Locality Sensitive Discriminant Analysis*

Deng Cai
Department of Computer Science
University of Illinois at Urbana Champaign
dengcai2@cs.uiuc.edu

Xiaofei He
Yahoo! Research Labs
hex@yahoo-inc.com

Kun Zhou
Microsoft Research Asia
kunzhou@microsoft.com

Jiawei Han
Department of Computer Science
University of Illinois at Urbana Champaign
hanj@cs.uiuc.edu

Hujun Bao
College of Computer Science
Zhejiang University
bao@cad.zju.edu.cn

Abstract

Linear Discriminant Analysis (LDA) is a popular data-analytic tool for studying the class relationship between data points. A major disadvantage of LDA is that it fails to discover the local geometrical structure of the data manifold. In this paper, we introduce a novel linear algorithm for discriminant analysis, called **Locality Sensitive Discriminant Analysis** (LSDA). When there is no sufficient training samples, local structure is generally more important than global structure for discriminant analysis. By discovering the local manifold structure, LSDA finds a projection which maximizes the margin between data points from different classes at each local area. Specifically, the data points are mapped into a subspace in which the nearby points with the same label are close to each other while the nearby points with different labels are far apart. Experiments carried out on several standard face databases show a clear improvement over the results of LDA-based recognition.

1 Introduction

Practical algorithms in supervised machine learning degrade in performance (prediction accuracy) when faced with many features that are not necessary for predicting the desired output. An important question in the fields of machine learning, knowledge discovery, computer vision and pattern recognition is how to extract a small number of good features. A common way to attempt to resolve this problem is to use dimensionality reduction techniques. Two of the most popular techniques for this purpose are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [Duda *et al.*, 2000].

PCA is an unsupervised method. It aims to project the data along the direction of maximal variance. LDA is supervised. It searches for the project 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. Both of them are spectral methods, i.e., methods based on eigenvalue decomposition of either the covariance matrix for PCA or the scatter matrices (within-class scatter matrix and between-class scatter matrix) for LDA. Intrinsically, these methods try to estimate the global statistics, i.e. mean and covariance. They may fail when there is no sufficient number of samples. Moreover, both PCA and LDA effectively see only the Euclidean structure. They fail to discover the underlying structure, if the data lives on or close to a submanifold of the ambient space.

Recently there has been a lot of interest in geometrically motivated approaches to data analysis in high dimensional spaces. Examples include ISOAMP [Tenenbaum *et al.*, 2000], Laplacian Eigenmap [Belkin and Niyogi, 2001], Locally Linear Embedding [Roweis and Saul, 2000]. These methods have been shown to be effective in discovering the geometrical structure of the underlying manifold. However, they are unsupervised in nature and fail to discover the discriminant structure in the data. In the meantime, manifold based semi-supervised learning has attracted considerable attention [Zhou *et al.*, 2003], [Belkin *et al.*, 2004]. These methods make use of both labeled and unlabeled samples. The labeled samples are used to discover the discriminant structure, while the unlabeled samples are used to discover the geometrical structure. When there is a large amount of unlabeled samples available, these methods may outperform traditional supervised learning algorithms such as Support Vector Machines and regression [Belkin *et al.*, 2004]. However, in some applications such as face recognition, the unlabeled samples may not be available, thus these semi-supervised learning methods can not be applied.

In this paper, we introduce a novel supervised dimension-

* The work was supported in part by the U.S. National Science Foundation NSF IIS-03-08215/IIS-05-13678, Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20030335083) and National Natural Science Foundation of China (No. 60633070). Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

ality reduction algorithm, called **Locality Sensitive Discriminant Analysis**, that exploits the geometry of the data manifold. We first construct a nearest neighbor graph to model the local geometrical structure of the underlying manifold. This graph is then split into *within-class graph* and *between-class graph* by using the class labels. In this way, the geometrical and discriminant structure of the data manifold can be accurately characterized by these two graphs. Using the notion of graph Laplacian [Chung, 1997], we can find a linear transformation matrix which maps the data points to a subspace. This linear transformation optimally preserves the local neighborhood information, as well as discriminant information. Specifically, at each local neighborhood, the margin between data points from different classes is maximized.

