Globally Maximizing, Locally Minimizing: Unsupervised Discriminant Projection with Applications to Face and Palm Biometrics

Jian Yang, David Zhang, *Senior Member*, *IEEE*, Jing-yu Yang, and Ben Niu

Abstract—This paper develops an unsupervised discriminant projection (UDP) technique for dimensionality reduction of high-dimensional data in small sample size cases. UDP can be seen as a linear approximation of a multimanifolds-based learning framework which takes into account both the local and nonlocal quantities. UDP characterizes the local scatter as well as the nonlocal scatter, seeking to find a projection that simultaneously maximizes the nonlocal scatter and minimizes the local scatter. This characteristic makes UDP more intuitive and more powerful than the most up-to-date method, Locality Preserving Projection (LPP), which considers only the local scatter for clustering or classification tasks. The proposed method is applied to face and palm biometrics and is examined using the Yale, FERET, and AR face image databases and the PolyU palmprint database. The experimental results show that UDP consistently outperforms LPP and PCA and outperforms LDA when the training sample size per class is small. This demonstrates that UDP is a good choice for real-world biometrics applications.

Index Terms—Dimensionality reduction, feature extraction, subspace learning, Fisher linear discriminant analysis (LDA), manifold learning, biometrics, face recognition, palmprint recognition.

1 Introduction

Dimensionality reduction is the construction of a meaningful low-dimensional representation of high-dimensional data. Since there are large volumes of high-dimensional data in numerous real-world applications, dimensionality reduction is a fundamental problem in many scientific fields. From the perspective of pattern recognition, dimensionality reduction is an effective means of avoiding the "curse of dimensionality" [1] and improving the computational efficiency of pattern matching.

Researchers have developed many useful dimensionality reduction techniques. These techniques can be broadly categorized into two classes: linear and nonlinear. Linear dimensionality reduction seeks to find a meaningful low-dimensional subspace in a high-dimensional input space. This subspace can provide a compact representation of higher-dimensional data when the structure of data embedded in the input space is linear. PCA and LDA are two well-known linear subspace learning methods which have been extensively used in pattern recognition and computer vision areas and have become the most popular techniques for face recognition and other biometrics [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [39].

Linear models, however, may fail to discover essential data structures that are nonlinear. A number of nonlinear dimensionality reduction techniques have been developed to address this problem, with two in particular attracting wide attention: kernel-based techniques and manifold learning-based techniques. The basic idea of kernel-based techniques is to implicitly map observed patterns into potentially much higher dimensional feature vectors by using a nonlinear mapping determined by a kernel. This makes it possible for the nonlinear structure of data in observation space to become linear in feature space, allowing the use of linear techniques to deal with the data. The representative techniques are kernel principal component analysis (KPCA) [15] and kernel Fisher discriminant (KFD) [16], [17]. Both have proven to be effective in many real-world applications [18], [19], [20].

In contrast with kernel-based techniques, the motivation of manifold learning is straightforward as it seeks to directly find the intrinsic low-dimensional nonlinear data structures hidden in observation space. The past few years have seen many manifold-based learning algorithms for discovering intrinsic low-dimensional embedding of data proposed. Among the most well-known are isometric feature mapping (ISOMAP) [22], local linear embedding (LLE) [23], and Laplacian Eigenmap [24]. Some experiments have shown that these methods can find perceptually meaningful embeddings for face or digit images. They also yielded impressive results on other artificial and real-world data sets. Recently, Yan et al. [33] proposed a general dimensionality reduction framework called *graph embedding*. LLE, ISOMAP, and Laplacian Eigenmap can all be reformulated as a unified model in this framework.

- *J. Yang is with the Biometric Research Centre, Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong and the Department of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, P.R. China. E-mail: csjyang@comp.polyu.edu.hk.*
- *D. Zhang and B. Niu are with the Biometric Research Centre, Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong. E-mail: {csdzhang, csniuben}@comp.polyu.edu.hk.*
- *J.-y. Yang is with the Department of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, P.R. China. E-mail: yangjy@mail.njust.edu.cn.*

Manuscript received 17 Jan. 2006; revised 5 June 2006; accepted 26 Sept. 2006; published online 18 Jan. 2007.
Recommended for acceptance by S. Prabhakar, J. Kittler, D. Maltoni, L. O'Gorman, and T. Tan.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org and reference IEEECS Log Number TPAMISI-0021-0106.
Digital Object Identifier no. 10.1109/TPAMI.2007.1008.

One problem with current manifold learning techniques is that they might be unsuitable for pattern recognition tasks. There are two reasons for this. First, as it is currently conceived, manifold learning is limited in that it is modeled based on a characterization of "locality," a modeling that has no direct connection to classification. This is unproblematic for existing manifold learning algorithms as they seek to model a simple manifold, for example, to recover an embedding of one person's face images [21], [22], [23]. However, if face images do exist on a manifold, different persons' face images could lie on different manifolds. To recognize faces, it would be necessary to distinguish between images from different manifolds. For achieving an optimal recognition result, the recovered embeddings corresponding to different face manifolds should be as separate as possible in the final embedding space. This poses a problem that we might call "classification-oriented multimanifolds learning." This problem cannot be addressed by current manifold learning algorithms, including some supervised versions [25], [26], [27] because they are all based on the characterization of "locality." The local quantity suffices for modeling a single manifold, but does not suffice for modeling multimanifolds for classification purposes. To make different embeddings corresponding to different classes mutually separate, however, it is crucial to have the "nonlocal" quantity, which embodies the distance between embeddings. In short, it is necessary to characterize the "nonlocality" when modeling multimanifolds.

The second reason why most manifold learning algorithms, for example, ISOMAP, LLE, and Laplacian Eigenmap, are unsuitable for pattern recognition tasks is that they can yield an embedding directly based on the training data set but, because of the implicitness of the nonlinear map, when applied to a new sample, they cannot find the sample's image in the embedding space. This limits the applications of these algorithms to pattern recognition problems. Although some research has shown that it is possible to construct an explicit map from input space to embedding space [28], [29], [30], the effectiveness of these kinds of maps on real-world classification problems still needs to be demonstrated.

Recently, He et al. [31], [32] proposed Locality Preserving Projections (LPP), which is a linear subspace learning method derived from Laplacian Eigenmap. In contrast to most manifold learning algorithms, LPP possesses the remarkable advantage that it can generate an explicit map. This map is linear and easily computable, like that of PCA or LDA. It is also effective, yielding encouraging results on face recognition tasks. Yet, as it is modeled on the basis of "locality," LPP, like most manifold learning algorithms, has the weakness of having no direct connection to classification. The objective function of LPP is to minimize the local quantity, i.e., the local scatter of the projected data. In some cases, this criterion cannot be guaranteed to yield a good projection for classification purposes. Assume, for example, that there exist two clusters of two-dimensional samples scattering uniformly in two ellipses C_1 and C_2, as shown in Fig. 1. If the locality radius δ is set as the length of the semimajor axis of the larger ellipse, the direction w_1 is a nice projection according to the criterion of LPP since, after all samples are projected onto w_1, the local scatter is minimal. But, it is obvious that w_1 is not good in terms of classification; the projected samples overlap in this direction. This example also shows that the nonlocal quantity, i.e., the intercluster scatter, may provide crucial information for discrimination. In this paper, we will address this issue and explore more effective projections for classification purposes.

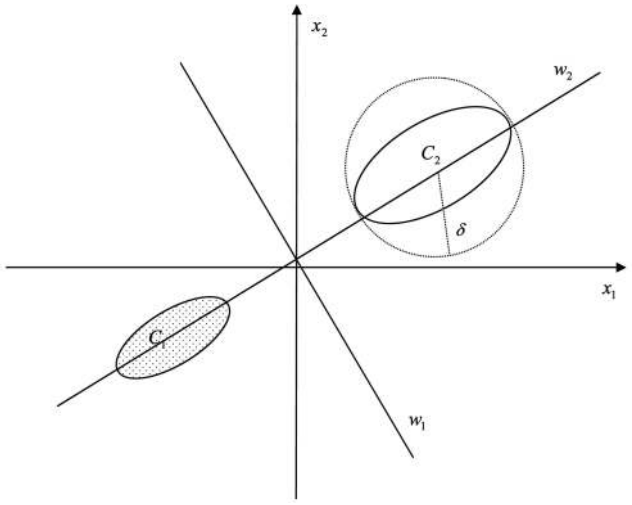


Fig. 1. Illustration of two clusters of samples in two-dimensional space and the projection directions.

Motivated by the idea of classification-oriented multimanifolds learning, we consider two quantities, local and nonlocal, at the same time in the modeling process. It should be pointed out that we don't attempt to build a framework for multimanifolds-based learning in this paper (although it is very interesting). We are more interested in its linear approximation, i.e., finding a simple and practical linear map for biometrics applications. To this end, we first present the techniques to characterize the local and nonlocal scatters of data. Then, based on this characterization, we propose a criterion which seeks to maximize the ratio of the nonlocal scatter to the local scatter. This criterion, similar to the classical Fisher criterion, is a Rayleigh quotient in form. Thus, it is not hard to find its optimal solutions by solving a generalized eigen-equation. Since the proposed method does not use the class-label information of samples in the learning process, this method is called the unsupervised discriminant projection (UDP), in contrast with the supervised discriminant projection of LDA.

In contrast with LPP, UDP has direct relations to classification since it utilizes the information of the "nonlocality." Provided that each cluster of samples in the observation space is exactly within a local neighbor, UDP can yield an optimal projection for clustering in the projected space, while LPP cannot. As shown in Fig. 1, w_2 is a good projection direction according the criterion of UDP, which is more discriminative than w_1. In addition, UDP will be demonstrated to be more effective than LPP in real-world biometrics applications, based on our experiments with three face image databases and one palmprint database.

In the literature, besides LPP, there are two methods most relevant to ours. One is Marginal Fisher Analysis (MFA) presented by Yan et al. [33] and the other is Local Discriminant Embedding (LDE) suggested by Chen et al. [34]. The two methods are very similar in formulation. Both of them combine *locality* and *class label* information to represent the intraclass compactness and interclass separability. So, MFA and LDE can be viewed as supervised variants of LPP or as localized variants of LDA since both methods focus on the

characterization of *intraclass locality* and *interclass locality*. In contrast, the proposed UDP retains the unsupervised characteristic of LPP and seeks to combine *locality* and *globality* information for discriminator design.

The remainder of this paper is organized as follows: Section 2 outlines PCA and LDA. Section 3 develops the idea of UDP and the relevant theory and algorithm. Section 4 describes a kernel weighted version of UDP. Section 5 discusses the relations between UDP and LDA/LPP. Section 6 describes some biometrics applications and the related experiments. Section 7 offers our conclusions.

