1160, 2011.
[41] B.Scholkopf, A.J.Smola, "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond", The MIT Press, 2002.
[42] COIL20 image database, http://www1.cs. columbia.edu/CAVE/software/soft lib/coil-20.phpS.
[43] M.H. Yang, "Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods," in: Proc. Fifth IEEE Int'l Conf. Automatic Face and Gesture Recognition, pp. 215-220, May 2002.
[44] Yale Univ. Face Database, http://cvc.yale.edu/ projects/yalefacesyalefaces. htmlS.
[45] The ORL database of faces, http://www.cl. cam.ac.uk/Research/DTG/.
[46] T. Sim, S. Baker, M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) Database", in: Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition, May 2002.
[47] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An Introduction to Kernel-Based Learning Algorithms," IEEE Trans. Neural Networks, vol.12, no.2, pp.181-201, 2001.