The paper is structured as follows: in Section 2, we provide a brief review of Linear Discriminant Analysis. The Locality Sensitive Discriminant Analysis (LSDA) algorithm is introduced in Section 3. In Section 4, we describe how to perform LSDA in Reproducing Kernel Hilbert Space (RKHS) which gives rise to kernel LSDA. The experimental results are presented in Section 5. Finally, we provide some concluding remarks in Section 6.

2 Related Works

The generic problem of linear dimensionality reduction is the following. Given a set $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m$ in $\mathbb{R}^n$, find a transformation matrix $A = (\mathbf{a}_1, \cdots, \mathbf{a}_d) \in \mathbb{R}^{n \times d}$ that maps these m points to a set of points $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_m$ in $\mathbb{R}^d$ ($d \ll n$), such that $\mathbf{y}_i$ "represents" $\mathbf{x}_i$, where $\mathbf{y}_i = A^T \mathbf{x}_i$.

Linear Discriminant Analysis (LDA) seeks directions that are efficient for discrimination. Suppose these data points belong to c classes and each point is associated with a label $l(\mathbf{x}_i) \in \{1, 2, \cdots, c\}$. The objective function of LDA is as follows:

$$\mathbf{a}_{opt} = \arg\max_{\mathbf{a}} \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_w \mathbf{a}} \tag{1}$$

$$S_b = \sum_{i=1}^{c} m_i (\boldsymbol{\mu}^i - \boldsymbol{\mu})(\boldsymbol{\mu}^i - \boldsymbol{\mu})^T \tag{2}$$

$$S_w = \sum_{i=1}^{c} \left(\sum_{j=1}^{m_i} (\mathbf{x}_j^i - \boldsymbol{\mu}^i)(\mathbf{x}_j^i - \boldsymbol{\mu}^i)^T \right) \tag{3}$$

where $\boldsymbol{\mu}$ is the total sample mean vector, m_i is the number of samples in the i-th class, $\boldsymbol{\mu}^i$ is the average vector of the i-th class, and $\mathbf{x}_j^i$ is the j-th sample in the i-th class. We call S_w the within-class scatter matrix and S_B the between-class scatter matrix. The basis functions of LDA are the eigenvectors of the following generalized eigen-problem associated with the largest eigenvalues:

$$S_b \mathbf{a} = \lambda S_w \mathbf{a} \tag{4}$$

Clearly, LDA aims to preserve the *global* class relationship between data points, while it fails to discover the intrinsic *local* geometrical structure of the data manifold. In many real world applications such as face recognition, there may not be sufficient training samples. In this case, it may not be able to accurately estimate the global structure and the local structure becomes more important.

Besides LDA, there is recently a lot of interests in graph based linear dimensionality reduction. The typical algorithms includes Locality Preserving Projections (LPP, [He and Niyogi, 2003]), Local Discriminant Embedding (LDE, [Chen *et al.*, 2005]), Marginal Fisher Analysis (MFA, [Yan *et al.*, 2005]), etc. LPP uses one graph to model the geometrical structure in the data. LDE and MFA are essentially the same. Both of them uses two graphs to model the discriminant structure in the data. However, these two algorithms implicitly consider that the within-class and between-class relations are equally important. This reduces the flexibility of the algorithms.

3 Locality Sensitive Discriminant Analysis

In this section, we introduce our *Locality Sensitive Discriminant Analysis* algorithm which respects both discriminant and geometrical structure in the data. We begin with a description of the locality sensitive discriminant objective function.

3.1 The Locality Sensitive Discriminant Objective Function for Dimensionality Reduction

As we described previously, naturally occurring data may be generated by structured systems with possibly much fewer degrees of freedom than the ambient dimension would suggest. Thus we consider the case when the data lives on or close to a submanifold of the ambient space. One hopes then to estimate geometrical and discriminant properties of the submanifold from random points lying on this unknown submanifold. In this paper, we consider the particular question of maximizing *local* margin between different classes.