2 Outline of PCA and LDA

2.1 PCA

PCA seeks to find a projection axis such that the *global scatter* is maximized after the projection of samples. The *global scatter* can be characterized by the mean square of the Euclidean distance between any pair of the projected sample points. Specifically, given a set of M training samples (pattern vectors) $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M$ in $\mathbb{R}^n$, we get their images $y_1, y_2, \cdots, y_M$ after the projection onto the projection axis $\mathbf{w}$. The global scatter is defined by

$$J_T(\mathbf{w}) \triangleq \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(y_i - y_j)^2. \tag{1}$$

It follows that

$$\begin{aligned} J_T(\mathbf{w}) &= \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x}_j)^2 \\ &= \mathbf{w}^T\left[\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T\right]\mathbf{w}. \end{aligned} \tag{2}$$

Let us denote

$$\mathbf{S}_T = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{3}$$

and the mean vector $\mathbf{m}_0 = \frac{1}{M}\sum_{j=1}^{M}\mathbf{x}_j$. Then, it can be proven that

$$\begin{aligned} \mathbf{S}_T &= \frac{1}{MM}\left[M\sum_{i=1}^{M}\mathbf{x}_i\mathbf{x}_i^T - \left(\sum_{i=1}^{M}\mathbf{x}_i\right)\left(\sum_{j=1}^{M}\mathbf{x}_j^T\right)\right] \\ &= \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i - \mathbf{m}_0)(\mathbf{x}_i - \mathbf{m}_0)^T. \end{aligned} \tag{4}$$

Equation (4) indicates that $\mathbf{S}_T$ is essentially the covariance matrix of data. So, the projection axis $\mathbf{w}$ that maximizes (2) can be selected as the eigenvector of $\mathbf{S}_T$ corresponding to the largest eigenvalue. Similarly, we can obtain a set of projection axes of PCA by selecting the d eigenvectors of $\mathbf{S}_T$ corresponding to the d largest eigenvalues.

2.2 LDA

LDA seeks to find a projection axis such that the Fisher criterion (i.e., the ratio of the *between-class scatter* to the *within-class scatter*) is maximized after the projection of samples. The between-class and within-class scatter matrices $\mathbf{S}_B$ and $\mathbf{S}_w$ are defined by

$$\mathbf{S}_B = \frac{1}{M}\sum_{i=1}^{c} l_i(\mathbf{m}_i - \mathbf{m}_0)(\mathbf{m}_i - \mathbf{m}_0)^{\mathrm{T}}, \tag{5}$$

$$\mathbf{S}_W = \sum_{i=1}^{c}\frac{l_i}{M}\mathbf{S}_W^{(i)} = \frac{1}{M}\sum_{i=1}^{c}\sum_{j=1}^{l_i}(\mathbf{x}_{ij} - \mathbf{m}_i)(\mathbf{x}_{ij} - \mathbf{m}_i)^{\mathrm{T}}, \tag{6}$$

where $\mathbf{x}_{ij}$ denotes the jth training sample in class i, c is the number of classes, l_i is the number of training samples in class i, $\mathbf{m}_i$ is the mean of the training samples in class i, and $\mathbf{S}_W^{(i)}$ denotes the covariance matrix of samples in class i.

It is easy to show that $\mathbf{S}_B$ and $\mathbf{S}_W$ are both nonnegative definite matrix and satisfy $\mathbf{S}_T = \mathbf{S}_B + \mathbf{S}_W$.

The Fisher criterion is defined by

$$J_F(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}}. \tag{7}$$

The stationary points of $J_F(\mathbf{w})$ are the generalized eigenvectors $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d$ of $\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w}$ corresponding to the d largest eigenvalues. These stationary points form the coordinate system of LDA.

3 Unsupervised Discriminant Projection (UDP)

3.1 Basic Idea of UDP

As discussed in Section 1, the locality characterization-based model does not guarantee a good projection for classification purposes. To address this, we will introduce the concept of nonlocality and give the characterizations of the nonlocal scatter and the local scatter. This will allow us to obtain a concise criterion for feature extraction by maximizing the ratio of nonlocal scatter to local scatter.

3.1.1 Characterize the Local Scatter

Recall that, in PCA, in order to preserve the global geometric structure of data in a transformed low-dimensional space, account is taken of the global scatter of samples. Correspondingly, if we aim to discover the local structure of data, we should take account of the local scatter (or *intralocality scatter*) of samples. The local scatter can be characterized by the mean square of the Euclidean distance between any pair of the projected sample points that are within any local δ-neighborhood ($\delta > 0$). Specifically, two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are viewed within a local δ-neighborhood provided that $||\mathbf{x}_i - \mathbf{x}_j||^2 < \delta$. Let us denote the set $\mathrm{U}^\delta = \{(i,j)\big|||\mathbf{x}_i - \mathbf{x}_j||^2 < \delta\}$. After the projection of $\mathbf{x}_i$ and $\mathbf{x}_j$ onto a direction $\mathbf{w}$, we get their images y_i and y_j. The local scatter is then defined by

$$J_L(\mathbf{w}) \triangleq \frac{1}{2}\frac{1}{M_L}\sum_{(i,j)\in\mathrm{U}^\delta}(y_i - y_j)^2 \propto \frac{1}{2}\frac{1}{MM}\sum_{(i,j)\in\mathrm{U}^\delta}(y_i - y_j)^2, \tag{8}$$

where M_L is the number of sample pairs satisfying $||\mathbf{x}_i - \mathbf{x}_j||^2 < \delta$.

Let us define the adjacency matrix $\mathbf{H}$, whose elements are given below:

$$H_{ij} = \begin{cases} 1, & ||\mathbf{x}_i - \mathbf{x}_j||^2 < \delta \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

It is obvious that the adjacency matrix $\mathbf{H}$ is a symmetric matrix. By virtue of the adjacency matrix $\mathbf{H}$, (8) can be rewritten by[1]

$$J_L(\mathbf{w}) = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}(y_i - y_j)^2. \tag{10}$$

It follows from (10) that

$$\begin{aligned} J_L(\mathbf{w}) &= \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}(\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x}_j)^2 \\ &= \mathbf{w}^T\left[\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T\right]\mathbf{w} \\ &= \mathbf{w}^T\mathbf{S}_L\mathbf{w}, \end{aligned} \tag{11}$$

where

$$\mathbf{S}_L = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T. \tag{12}$$

$\mathbf{S}_L$ is called the local scatter (covariance) matrix.

Due to the symmetry of $\mathbf{H}$, we have

$$\begin{aligned} \mathbf{S}_L &= \frac{1}{2}\frac{1}{MM}\left(\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}\mathbf{x}_i\mathbf{x}_i^T + \sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}\mathbf{x}_j\mathbf{x}_j^T - 2\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}\mathbf{x}_i\mathbf{x}_j^T\right) \\ &= \frac{1}{MM}\left(\sum_{i=1}^{M} D_{ii}\mathbf{x}_i\mathbf{x}_i^T - \sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}\mathbf{x}_i\mathbf{x}_j^T\right) \\ &= \frac{1}{MM}\left(\mathbf{X}\mathbf{D}\mathbf{X}^T - \mathbf{X}\mathbf{H}\mathbf{X}^T\right) \\ &= \frac{1}{MM}\mathbf{X}\mathbf{L}\mathbf{X}^T, \end{aligned} \tag{13}$$

where $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M)$ and $\mathbf{D}$ is a diagonal matrix whose elements on diagonal are column (or row since $\mathbf{H}$ is a symmetric matrix) sum of $\mathbf{H}$, i.e., $D_{ii} = \sum_{j=1}^{M} H_{ij}$. $\mathbf{L} = \mathbf{D} - \mathbf{H}$ is called the local scatter kernel (LSK) matrix in this paper (this matrix is called the Laplacian matrix in [24]).

It is obvious that $\mathbf{L}$ and $\mathbf{S}_L$ are both real symmetric matrices. From (11) and (13), we know that $\mathbf{w}^T\mathbf{S}_L\mathbf{w} \geq 0$ for any nonzero vector $\mathbf{w}$. So, the local scatter matrix $\mathbf{S}_L$ must be nonnegative definite.

In the above discussion, we use δ-neighborhoods to characterize the "locality" and the local scatter. This way is geometrically intuitive but unpopular because, in practice, it is hard to choose a proper neighborhood radius δ. To avoid this difficulty, the method of K-nearest neighbors is always used instead in real-world applications. The K-nearest neighbors method can determine the following adjacency matrix $\mathbf{H}$, with elements given by:

$$H_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_j \text{ is among K nearest nieghbors of } \mathbf{x}_i \\ & \quad \textit{and } \mathbf{x}_i \text{ is among K nearest nieghbors of } \mathbf{x}_j \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

1. In (8), the only difference between expressions in the middle and on the right is a coefficient. This difference is meaningless for the characterization of the scatter. For convenience, we use the expression on the right. The same operation is used in (15).

The local scatter can be characterized similarly by a K-nearest neighbor adjacency matrix if (9) is replaced by (14).

3.1.2 Characterize the Nonlocal Scatter

The nonlocal scatter (i.e., the interlocality scatter) can be characterized by the mean square of the Euclidean distance between any pair of the projected sample points that are outside any local δ-neighborhoods ($\delta > 0$). The nonlocal scatter is defined by

$$J_N(\mathbf{w}) \triangleq \frac{1}{2}\frac{1}{M_N}\sum_{(i,j)\notin \mathrm{U}^\delta}(y_i - y_j)^2 \propto \frac{1}{2}\frac{1}{MM}\sum_{(i,j)\notin \mathrm{U}^\delta}(y_i - y_j)^2, \tag{15}$$

where M_N is the number of sample pairs satisfying $||\mathbf{x}_i - \mathbf{x}_j||^2 \geq \delta$.

By virtue of the adjacency matrix $\mathbf{H}$ in (9) or (14), the nonlocal scatter can be rewritten by

$$J_N(\mathbf{w}) = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(1 - H_{ij})(y_i - y_j)^2. \tag{16}$$

It follows from (16) that

$$\begin{aligned} J_N(\mathbf{w}) &= \mathbf{w}^T\left[\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(1 - H_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T\right]\mathbf{w} \\ &= \mathbf{w}^T\mathbf{S}_N\mathbf{w}, \end{aligned} \tag{17}$$

where

$$\mathbf{S}_N = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(1 - H_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T. \tag{18}$$

$\mathbf{S}_N$ is called the nonlocal scatter (covariance) matrix. It is easy to show $\mathbf{S}_N$ is also a nonnegative definite matrix. And, it follows that

$$\begin{aligned} \mathbf{S}_N &= \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(1 - H_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \\ &= \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \\ &\quad - \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \\ &= \mathbf{S}_T - \mathbf{S}_L. \end{aligned}$$

That is, $\mathbf{S}_T = \mathbf{S}_L + \mathbf{S}_N$. Thus, we have $J_T(\mathbf{w}) = J_L(\mathbf{w}) + J_N(\mathbf{w})$.

3.1.3 Determine a Criterion: Maximizing the Ratio of Nonlocal Scatter to Local Scatter

The technique of Locality Preserving Projection (LPP) [31] seeks to find a linear subspace which can preserve the local structure of data. The objective of LPP is actually to minimize the local scatter $J_L(\mathbf{w})$. Obviously, the projection direction determined by LPP can ensure that, if samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are close, their projections y_i and y_j are close as well. But, LPP cannot guarantee that, if samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are not close, their projections y_i and y_j are not either. This means that it may happen that two mutually distant samples belonging to

different classes may result in close images after the projection of LPP. Therefore, LPP does not necessarily yield a good projection suitable for classification.

For the purpose of classification, we try is to find a projection which will draw the close samples closer together while simultaneously making the mutually distant samples even more distant from each other. From this point of view, a desirable projection should be the one that, at the same time, minimizes the local scatter $J_L(\mathbf{w})$ and maximizes the nonlocal scatter $J_N(\mathbf{w})$. As it happens, we can obtain just such a projection by maximizing the following criterion:

$$J(\mathbf{w}) = \frac{J_N(\mathbf{w})}{J_L(\mathbf{w})} = \frac{\mathbf{w}^T\mathbf{S}_N\mathbf{w}}{\mathbf{w}^T\mathbf{S}_L\mathbf{w}}. \tag{19}$$

Since $J_T(\mathbf{w}) = J_L(\mathbf{w}) + J_N(\mathbf{w})$ and $\mathbf{S}_T = \mathbf{S}_L + \mathbf{S}_N$, the above criterion is equivalent to

$$J_e(\mathbf{w}) = \frac{J_T(\mathbf{w})}{J_L(\mathbf{w})} = \frac{\mathbf{w}^T\mathbf{S}_T\mathbf{w}}{\mathbf{w}^T\mathbf{S}_L\mathbf{w}}. \tag{20}$$

The criterion in (20) indicates that we can find the projection by at the same time globally maximizing (maximizing the global scatter) and locally minimizing (minimizing the local scatter).

The criterion in (19) or (20) is formally similar to the Fisher criterion in (7) since they are both Rayleigh quotients. Differently, the matrices $\mathbf{S}_L$ and $\mathbf{S}_N$ in (19) can be constructed without knowing the class-label of samples, while $\mathbf{S}_B$ and $\mathbf{S}_W$ in (7) cannot be so constructed. This means Fisher discriminant projection is supervised, while the projection determined by $J(\mathbf{w})$ can be obtained in an unsupervised manner. In this paper, then, this projection is called an Unsupervised Discriminant Projection (UDP).

3.2 Algorithmic Derivations of UDP in Small Sample Size Cases

If the local scatter matrix $\mathbf{S}_L$ is nonsingular, the criterion in (19) can be maximized directly by calculating the generalized eigenvectors of the following generalized eigen-equation:

$$\mathbf{S}_N\mathbf{w} = \lambda\mathbf{S}_L\mathbf{w}. \tag{21}$$

The projection axes of UDP can be selected as the generalized eigenvectors $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d$ of $\mathbf{S}_N\mathbf{w} = \lambda\mathbf{S}_L\mathbf{w}$ corresponding to d largest positive eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$.

In real-world biometrics applications of such face and palm recognition, however, $\mathbf{S}_L$ is always singular due to the limited number of training samples. In such cases, the classical algorithm cannot be used directly to solve the generalized eigen-equation. In addition, from (12) and (18), we know $\mathbf{S}_L$ and $\mathbf{S}_N$ are both $n \times n$ matrices (where n is the dimension of the image vector space). It is computationally very expensive to construct these large-sized matrices in the high-dimensional input space. Fortunately, we can avoid these difficulties by virtue of the theory we built for LDA (or KFD) in small sample size cases [9], [20]. Based on this theory, the local and nonlocal scatter matrices can be constructed using the PCA-transformed low-dimensional data and the singularity difficulty can be avoided. The relevant theory is given below.

Suppose $\beta_1, \beta_2, \cdots, \beta_n$ are n orthonormal eigenvectors of $\mathbf{S}_T$ and the first m ($m = \mathrm{rank}(\mathbf{S}_T)$) eigenvectors correspond to positive eigenvalues $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m$. Define the subspace $\Psi_T = \mathrm{span}\{\beta_1, \cdots, \beta_m\}$ and denote its orthogonal complement $\Psi_T^\perp = \mathrm{span}\{\beta_{m+1}, \cdots, \beta_n\}$. Obviously, Ψ_T is the range space of $\mathbf{S}_T$ and $\Psi_T^\perp$ is the corresponding null space.

Lemma 1 [4], [36]. *Suppose that $\mathbf{A}$ is an $n \times n$ nonnegative definite matrix and φ is an n-dimensional vector, then $\varphi^T\mathbf{A}\varphi = 0$ if and only if $\mathbf{A}\varphi = 0$.*

Since $\mathbf{S}_L$, $\mathbf{S}_N$, and $\mathbf{S}_T$ are all nonnegative definite and $\mathbf{S}_T = \mathbf{S}_L + \mathbf{S}_N$, it's easy to get:

Lemma 2. *If $\mathbf{S}_T$ is singular, $\varphi^T\mathbf{S}_T\varphi = 0$ if and only if $\varphi^T\mathbf{S}_L\varphi = 0$ and $\varphi^T\mathbf{S}_N\varphi = 0$.*

Since $\mathbb{R}^n = \mathrm{span}\{\beta_1, \beta_2, \cdots, \beta_n\}$, for an arbitrary $\varphi \in \mathbb{R}^n$, φ can be denoted by

$$\varphi = k_1\beta_1 + \cdots + k_m\beta_m + k_{m+1}\beta_{m+1} + \cdots + k_n\beta_n. \tag{22}$$

Let $\mathbf{w} = k_1\beta_1 + \cdots + k_m\beta_m$ and $\mathbf{u} = k_{m+1}\beta_{m+1} + \cdots + k_n\beta_n$, then, from the definition of Ψ_T and $\Psi_T^\perp$, φ can be denoted by $\varphi = \mathbf{w} + \mathbf{u}$, where $\mathbf{w} \in \Psi_T$ and $\mathbf{u} \in \Psi_T^\perp$.

Definition 1. *For an arbitrary $\varphi \in \mathbb{R}^n$, φ can be denoted by $\varphi = \mathbf{w} + \mathbf{u}$, where $\mathbf{w} \in \Psi_T$ and $\mathbf{u} \in \Psi_T^\perp$. The compression mapping $L : \mathbb{R} \to \Psi_T$ is defined by $\varphi = \mathbf{w} + \mathbf{u} \to \mathbf{w}$.*

It is easy to verify that L is a linear transformation from $\mathbb{R}^n$ to its subspace Ψ_T.

Theorem 1. *Under the compression mapping $L : \mathbb{R}^n \to \Psi_T$ determined by $\varphi = \mathbf{w} + \mathbf{u} \to \mathbf{w}$, the UDP criterion satisfies $J(\varphi) = J(\mathbf{w})$.*

Proof. Since $\Psi_T^\perp$ is the null space of $\mathbf{S}_T$, for any $\mathbf{u} \in \Psi_T^\perp$, we have $\mathbf{u}^{\mathrm{T}}\mathbf{S}_T\mathbf{u} = 0$.

From Lemma 2, it follows that $\mathbf{u}^{\mathrm{T}}\mathbf{S}_L\mathbf{u} = 0$. Since $\mathbf{S}_L$ is a nonnegative definite matrix, we have $\mathbf{S}_L\mathbf{u} = 0$ by Lemma 1. Hence,

$$\varphi^{\mathrm{T}}\mathbf{S}_L\varphi = \mathbf{w}^{\mathrm{T}}\mathbf{S}_L\mathbf{w} + 2\mathbf{w}^{\mathrm{T}}\mathbf{S}_L\mathbf{u} + \mathbf{u}^{\mathrm{T}}\mathbf{S}_L\mathbf{u} = \mathbf{w}^{\mathrm{T}}\mathbf{S}_L\mathbf{w}.$$

Similarly, it can be derived that

$$\varphi^{\mathrm{T}}\mathbf{S}_N\varphi = \mathbf{w}^{\mathrm{T}}\mathbf{S}_N\mathbf{w} + 2\mathbf{w}^{\mathrm{T}}\mathbf{S}_N\mathbf{u} + \mathbf{u}^{\mathrm{T}}\mathbf{S}_N\mathbf{u} = \mathbf{w}^{\mathrm{T}}\mathbf{S}_N\mathbf{w}.$$

Therefore, $J(\varphi) = J(\mathbf{w})$. □

According to Theorem 1, we can conclude that the optimal projection axes can be derived from Ψ_T without any loss of effective discriminatory information with respect to the UDP criterion. From linear algebra theory, Ψ_T is isomorphic to an m-dimensional Euclidean space $\mathbb{R}^m$ and the corresponding *isomorphic mapping* is

$$\mathbf{w} = \mathbf{P}\mathbf{v}, \text{where } \mathbf{P} = (\beta_1, \beta_2, \cdots, \beta_m), \mathbf{v} \in \mathbb{R}^m, \tag{23}$$

which is a one-to-one mapping from $\mathbb{R}^m$ onto Ψ_T.

From the isomorphic mapping $\mathbf{w} = \mathbf{P}\mathbf{v}$, the UDP criterion function $J(\mathbf{w})$ becomes

$$J(\mathbf{w}) = \frac{\mathbf{v}^{\mathrm{T}}(\mathbf{P}^{\mathrm{T}}\mathbf{S}_N\mathbf{P})\mathbf{v}}{\mathbf{v}^{\mathrm{T}}(\mathbf{P}^{\mathrm{T}}\mathbf{S}_L\mathbf{P})\mathbf{v}} = \frac{\mathbf{v}^{\mathrm{T}}\tilde{\mathbf{S}}_N\mathbf{v}}{\mathbf{v}^{\mathrm{T}}\tilde{\mathbf{S}}_L\mathbf{v}} \triangleq \tilde{J}(\mathbf{v}), \tag{24}$$

where $\tilde{\mathbf{S}}_N = \mathbf{P}^{\mathrm{T}}\mathbf{S}_N\mathbf{P}$ and $\tilde{\mathbf{S}}_L = \mathbf{P}^{\mathrm{T}}\mathbf{S}_L\mathbf{P}$. It is easy to prove that $\tilde{\mathbf{S}}_N$ and $\tilde{\mathbf{S}}_L$ are both $m \times m$ semipositive definite matrices. This means $\tilde{J}(\mathbf{v})$ is a function of a generalized Rayleigh quotient like $J(\mathbf{w})$.

By the property of isomorphic mapping and (24), the following theorem holds:

Theorem 2. *Let $\mathbf{w} = \mathbf{P}\mathbf{v}$ be an isomorphic mapping from $\mathbb{R}^m$ onto Ψ_T. Then, $\mathbf{w}^* = \mathbf{P}\mathbf{v}^*$ is the stationary point of the UDP*

criterion function $J(\mathbf{w})$ if and only if $\mathbf{v}^*$ is the stationary point of the function $\tilde{J}(\mathbf{v})$.