Given m data points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\} \subset \mathbb{R}^n$ sampled from the underlying submanifold $\mathcal{M}$, one can build a nearest neighbor graph G to model the local geometrical structure of $\mathcal{M}$. For each data point $\mathbf{x}_i$, we find its k nearest neighbors and put an edge between $\mathbf{x}_i$ and its neighbors. Let $N(\mathbf{x}_i) = \{\mathbf{x}_i^1, \cdots, \mathbf{x}_i^k\}$ be the set of its k nearest neighbors. Thus, the weight matrix of G can be defined as follows:

$$W_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The nearest neighbor graph G with weight matrix W characterizes the local geometry of the data manifold. It has been frequently used in manifold based learning techniques, such as [Belkin and Niyogi, 2001], [Tenenbaum *et al.*, 2000], [Roweis and Saul, 2000], [He and Niyogi, 2003]. However, this graph fails to discover the discriminant structure in the data.

In order to discover both geometrical and discriminant structure of the data manifold, we construct two graphs, i.e. *within-class graph* G_w and *between-class graph* G_b. Let $l(\mathbf{x}_i)$ be the class label of $\mathbf{x}_i$. For each data point $\mathbf{x}_i$, the set $N(\mathbf{x}_i)$ can be naturally split into two subsets, $N_b(\mathbf{x}_i)$ and $N_w(\mathbf{x}_j)$. $N_w(\mathbf{x}_i)$ contains the neighbors sharing the same label with $\mathbf{x}_i$, while $N_b(\mathbf{x}_i)$ contains the neighbors having different labels. Specifically,

$$N_w(\mathbf{x}_i) = \{\mathbf{x}_i^j | l(\mathbf{x}_i^j) = l(\mathbf{x}_i), 1 \le j \le k\}$$

$$N_b(\mathbf{x}_i) = \{\mathbf{x}_i^j | l(\mathbf{x}_i^j) \neq l(\mathbf{x}_i), 1 \le j \le k\}$$

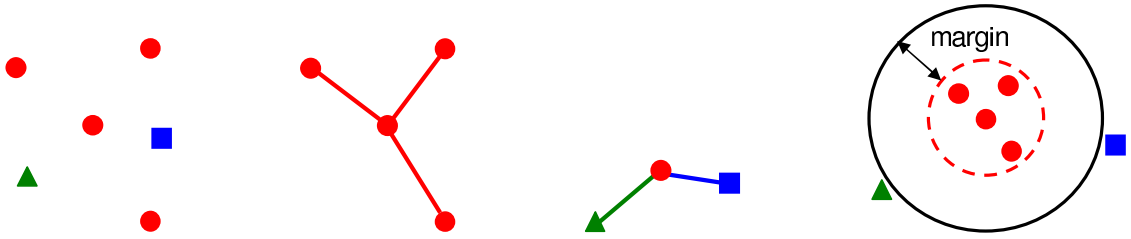

Figure 1: (a) The center point has five neighbors. The points with the same color and shape belong to the same class. (b) The *within-class graph* connects nearby points with the same label. (c) The *between-class graph* connects nearby points with different labels. (d) After Locality Sensitive Discriminant Analysis, the margin between different classes is maximized.

Clearly, $N_b(\mathbf{x}_i) \cap N_w(\mathbf{x}_i) = \varnothing$ and $N_b(\mathbf{x}_i) \cup N_w(\mathbf{x}_i) = N(\mathbf{x}_i)$. Let W_w and W_b be the weight matrices of G_w and G_b, respectively. We define:

$$W_{b,ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N_b(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_b(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

$$W_{w,ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N_w(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_w(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

It is clear to see $W = W_b + W_w$ and the nearest neighbor graph G can be thought of as a combination of within-class graph G_w and between-class graph G_b.

Now consider the problem of mapping the within-class graph and between-class graph to a line so that connected points of G_w stay as close together as possible while connected points of G_b stay as distant as possible. Let $\mathbf{y} = (y_1, y_2, \cdots, y_m)^T$ be such a map. A reasonable criterion for choosing a "good" map is to optimize the following two objective functions:

$$\min \sum_{ij} (y_i - y_j)^2 W_{w,ij} \tag{8}$$

$$\max \sum_{ij} (y_i - y_j)^2 W_{b,ij} \tag{9}$$

under appropriate constraints. The objective function (8) on within-class graph incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart while they are actually in the same class. Likewise, the objective function (9) on between-class graph incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped close together while they actually belong to different classes. Therefore, minimizing (8) is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close and sharing the same label then y_i and y_j are close as well. Also, maximizing (9) is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close but have different labels then y_i and y_j are far apart. The learning procedure is illustrated in Figure 1.