From Theorem 2, it is easy to draw the following conclusion:

Proposition 1. *If $\mathbf{v}_1, \cdots, \mathbf{v}_d$ are the generalized eigenvectors of $\tilde{\mathbf{S}}_N\mathbf{v} = \lambda\tilde{\mathbf{S}}_L\mathbf{v}$ corresponding to the d largest eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d > 0$, then, $\mathbf{w}_1 = \mathbf{P}\mathbf{v}_1, \cdots, \mathbf{w}_d = \mathbf{P}\mathbf{v}_d$ are the optimal projection axes of UDP.*

Now, our work is to find the generalized eigenvectors of $\tilde{\mathbf{S}}_N\mathbf{v} = \lambda\tilde{\mathbf{S}}_L\mathbf{v}$. First of all, let us consider the construction of $\tilde{\mathbf{S}}_L$ and $\tilde{\mathbf{S}}_N$. From (13), we know $\mathbf{S}_L = \frac{1}{MM}\mathbf{X}\mathbf{L}\mathbf{X}^T$. Thus,

$$\tilde{\mathbf{S}}_L = \mathbf{P}^{\mathrm{T}}\mathbf{S}_L\mathbf{P} = \frac{1}{MM}(\mathbf{P}^{\mathrm{T}}\mathbf{X})\mathbf{L}(\mathbf{P}^{\mathrm{T}}\mathbf{X})^{\mathrm{T}} = \frac{1}{MM}\tilde{\mathbf{X}}\mathbf{L}\tilde{\mathbf{X}}^{\mathrm{T}}, \quad (25)$$

where $\tilde{\mathbf{X}} = \mathbf{P}^{\mathrm{T}}\mathbf{X}$. Since $\mathbf{P} = (\beta_1, \cdots, \beta_m)$ and $\beta_1, \cdots, \beta_m$ are principal eigenvectors of $\mathbf{S}_T$, $\tilde{\mathbf{X}} = \mathbf{P}^{\mathrm{T}}\mathbf{X}$ is the PCA transform of the data matrix $\mathbf{X}$.

After constructing $\tilde{\mathbf{S}}_L$, we can determine $\tilde{\mathbf{S}}_N$ by

$$\begin{aligned}\tilde{\mathbf{S}}_N &= \mathbf{P}^{\mathrm{T}}\mathbf{S}_N\mathbf{P} = \mathbf{P}^{\mathrm{T}}(\mathbf{S}_T - \mathbf{S}_L)\mathbf{P} \\ &= \mathbf{P}^{\mathrm{T}}\mathbf{S}_T\mathbf{P} - \tilde{\mathbf{S}}_L = \mathrm{diag}(\mu_1, \cdots, \mu_m) - \tilde{\mathbf{S}}_L,\end{aligned} \quad (26)$$

where $\mu_1, \cdots, \mu_m$ are m largest nonzero eigenvalues of $\mathbf{S}_T$ corresponding to $\beta_1, \cdots, \beta_m$.

It should be noted that the above derivation is based on the whole range space of $\mathbf{S}_T$ (i.e., all nonzero eigenvectors of $\mathbf{S}_T$ are used to form this subspace). In practice, however, we always choose the number of principal eigenvectors, m, smaller than the real rank of $\mathbf{S}_T$ such that most of the spectrum energy is retained and $\tilde{\mathbf{S}}_L$ is well-conditioned (at least nonsingular) in the transformed space. In this case, the developed theory can be viewed as an approximate one and the generalized eigenvectors of $\tilde{\mathbf{S}}_N\mathbf{v} = \lambda\tilde{\mathbf{S}}_L\mathbf{v}$ can be calculated directly using the classical algorithm.

3.3 UDP Algorithm

In summary of the preceding description, the following provides the UDP algorithm:

Step 1. Construct the adjacency matrix: For the given training data set $\{\mathbf{x}_i | i = 1, \cdots, M\}$, find K nearest neighbors of each data point and construct the adjacency matrix $\mathbf{H} = (H_{ij})_{M\times M}$ using (14).

Step 2. Construct the local scatter kernel (LSK) matrix: Form an $M \times M$ diagonal matrix $\mathbf{D}$, whose elements on the diagonal are given by $D_{ii} = \sum_{j=1}^{M} H_{ij}$, $i = 1, \cdots, M$. Then, the LSK matrix is $\mathbf{L} = \mathbf{D} - \mathbf{H}$.

Step 3. Perform PCA transform of data: Calculate $\mathbf{S}_T$'s m largest positive eigenvalues $\mu_1, \cdots, \mu_m$ and the associated m orthonormal eigenvectors $\beta_1, \cdots, \beta_m$ using the technique presented in [2], [3]. Let $\tilde{\mathbf{S}}_T = \mathrm{diag}(\mu_1, \cdots, \mu_m)$ and $\mathbf{P} = (\beta_1, \cdots, \beta_m)$. Then, we get $\tilde{\mathbf{X}} = \mathbf{P}^T\mathbf{X}$, where $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M)$.

Step 4. Construct the two matrices $\tilde{\mathbf{S}}_L = \tilde{\mathbf{X}}\mathbf{L}\tilde{\mathbf{X}}^{\mathrm{T}}/M^2$ and $\tilde{\mathbf{S}}_N = \tilde{\mathbf{S}}_T - \tilde{\mathbf{S}}_L$. Calculate the generalized eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_d$ of $\tilde{\mathbf{S}}_N\mathbf{v} = \lambda\tilde{\mathbf{S}}_L\mathbf{v}$ corresponding to the d largest positive eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. Then, the d projection axes of UDP are $\mathbf{w}_j = \mathbf{P}\mathbf{v}_j$, $j = 1, \cdots, d$.

After obtaining the projection axes, we can form the following linear transform for a given sample $\mathbf{X}$:

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x}, \text{where } \mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d). \quad (27)$$

The feature vector $\mathbf{y}$ is used to represent the sample $\mathbf{x}$ for recognition purposes.

Concerning the UDP Algorithm, a remark should be made on the choice of m in Step 3. Liu and Wechsler [10] suggested a criterion for choosing the number of principal components in the PCA phase of their enhanced Fisher discriminant models. That is, a proper balance should be preserved between the *data energy* and the *eigenvalue magnitude* of the within-class scatter matrix [10]. This criterion can be borrowed here for the choice of m. First, to make $\tilde{\mathbf{S}}_L$ nonsingular, an m should be chosen that is less than the rank of LSK matrix $\mathbf{L}$. Second, to avoid overfitting, the trailing eigenvalues of $\tilde{\mathbf{S}}_L$ should not be too small.

4 Extension: UDP with Kernel Weighting

In this section, we will build a kernel-weighted version of UDP. We know that Laplacian Eigenmap [24] and LPP [31], [32] use kernel coefficients to weight the edges of the adjacency graph, where a heat kernel (Gaussian kernel) is defined by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\Big(-||\mathbf{x}_i - \mathbf{x}_j||^2/t\Big). \quad (28)$$

Obviously, for any $\mathbf{x}_i$, $\mathbf{x}_j$, and parameter t, $0 < k(\mathbf{x}_i, \mathbf{x}_j) \leq 1$ always holds. Further, the kernel function is a strictly monotone decreasing function with respect to the distance between two variables $\mathbf{x}_i$ and $\mathbf{x}_j$.

The purpose of the kernel weighting is to indicate the *degree* of $\mathbf{x}_i$ and $\mathbf{x}_j$ belonging to a local δ-neighborhood. If $||\mathbf{x}_i - \mathbf{x}_j||^2 < \delta$, the smaller the distance is, the larger the *degree* would be. Otherwise, the degree is zero. The kernel weighting, like other similar weightings, may be helpful in alleviating the effect of the outliers on the projection directions of the linear models and, thus, makes these models more robust to outliers [35].

4.1 Fundamentals

Let $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The kernel weighted global scatter can be characterized by

$$\begin{aligned}J_T(\mathbf{w}) &= \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}K_{ij}(y_i - y_j)^2 \\ &= \mathbf{w}^T\left[\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}K_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T\right]\mathbf{w}.\end{aligned} \quad (29)$$

Here, we can view that all training samples are within a δ-neighborhood (it is possible as long as δ is large enough). K_{ij} indicates the *degree* of $\mathbf{x}_i$ and $\mathbf{x}_j$ belonging to such a neighborhood.

Let us denote

$$\mathbf{S}_T = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}K_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T. \quad (30)$$

Similarly to the derivation of (13), we have

$$\mathbf{S}_T = \frac{1}{MM}(\mathbf{X}\mathbf{D}_T\mathbf{X}^{\mathrm{T}} - \mathbf{X}\mathbf{K}\mathbf{X}^{\mathrm{T}}) = \frac{1}{MM}\mathbf{X}\mathbf{L}_T\mathbf{X}^{\mathrm{T}}, \quad (31)$$

where $\mathbf{K} = (K_{ij})_{M\times M}$ and $\mathbf{D}_T$ is a diagonal matrix whose elements on the diagonal are the column (or row) sum of $\mathbf{K}$, i.e., $(\mathbf{D}_T)_{ii} = \sum_{j=1}^{M} K_{ij}$. $\mathbf{L}_T = \mathbf{D}_T - \mathbf{K}$ is called the global scatter kernel (GSK) matrix.

If the matrix $\mathbf{S}_T$ defined in (31) is used as the generation matrix and its principal eigenvectors are selected as projection axes, the *kernel weighted version of PCA* can be obtained.

If we redefine the adjacency matrix as

$$H_{ij} = \begin{cases} K_{ij}, & ||\mathbf{x}_i - \mathbf{x}_j||^2 < \delta \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

or

$$H_{ij} = \begin{cases} K_{ij}, & \text{if } \mathbf{x}_j \text{ is among K nearest nieghbors of } \mathbf{x}_i \\ & \quad \textit{and } \mathbf{x}_i \text{ is among K nearest nieghbors of } \mathbf{x}_j \\ 0 & \text{otherwise,} \end{cases} \tag{33}$$

the kernel-weighted local scatter can still be characterized by (10) or (11) and the kernel-weighted local scatter matrix can be expressed by (13).

The kernel-weighted nonlocal scatter is characterized by

$$\begin{aligned} J_N(\mathbf{w}) &= \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(K_{ij} - H_{ij})(y_i - y_j)^2 \\ &= \mathbf{w}^T\left[\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(K_{ij} - H_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T\right]\mathbf{w} \end{aligned} \tag{34}$$

and the corresponding nonlocal scatter matrix is

$$\mathbf{S}_N = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(K_{ij} - H_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T = \mathbf{S}_T - \mathbf{S}_L. \tag{35}$$

4.2 Algorithm of UDP with Kernel Weighting

The UDP algorithm in Section 3.3 can be modified to obtain its kernel weighted version. In Step 1, the adjacency matrix $\mathbf{H} = (H_{ij})_{M\times M}$ is constructed instead using (33) and, in Step 3, the PCA transform is replaced by its kernel-weighted version. For computational efficiency, the eigenvectors of $\mathbf{S}_T$ defined in (31) can be calculated in the following way.

Since $\mathbf{L}_T$ is a real symmetric matrix, its eigenvalues are all real. Calculate all of its eigenvalues and the corresponding eigenvectors. Suppose λ is the diagonal matrix of eigenvalues of $\mathbf{L}_T$ and $\mathbf{Q}$ is the full matrix whose columns are the corresponding eigenvectors, $\mathbf{L}_T$ can be decomposed by

$$\mathbf{L}_T = \mathbf{Q}\Lambda\mathbf{Q}^{\mathrm{T}} = \mathbf{Q}_L\mathbf{Q}_L^T, \text{where } \mathbf{Q}_L = \mathbf{Q}\Lambda^{\frac{1}{2}}. \tag{36}$$

From (36), it follows that $\mathbf{S}_T = \frac{1}{MM}(\mathbf{X}\mathbf{Q}_L)(\mathbf{X}\mathbf{Q}_L)^T$. Let us define $\mathbf{R} = \frac{1}{MM}(\mathbf{X}\mathbf{Q}_L)^T(\mathbf{X}\mathbf{Q}_L)$, which is an $M \times M$ nonnegative definite matrix. Calculate $\mathbf{R}$'s orthonormal eigenvectors $\alpha_1, \alpha_2, \cdots, \alpha_d$ which correspond to the d largest nonzero eigenvlaues $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_d$. Then, from the theorem of singular value decomposition (SVD) [36], the orthonormal eigenvectors $\beta_1, \beta_2, \cdots, \beta_d$ of $\mathbf{S}_T$ corresponding to the d largest nonzero eigenvlaues $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_d$ are

$$\beta_j = \frac{1}{M\sqrt{\mu_j}}\mathbf{X}\mathbf{Q}_L\alpha_j, j = 1, \cdots, d. \tag{37}$$

We should make the claim that the UDP algorithm given in Section 3.3 is a special case of its kernel weighted version since, when the kernel parameter $t = +\infty$, the weight $K_{ij} = 1$ for any i and j. For convenience, in this paper, we also refer to the kernel weighted UDP version simply as UDP and the UDP algorithm in Section 3.3 is denoted by UDP ($t = +\infty$).

5 Links to Other Linear Projection Techniques: LDA and LPP

5.1 Comparisons with LPP

UDP and LPP are both unsupervised subspace learning techniques. Their criteria, however, are quite different. UDP maximizes the ratio of the nonlocal scatter (or the global scatter) to the local scatter whereas LPP minimizes the local scatter.

The local scatter criterion can be minimized in different ways subject to different constraints. One way is to assume that the projection axes are mutually orthogonal (PCA is actually based on this constraint so as to maximize the global scatter criterion). This constraint-based optimization model is

$$\arg\min_{\mathbf{w}^{\mathrm{T}}\mathbf{w}=1} f(\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{S}_L\mathbf{w}. \tag{38}$$

By solving this optimization model, we get a set of orthogonal projection axes $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d$.

The other way to minimize the local scatter criterion is to assume that the projection axes are conjugately orthogonal. LPP is, in fact, based on these constraints. Let us define the matrix $\mathbf{S}_D = \mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}$. Then, the optimization model of LPP (based on the $\mathbf{S}_D$-orthogonality constraints) is given by

$$\arg\min_{\mathbf{w}^{\mathrm{T}}\mathbf{S}_D\mathbf{w}=1} f(\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{S}_L\mathbf{w}, \tag{39}$$

which is equivalent to

$$\arg\min J_P(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_L\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_D\mathbf{w}} \Leftrightarrow \arg\max J_P(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_D\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_L\mathbf{w}}. \tag{40}$$

Therefore, LPP essentially maximizes the ratio of $\mathbf{w}^{\mathrm{T}}\mathbf{S}_D\mathbf{w}$ to the local scatter. But, this criterion has no direct link to classification. Since the purpose of the constraint $\mathbf{w}^{\mathrm{T}}\mathbf{S}_D\mathbf{w} = \mathbf{w}^{\mathrm{T}}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\mathbf{w} = \mathbf{y}^{\mathrm{T}}\mathbf{S}_D\mathbf{y} = 1$ is just to remove an arbitrary scaling factor in the embedding and that of the matrix $\mathbf{D}$ is to provide a natural measure of the vertices of the adjacency graph [24], maximizing (or normalizing) $\mathbf{w}^{\mathrm{T}}\mathbf{S}_D\mathbf{w}$ does not make sense with respect to discrimination. In contrast, the criterion of UDP has a more transparent link to classification or clustering. Its physical meaning is very clear: If samples belong to the same cluster, they become closer after the projection; otherwise, they become as far apart as possible.

5.2 Connections to LDA

Compared with LPP, UDP has a more straightforward connection to LDA. Actually, LDA can be regarded as a special case of UDP if we assume that each class has the same

Fig. 2. Sample images of one person in the Yale database.

number of training samples (i.e., the class priori probabilities are same). When the data has an ideal clustering, i.e., each local neighborhood contains exactly the same number of training samples belonging to the same class, UDP is LDA. In this case, the adjacency matrix is

$$H_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same class} \\ 0 & \text{otherwise.} \end{cases} \tag{41}$$

And, in this case, there exist c local neighborhoods, each of which corresponds to a cluster of samples in one pattern class. Suppose that the kth neighborhood is formed by all l samples of Class k. Then, the local scatter matrix of the samples in the kth neighborhood is

$$\mathbf{S}_L^{(k)} = \frac{1}{2}\frac{1}{l^2}\sum_{i,j}\left(\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\right)\left(\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\right)^{\mathrm{T}}. \tag{42}$$

Following the derivation of $\mathbf{S}_T$ in (4), we have

$$\mathbf{S}_L^{(k)} = \frac{1}{l}\sum_{j=1}^{l}\left(\mathbf{x}_j^{(k)} - \mathbf{m}_k\right)\left(\mathbf{x}_j^{(k)} - \mathbf{m}_k\right)^{\mathrm{T}}. \tag{43}$$

So, $\mathbf{S}_L^{(k)} = \mathbf{S}_W^{(k)}$, where $\mathbf{S}_W^{(k)}$ is the covariance matrix of samples in Class k.

From the above derivation and (12), the whole local scatter matrix is

$$\begin{aligned} \mathbf{S}_L &= \frac{1}{2}\frac{1}{M^2}\sum_{k=1}^{c}\sum_{i,j}\left(\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\right)\left(\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\right)^{\mathrm{T}} \\ &= \frac{1}{2}\frac{1}{M^2}\sum_{k=1}^{c} 2l^2\mathbf{S}_L^{(k)} = \frac{l}{M}\sum_{k=1}^{c}\frac{l}{M}\mathbf{S}_W^{(k)} = \frac{l}{M}\mathbf{S}_W. \end{aligned} \tag{44}$$

Then, the nonlocal scatter matrix is

$$\mathbf{S}_N = \mathbf{S}_T - \mathbf{S}_L = \mathbf{S}_T - \frac{l}{M}\mathbf{S}_W. \tag{45}$$

Further, it can be shown that the following equivalent relationships hold:

$$\begin{aligned} J(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_N\mathbf{w}}{\mathbf{w}^T\mathbf{S}_L\mathbf{w}} &\Leftrightarrow \frac{\mathbf{w}^T(\mathbf{S}_T - \frac{l}{M}\mathbf{S}_W)\mathbf{w}}{\mathbf{w}^T(\frac{l}{M}\mathbf{S}_W)\mathbf{w}} \Leftrightarrow \frac{\mathbf{w}^T\mathbf{S}_T\mathbf{w}}{\mathbf{w}^T(\frac{l}{M}\mathbf{S}_W)\mathbf{w}} \\ &\Leftrightarrow \frac{\mathbf{w}^T\mathbf{S}_T\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}} \Leftrightarrow \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}} = J_F(\mathbf{w}). \end{aligned} \tag{46}$$

Therefore, UDP is LDA in the case where each local neighborhood contains exactly the same number of training samples belonging to the same class.

The connection between LPP and LDA was disclosed in [32] provided that a (41)-like adjacency relationship is given. In addition, LPP needs another assumption, i.e., the sample mean of the data set is zero, to connect itself to LDA, while UDP does not. So, the connection between UDP and LDA is more straightforward.

6 Biometrics Applications: Experiments and Analysis

In this section, the performance of UDP is evaluated on the Yale, FERET, and AR face image databases and PolyU Palmprint database and compared with the performances of PCA, LDA, and Laplacianface (LPP).

6.1 Experiment Using the Yale Database

The Yale face database contains 165 images of 15 individuals (each person providing 11 different images) under various facial expressions and lighting conditions. In our experiments, each image was manually cropped and resized to 100×80 pixels. Fig. 2 shows sample images of one person.

The first experiment was performed using the first six images (i.e., center-light, with glasses, happy, left-light, without glasses, and normal) per class for training, and the remaining five images (i.e., right-light, sad, sleepy, surprised, and winking) for testing. For feature extraction, we used, respectively, PCA (eigenface), LDA (Fisherface), LPP (Laplacianface), and the proposed UDP. Note that Fisherface, Laplacianface, and UDP all involve a PCA phase. In this phase, we keep nearly 98 percent image energy and select the number of principal components, m, as 60 for each method. The K-nearest neighborhood parameter K in UDP and Laplacianface can be chosen as $\mathrm{K} = l - 1 = 5$, where l denotes the number of training samples per class. The justification for this choice is that each sample should connect with the remaining $l-1$ samples of the same class provided that within-class samples are well clustered in the observation

TABLE 1
The Maximal Recognition Rates (Percent) of PCA, LDA, Laplacianface, and UDP on the Yale Database and the Corresponding Dimensions (Shown in Parentheses) When the First Six Samples Per Class Are Used for Training

Measure	PCA	LDA	Laplacianface ($t=+\infty$)	UDP ($t=+\infty$)	Laplacianface ($t^*=800$)	UDP ($t^*=800$)
Euclidean	90.7 (28)	86.7 (14)	90.7 (30)	96.0 (18)	90.7 (24)	97.3 (18)
Cosine	90.7 (40)	96.0 (14)	97.3 (30)	98.7 (18)	98.7 (24)	100 (18)

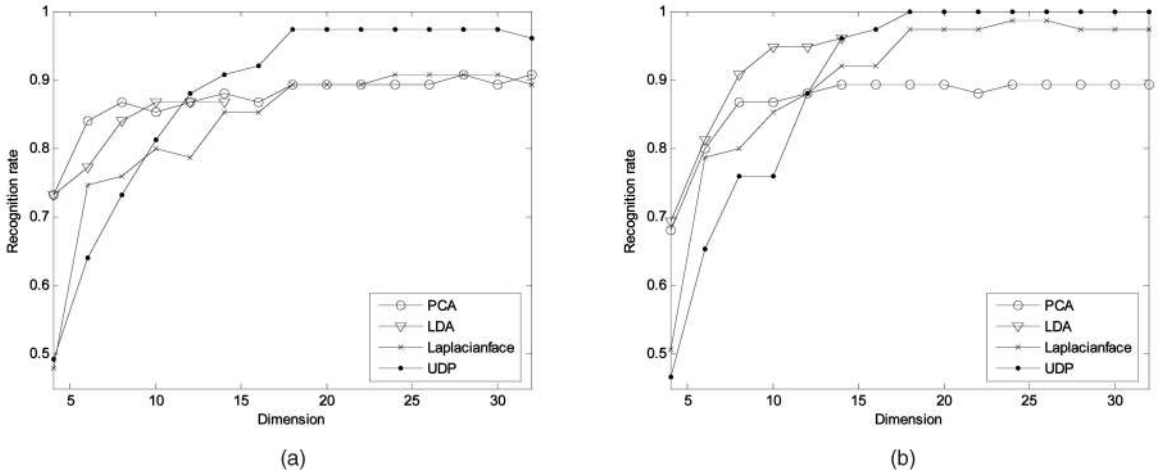


Fig. 3. The recognition rates of PCA, LDA, Laplacianface, and UDP versus the dimensions when (a) Euclidean distance is used and (b) cosine distance is used.

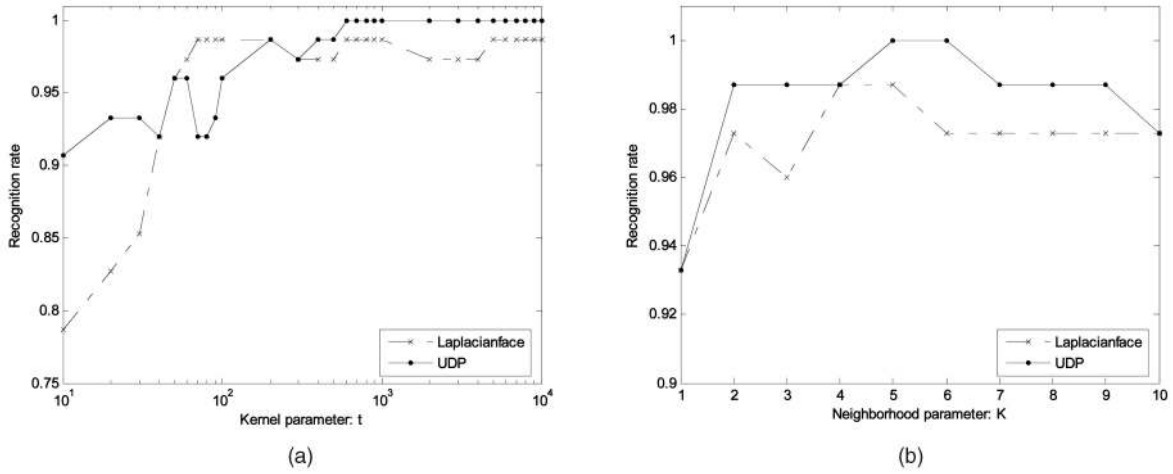


Fig. 4. (a) The maximal recognition rates of Laplacianface and UDP versus the variation of kernel parameter t. (b) The maximal recognition rates of Laplacianface and UDP versus the variation of K-nearest neighborhood parameter K.

space. There are generally two ways to select the Gaussian kernel parameter t. One way is to choose $t=+\infty$, which represents a special case of LPP (or UDP). The other way is to determine a proper parameter t^* within the interval $(0,+\infty)$ using the global-to-local strategy [20] to make the recognition result optimal. Finally, the nearest neighbor (NN) classifiers with Euclidean distance and cosine distance are, respectively, employed for classification. The maximal recognition rate of each method and the corresponding dimension are given in Table 1. The recognition rates versus the variation of dimensions are illustrated in Fig. 3. The recognition rates of Laplacianface and UDP versus the variation of the kernel parameter t and those versus the K-nearest neighborhood parameter K are, respectively, illustrated in Figs. 4a and 4b.

From Table 1, we can see three main points. First, UDP outperforms Laplacianface under each distance measure, whether the kernel parameter t is infinity or optimally chosen (t^* = 800). Second, UDP and Laplacianface with cosine distances both perform better than LDA and PCA with cosine or Euclidean distances. Third, the cosine distance metric can significantly improve the performance of LDA, Laplacianface, and UDP, but it has no substantial effect on the performance of PCA. Fig. 3 shows that UDP (t^* = 800)

TABLE 2
The Maximal Average Recognition Rates (Percent) of PCA, LDA, Laplacianface, and UDP across 20 Runs on the Yale Database and the Corresponding Standard Deviations (Std) and Dimensions (Shown in Parentheses)

Measure	PCA	LDA	Laplacianface ($t = +\infty$)	UDP ($t = +\infty$)	Laplacianface ($t^* = 800$)	UDP ($t^* = 800$)
Euclidean	91.7 ± 5.4 (28)	87.1 ± 9.8 (14)	89.2 ± 4.7 (44)	91.9 ± 5.1 (28)	90.3 ± 5.1 (24)	92.3 ± 5.9 (28)
Cosine	90.1 ± 6.9 (24)	92.1 ± 6.7 (14)	94.2 ± 3.3 (48)	95.1 ± 4.3 (28)	95.0 ± 2.9 (24)	95.5 ± 4.1 (28)

Fig. 5. Samples of the cropped images in a subset of the FERET database.

outperforms Laplacianface ($t^* = 800$), LDA and PCA when the dimension is over 16, no matter what distance metric is used. Further, the recognition rate of UDP with cosine distance retains 100 percent as the dimension varies from 18 to 32. Fig. 4a indicates that the performances of UDP and Laplacianface (with cosine distance) become robust when the parameter t is over 200 and UDP consistently outperforms Laplacianface when t is larger than 400. The recognition rate of UDP retains 100 percent as t varies from 600 to 10,000. Fig. 4b shows that the performances of UDP and Laplacianface vary with the variation of the K-nearest neighborhood parameter K. When K is chosen as $l - 1 = 5$, both methods achieve their top recognition rates. So, we will choose $\mathrm{K} = l - 1$ for our experiments.

Why can the unsupervised method UDP (or Laplacianface) outperform the supervised method LDA? In our opinion, the possible reason is that UDP (or Laplacianface) is more robust than LDA to outliers. In the training set of this experiment, the "left-light" image of each class can be viewed as an outlier. The outlier images may cause errors in the estimate of within-class scatter and, thus, make LDA projection inaccurate. In contrast, UDP builds the adjacency relationship of data points using k-nearest neighbors and groups the data in a natural way. Most outlier images of different persons are grouped into new different clusters. By this means, the number of clusters increases, but the negative influence of outliers onto within-class scatter is eliminated. So, the resulting projection of UDP is more accurate and discriminative. Since the number of clusters increases, UDP generally needs more features than LDA to achieve its best performance. This also gives the reason why LDA can outperform UDP using a few features, as shown in Fig. 3.

In the second experiment, 20-fold cross-validation tests are performed to reevaluate the performance of PCA, LDA, Laplacianface, and UDP. In each test, six images of each subject are randomly chosen for training, while the remaining five images are used for testing. The parameters involved in each method are set as the same as those used in the first experiment. Table 2 shows the maximal average recognition rates across 20 runs of each method under nearest neighbor classifiers with two distance metrics and their corresponding standard deviations (*std*) and dimensions. From Table 2, it can be seen that UDP outperforms other methods and the cosine distance metric is still helpful in improving the performance of LDA, Laplacianface, and UDP. These conclusions are, on the whole, consistent with those drawn from the first experiment.

Since the cosine distance is more effective than the Euclidean distance for LDA, Laplacianface, and UDP, in the following experiments we use only this distance metric.

6.2 Experiment Using the FERET Database

The FERET face image database has become a standard database for testing and evaluating state-of-the-art face recognition algorithms [37], [38], [39]. The proposed method was tested on a subset of the FERET database. This subset includes 1,000 images of 200 individuals (each one has five images). It is composed of the images whose names are marked with two-character strings: "*ba*," "*bj*," "*bk*," "*be*," "*bf*." This subset involves variations in facial expression, illumination, and pose. In our experiment, the facial portion of each original image was automatically cropped based on the location of eyes and mouth, and the cropped image was resized to 80×80 pixels and further preprocessed by histogram equalization. Some sample images of one person are shown in Fig. 5.

In our test, we use the first two images (i.e., "*ba*" and "*bj*") per class for training and the remaining three images (i.e., "*bk*," "*be*," and "*bf*") for testing. PCA, LDA, Laplacianface, and UDP are used for feature extraction. In the PCA phase of LDA, Laplacianface, and UDP, the number of principal components, m, is set as 120. The K-nearest neighborhood parameter K in Laplacianface and UDP is chosen as $\mathrm{K} = l - 1 = 1$. After feature extraction, a nearest neighbor classifier with cosine distance is employed for classification. The maximal recognition rate of each method and the corresponding dimension are given in Table 3. The recognition rate curve versus the variation of dimensions is shown in Fig. 6.

TABLE 3
The Maximal Recognition Rates (Percent) of PCA, LDA, Laplacianface, and UDP on a Subset of the FERET Database and the Corresponding Dimensions

Method	PCA	LDA	Laplacianface (t = $+\infty$)	UDP (t = $+\infty$)	Laplacianface (t* = 300)	UDP (t* =7000)
Recognition rate	73.3	75.0	77.0	80.7	78.5	81.2
Dimension	85	100	105	90	90	110

Table 3 demonstrates again that UDP outperforms PCA, LDA, and Laplacianface, whether the kernel parameter t is infinity or optimally chosen (t* = 7,000 for UDP and t* = 300 for Laplacianface). Fig. 6 indicates that UDP consistently performs better than other methods when the dimension is over 45.

6.3 Experiment Using the AR Database

The AR face [40], [41] contains over 4,000 color face images of 126 people (70 men and 56 women), including frontal views of faces with different facial expressions, lighting conditions, and occlusions. The pictures of 120 individuals (65 men and 55 women) were taken in two sessions (separated by two weeks) and each section contains 13 color images. Twenty face images (each session containing 10) of these 120 individuals are selected and used in our experiment. The face portion of each image is manually cropped and then normalized to 50×40 pixels. The sample images of one person are shown in Fig. 7. These images vary as follows:

1. neutral expression,
2. smiling,
3. angry,
4. screaming,
5. left light on,
6. right light on,
7. all sides light on,
8. wearing sun glasses,
9. wearing sun glasses and left light on, and
10. wearing sun glasses and right light on.

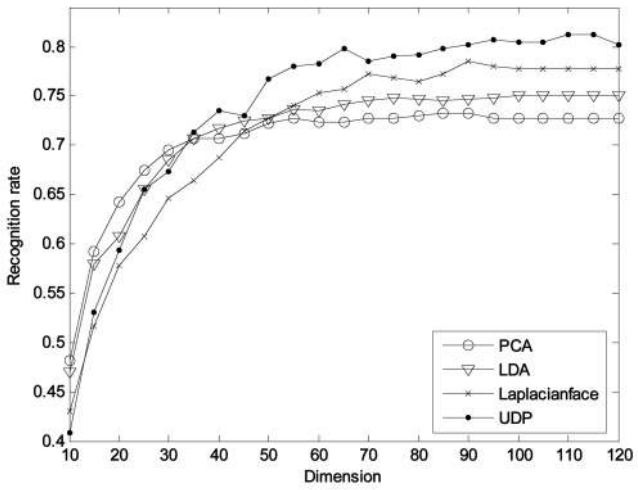


Fig. 6. The recognition rates of PCA, LDA, Laplacianface, and UDP versus the dimensions when cosine distance is used on a subset of FERET database.

In our experiments, l images (l varies from 2 to 6) are randomly selected from the image gallery of each individual to form the training sample set. The remaining $20 - 1$ images are used for testing. For each l, we perform cross-validation tests and run the system 20 times. PCA, LDA, Laplacianface, and UDP are, respectively, used for face representation. In the PCA phase of LDA, Laplacianface, and UDP, the number of principal components, m, is set as 50, 120, 180, 240, and 300, respectively, corresponding to $l = 2$, 3, 4, 5, and 6. The K-nearest neighborhood parameter K in Laplacianface and UDP is chosen as $\mathrm{K} = l - 1$. Finally, a nearest-neighbor classifier with cosine distance is employed for classification. The maximal average recognition rate and the *std* across 20 runs of tests of each method are shown in Table 4. The recognition rate curve versus the variation of training sample sizes is shown in Fig. 8.

From Table 4 and Fig. 8, we can see first that UDP overall outperforms Laplacianface, whether the kernel parameter is infinity or optimally chosen and second that as unsupervised methods, UDP and Laplacianface both significantly outperform PCA, irrespective of the variation in training sample size. These two points are consistent with the experimental results in Sections 6.1 and 6.2. In addition, we can see some inconsistent results. First, with reference to the impact of the kernel weighting on the performance of UDP and Laplacianface, in this experiment, UDP and Laplacianface both perform well without kernel weighting (i.e., $t = +\infty$). The heat-kernel (i.e., Gaussian kernel) weighting by optimally choosing t* = 300 for Laplacianface and t* = 500 for UDP from the interval $(0, +\infty)$, however, does little to improve the recognition accuracy.

Another inconsistent point that is worth remarking upon concerns the performance comparison of UDP and LDA. UDP outperforms LDA when l is less than 5, while LDA outperforms UDP when l is over 5. This means that, once the given training sample size per class becomes large, LDA may achieve better results than UDP. It is not hard to interpret this phenomenon from a statistical point of view. While there are more and more samples per class provided for training, the within-class scatter matrix can be evaluated more accurately and becomes better-conditioned, so LDA will become more robust. However, with the increase of the training sample size, more boundary points might exist between arbitrary two data clusters in input space. This makes it more difficult for UDP (or LPP) to choose a proper locality radius or the K-nearest neighborhood parameter K to characterize the "locality."

Nevertheless, UDP does have an advantage over LDA with respect to a specific biometrics problem like face recognition. Fig. 8 indicates that, the smaller the training sample size is, the more significant the performance difference between UDP and LDA becomes. This advantage of UDP in small sample size cases is really helpful in practice.

Fig. 7. Samples of the cropped images of one person in the AR database.

TABLE 4
The Maximal Average Recognition Rates (Percent) and Standard Deviations (Std) of PCA, LDA, Laplacianface, and UDP with Different Training Sample Sizes on the AR Database

# / class	PCA	LDA	Laplacianface (t = +∞)	UDP (t = +∞)	Laplacianface (t* = 300)	UDP (t* = 500)
2	71.2 ± 6.0	70.7 ± 11.5	75.5 ± 8.1	76.7 ± 7.7	75.6 ± 8.1	76.6 ± 7.9
3	74.4 ± 5.7	82.1 ± 13.5	85.1 ± 7.4	86.9 ± 8.0	85.2 ± 7.5	86.8 ± 7.9
4	80.2 ± 6.0	91.2 ± 11.4	91.7 ± 4.5	93.3 ± 4.7	91.7 ± 4.5	93.2 ± 4.9
5	81.4 ± 6.2	93.9 ± 8.0	92.6 ± 4.9	93.9 ± 5.1	92.5 ± 5.0	93.9 ± 5.0
6	84.5 ± 4.3	96.7 ± 2.4	94.2 ± 2.7	95.5 ± 2.0	94.1 ± 2.9	95.6 ± 2.0

This is because face recognition is typically a small sample size problem. There are generally a few images of one person provided for training in many real-world applications.

6.4 Experiment Using the PolyU Palmprint Database

The PolyU palmprint database contains 600 gray-scale images of 100 different palms with six samples for each palm (http://www4.comp.polyu.edu.hk/~biometrics/). Six samples from each of these palms were collected in two sessions, where the first three were captured in the first session and the other three in the second session. The average interval between the first and the second sessions is two months. In our experiments, the central part of each original image was automatically cropped using the algorithm mentioned in [42]. The cropped images were resized to 128×128 pixels and preprocessed using histogram equalization. Fig. 9 shows some sample images of two palms.

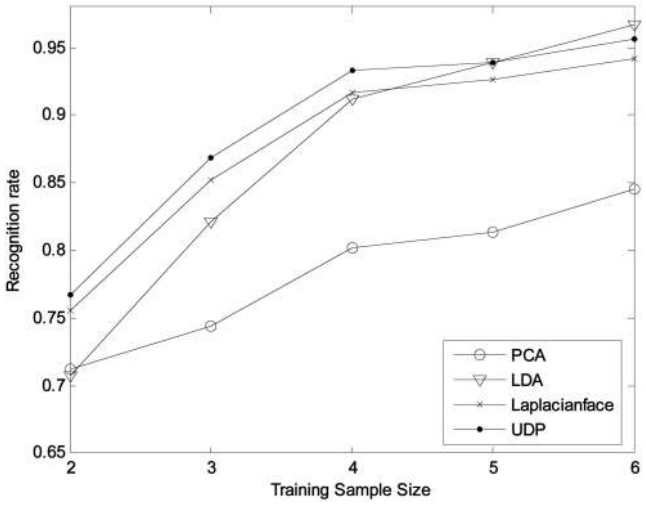


Fig. 8. The maximal average recognition rates of PCA, LDA, Laplacianface, and UDP versus the variation of the training sample size.

According to the protocol of this database, the images captured in the first session are used for training and the images captured in the second session for testing. Thus, for each palm class, there are three training samples and three testing samples. PCA, LDA, Laplacianface, and UDP are used for palm feature extraction. In the PCA phase of LDA, Laplacianface, and UDP, the number of principal components, m, is set as 150. The K-nearest neighborhood parameter K in Laplacianface and UDP is chosen as $K = l - 1 = 2$. After feature extraction, a nearest neighbor classifier with cosine distance is employed for classification. The maximal recognition rate of each method and the corresponding dimension are listed in Table 5. The recognition rate curve versus the variation of dimensions is shown in Fig. 10.

From Table 3, we can see that UDP outperforms PCA, LDA, and Laplacianface. The recognition rate of UDP (when t* = 200) is up to 99.7 percent, i.e., only one sample was missed. Fig. 6 shows that UDP consistently performs better than other methods, irrespective of the dimensional variation. These results demonstrate that UDP is also a good tool for palm recognition.

7 Conclusions and Future Work

In this paper, we develop an unsupervised discriminant projection (UDP) technique for dimensionality reduction of high-dimensional data in small sample size cases. The projection of UDP can be viewed as a linear approximation of the nonlinear map that uncovers and separates embeddings corresponding to different manifolds in the final embedding space. UDP considers the local and nonlocal scatters at the same time and seeks to find a projection maximizing the ratio of the nonlocal scatter to the local scatter. The consideration of the nonlocal quantity makes UDP more intuitive and more powerful than LPP for classification or clustering tasks. Our experimental results on three popular face image databases and one palmprint database demonstrate that UDP is more effective than LPP

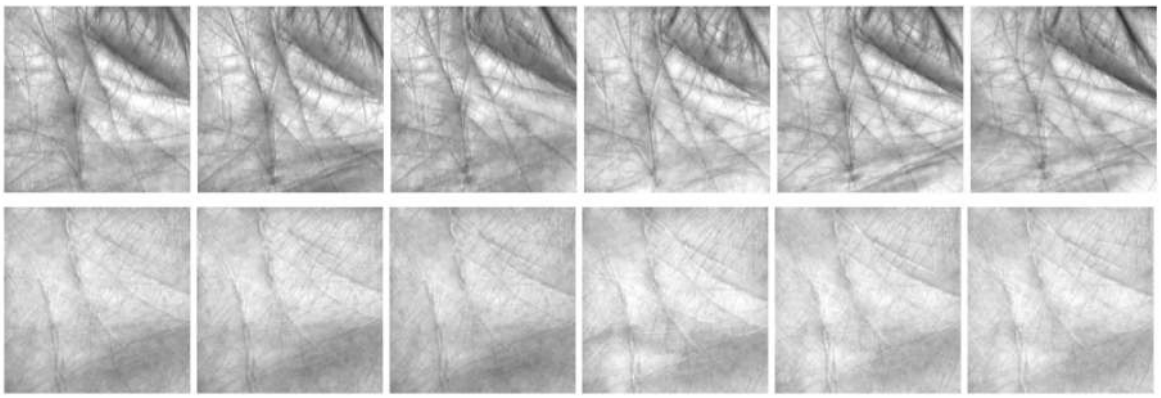


Fig. 9. Samples of the cropped images in the PolyU Palmprint database.

TABLE 5
The Maximal Recognition Rates (Percent) of PCA, LDA, Laplacianface, and UDP on the PolyU Palmprint Database and the Corresponding Dimensions

Method	PCA	LDA	Laplacianface ($t = +\infty$)	UDP ($t = +\infty$)	Laplacianface ($t^* = 300$)	UDP ($t^* = 200$)
Recognition rate	86.0	95.7	98.3	99.3	99.0	99.7
Dimension	90	90	140	90	100	100

and PCA. In addition, UDP is more discriminative than LDA when the training sample size per class is small.

Our experimental results on the AR database, however, also reveal a drawback of UDP (LPP actually has the same problem). That is, as the training sample size per class becomes large, LDA can outperform UDP. This problem is unnoticeable in most real-world biometrics applications since the given training sample size is always very small. But, it may become prominent once UDP is applied to large sample size problems. To address this, we need a more precise characterization of the local scatter and the nonlocal scatter when the given training sample size per class is relatively large. A possible way is to use the provided class label information (for example, borrowing Yan et al.'s [33] and Chen's [34] ideas) to facilitate this characterization and then to build a semisupervised hybrid system.

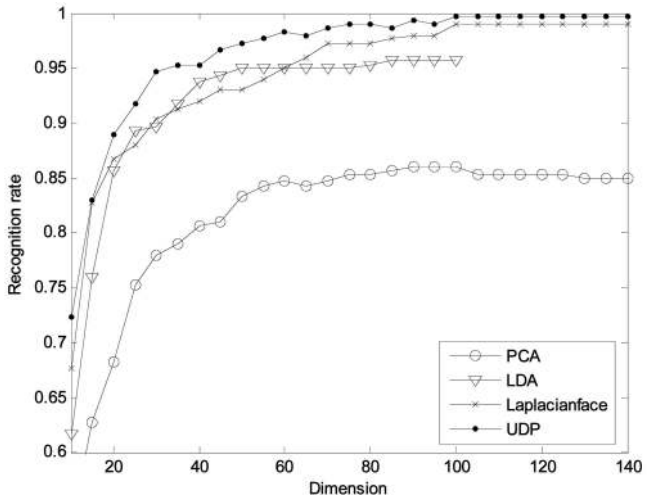


Fig. 10. The recognition rates of PCA, LDA, Laplacianface, and UDP versus the dimensions when cosine distance is used on the PolyU Palmprint database.

As a generator of weighting coefficients, the Gaussian kernel (or heat kernel) is examined in this paper. It is demonstrated to be effective in most cases. But, in some cases, it fails to improve the performance of UDP or LPP. Are there more effective kernels for weighting the proposed method? This is a problem deserving further investigation. In addition, in this paper, we focus on developing a linear projection technique and applying it to biometrics but do not address another interesting problem, i.e., modeling multimanifolds for classification purposes. When different classes of data lie on different manifolds, it is of central importance to uncover the embeddings corresponding to different manifolds and, at the same time, to make different embeddings as separable as possible in the final embedding space. We will address this problem and try to build a general framework for classification-oriented multimanifolds learning in the near future. This framework may result in more effective features for biometrics tasks.

ACKNOWLEDGMENTS

This work is partially supported by the UGC/CRC fund from the HKSAR Government, the central fund from the Hong Kong Polytechnic University, and the National Science Foundation of China under Grants No. 60332010, No. 60503026, No. 60472060, No. 60473039, and No. 60632050. Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office. The authors would like to thank all the guest editors and anonymous reviewers for their constructive advices.

REFERENCES

[1] A.K. Jain, R.P.W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-37, Jan. 2000.

[2] M. Kirby and L. Sirovich, "Application of the KL Procedure for the Characterization of Human Faces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 1, pp. 103-108, Jan. 1990.

[3] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.

[4] K. Liu, Y.-Q. Cheng, J.-Y. Yang, and X. Liu, "An Efficient Algorithm for Foley-Sammon Optimal Set of Discriminant Vectors by Algebraic Method," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 6, no. 5, pp. 817-829, 1992.

[5] D.L. Swets and J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 831-836, Aug. 1996.

[6] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriengman, "Eigenfaces versus Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 711-720, July 1997.

[7] L.F. Chen, H.Y. M. Liao, J.C. Lin, M.D. Kao, and G.J. Yu, "A New LDA-Based Face Recognition System which Can Solve the Small Sample Size Problem," *Pattern Recognition,* vol. 33, no. 10, pp. 1713-1726, 2000.

[8] H. Yu and J. Yang, "A Direct LDA Algorithm for High-Dimensional Data—With Application to Face Recognition," *Pattern Recognition,* vol. 34, no. 10, pp. 2067-2070, 2001.

[9] J. Yang and J.Y. Yang, "Why Can LDA Be Performed in PCA Transformed Space?" *Pattern Recognition,* vol. 36, no. 2, pp. 563-566, 2003.

[10] C.J. Liu and H. Wechsler, "A Shape- and Texture-Based Enhanced Fisher Classifier for Face Recognition," *IEEE Trans. Image Processing,* vol. 10, no. 4, pp. 598-608, 2001.

[11] W. Zhao, A. Krishnaswamy, R. Chellappa, D. Swets, and J. Weng, "Discriminant Analysis of Principal Components for Face Recognition," *Face Recognition: From Theory to Applications,* H. Wechsler, P.J. Phillips, V. Bruce, F.F. Soulie, and T.S. Huang, eds., pp. 73-85, Springer-Verlag, 1998.

[12] J. Yang, D. Zhang, A.F. Frangi, and J.-y. Yang, "Two-Dimensional PCA: A New Approach to Face Representation and Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 1, pp. 131-137, Jan. 2004.

[13] J. Yang, D. Zhang, X. Yong, and J.-y Yang, "Two-Dimensional Discriminant Transform for Face Recognition," *Pattern Recognition,* vol. 38, no. 7, pp. 1125-1129, 2005.

[14] J. Ye and 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, June 2005.

[15] B. Schölkopf, A. Smola, and K.R. Muller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation,* vol. 10, no. 5, pp. 1299-1319, 1998.

[16] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K.-R. Müller, "Constructing Descriptive and Discriminative Nonlinear Features: Rayleigh Coefficients in Kernel Feature Spaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, pp. 623-628, May 2003.

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

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

[19] 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.

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

[21] H.S. Seung and D.D. Lee, "The Manifold Ways of Perception," *Science,* vol. 290. pp. 2268-2269, 2000.

[22] J.B. Tenenbaum, V. deSilva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, pp. 2319-2323, 2000.

[23] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, pp. 2323-2326, 2000.

[24] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation,* vol. 15, no. 6, pp. 1373-1396, 2003.

[25] O. Kouropteva, O. Okun, and M. Pietikainen, "Supervised Locally Linear Embedding Algorithm for Pattern Recognition," *Lecture Notes in Computer Science,* vol. 2652, pp. 386-394, 2003.

[26] D. Ridder, M. Loog, and M. Reinders, "Local Fisher Embedding," *Proc. 17th Int'l Conf. Pattern Recognition,* 2004.

[27] N. Vlassis, Y. Motomura, and B. Krose, "Supervised Dimension Reduction of Intrinsically Lowdimensional Data," *Neural Computation,* vol. 14, no. 1, pp. 191-215, 2002.

[28] L.K. Saul and S.T. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds," *J. Machine Learning Research,* vol. 4, pp. 119-155, 2003.

[29] M. Brand, "Charting a Manifold," *Proc. 15th Conf. Neural Information Processing Systems,* 2002.

[30] Y. Bengio, J-F. Paiement, and P. Vincent, "Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering," *Proc. 16th Conf. Neural Information Processing Systems,* 2003.

[31] X. He and P. Niyogi, "Locality Preserving Projections," *Proc. 16th Conf. Neural Information Processing Systems,* 2003.

[32] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face Recognition Using Laplacianfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 3, pp. 328-340, Mar. 2005.

[33] S. Yan, D. Xu, B. Zhang, and H.-J. Zhang, "Graph Embedding: A General Framework for Dimensionality Reduction," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 830-837, 2005.

[34] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local Discriminant Embedding and Its Variants," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 846-853, 2005.

[35] Y. Koren and L. Carmel, "Robust Linear Dimensionality Reduction," *IEEE Trans. Visualization and Computer Graphics,* vol. 10, no. 4, pp. 459-470, July/Aug. 2004.

[36] G.H. Golub and C.F. VanLoan, *Matrix Computations,* third ed. Johns Hopkins Univ. Press, 1996.

[37] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 10, pp. 1090-1104, Oct. 2000.

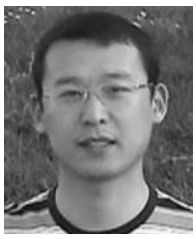
[38] P.J. Phillips, "The Facial Recognition Technology (FERET) Database," http://www.itl.nist.gov/iad/humanid/feret/feret_master.html, 2006.

[39] P.J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET Database and Evaluation Procedure for Face Recognition Algorithms," *Image and Vision Computing,* vol. 16, no. 5, pp. 295-306, 1998.

[40] A.M. Martinez and R. Benavente, "The AR Face Database," http://rvl1.ecn.purdue.edu/aleix/~aleix_face_DB.html, 2006.

[41] A.M. Martinez and R. Benavente, "The AR Face Database," CVC Technical Report #24, June 1998.

[42] D. Zhang, *Palmprint Authentication.* Kluwer Academic, 2004.



Jian Yang received the BS degree in mathematics from the Xuzhou Normal University in 1995. He received the MS degree in applied mathematics from the Changsha Railway University in 1998 and the PhD degree from the Nanjing University of Science and Technology (NUST) Department of Computer Science on the subject of pattern recognition and intelligence systems in 2002. From January to December 2003, he was a postdoctoral researcher at the University of Zaragoza and affiliated with the Division of Bioengineering of the Aragon Institute of Engineering Research (I3A). In the same year, he was awarded the RyC program Research Fellowship, sponsored by the Spanish Ministry of Science and Technology. Now, he is a professor in the Department of Computer Science of NUST and, at the same time, a postdoctoral research fellow at Biometrics Centre of Hong Kong Polytechnic University. He is the author of more than 40 scientific papers in pattern recognition and computer vision. His current research interests include pattern recognition, computer vision, and machine learning.

David Zhang received the BS degree in computer science from Peking University, the MSc degree in computer science in 1982, and the PhD degree in 1985 from the Harbin Institute of Technology (HIT). From 1986 to 1988, he was a postdoctoral fellow at Tsinghua University and then an associate professor at the Academia Sinica, Beijing. In 1994, he received a second PhD degree in electrical and computer engineering from the University of Waterloo, Ontario, Canada. Currently, he is a chair professor at the Hong Kong Polytechnic University, where he is the founding director of the Biometrics Technology Centre (UGC/CRC) supported by the Hong Kong SAR Government. He also serves as a adjunct professor at Tsinghua University, Shanghai Jiao Tong University, Beihang University, Harbin Institute of Technology, and the University of Waterloo. He is the founder and editor-in-chief of the *International Journal of Image and Graphics* (IJIG), book editor of the Springer International Series on Biometrics (KISB), organizer of the International Conference on Biometrics Authentication (ICBA), associate editor of more than 10 international journals including the *IEEE Transactions on SMC-A/SMC-C/Pattern Recognition*, technical committee chair of IEEE CIS and the author of more than 10 books and 160 journal papers. Professor Zhang is a Croucher Senior Research Fellow, a Distinguished Speaker of the IEEE Computer Society, and a fellow of the International Association of Pattern Recognition (IAPR). He is a senior member of the IEEE.

Jing-yu Yang received the BS degree in computer science from Nanjing University of Science and Technology (NUST), Nanjing, China. From 1982 to 1984, he was a visiting scientist at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. From 1993 to 1994, he was a visiting professor at the Department of Computer Science, Missuria University. In 1998, he acted as a visiting professor at Concordia University in Canada. He is currently a professor and chairman in the Department of Computer Science at NUST. He is the author of more than 300 scientific papers in computer vision, pattern recognition, and artificial intelligence. He has won more than 20 provincial awards and national awards. His current research interests are in the areas of pattern recognition, robot vision, image processing, data fusion, and artificial intelligence.

Ben Niu received the BSc degree in 1999 and the MSc degree in 2002, both in applied mathematics, from the Hebei University, People's Republic of China. He is now a research assistant in the Department of Computing, Hong Kong Polytechnic University. His current research interest is in data mining, machine learning, and case-based reasoning.