3.2 Optimal Linear Embedding

In this subsection, we describe our Locality Sensitive Discriminant Analysis algorithm which solves the objective functions (8) and (9). Suppose $\mathbf{a}$ is a projection vector, that is, $\mathbf{y}^T = \mathbf{a}^T X$, where $X = (\mathbf{x}_1, \cdots, \mathbf{x}_m)$ is a $n \times m$ matrix.

By simple algebra formulation, the objective function (8) can be reduced to

$$\begin{aligned} & \frac{1}{2} \sum_{ij} (y_i - y_j)^2 W_{w,ij} \\ = & \frac{1}{2} \sum_{ij} \left(\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_j\right)^2 W_{w,ij} \\ = & \sum_i \mathbf{a}^T \mathbf{x}_i D_{w,ii} \mathbf{x}_i^T \mathbf{a} - \sum_{ij} \mathbf{a}^T \mathbf{x}_i W_{w,ij} \mathbf{x}_j^T \mathbf{a} \\ = & \mathbf{a}^T X D_w X^T \mathbf{a} - \mathbf{a}^T X W_w X^T \mathbf{a} \end{aligned}$$

where D_w is a diagonal matrix; its entries are column (or row, since W_w is symmetric) sum of W_w, $D_{w,ii} = \sum_j W_{w,ij}$. Similarly, the objective function (9) can be reduced to

$$\begin{aligned} & \frac{1}{2} \sum_{ij} (y_i - y_j)^2 W_{b,ij} \\ = & \frac{1}{2} \sum_{ij} \left(\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_j\right)^2 W_{b,ij} \\ = & \mathbf{a}^T X (D_b - W_b) X^T \mathbf{a} \\ = & \mathbf{a}^T X L_b X^T \mathbf{a} \end{aligned}$$

where D_b is a diagonal matrix; its entries are column (or row, since W_b is symmetric) sum of W_b, $D_{b,ii} = \sum_j W_{b,ij}$. $L_b = D_b - W_b$ is the Laplacian matrix of G_b.

Note that, the matrix D_w provides a natural measure on the data points. If $D_{w,ii}$ is large, then it implies that the class containing $\mathbf{x}_i$ has a high density around $\mathbf{x}_i$. Therefore, the bigger the value of $D_{w,ii}$ is, the more "important" is $\mathbf{x}_i$. Therefore, we impose a constraint as follows:

$$\mathbf{y}^T D_w \mathbf{y} = 1 \Rightarrow \mathbf{a}^T X D_w X^T \mathbf{a} = 1$$

Thus, the objective function (8) becomes the following:

$$\min_{\mathbf{a}} \; 1 - \mathbf{a}^T X W_w X^T \mathbf{a} \tag{10}$$

or equivalently,

$$\max_{\mathbf{a}} \; \mathbf{a}^T X W_w X^T \mathbf{a} \tag{11}$$

And the objective function (9) can be rewritten as follows:

$$\max_{\mathbf{a}} \; \mathbf{a}^T X L_b X^T \mathbf{a} \tag{12}$$

Finally, the optimization problem reduces to finding:

$$\arg \max_{\substack{\mathbf{a} \\ \mathbf{a}^T X D_w X^T \mathbf{a} = 1}} \mathbf{a}^T X \big(\alpha L_b + (1-\alpha) W_w\big) X^T \mathbf{a} \qquad (13)$$

where α is a suitable constant and $0 \leq \alpha \leq 1$. The projection vector $\mathbf{a}$ that minimizes (13) is given by the maximum eigenvalue solution to the generalized eigenvalue problem:

$$X\big(\alpha L_b + (1-\alpha) W_w\big) X^T \mathbf{a} = \lambda X D_w X^T \mathbf{a} \qquad (14)$$

Let the column vector $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_d$ be the solutions of equation (14), ordered according to their eigenvalues, $\lambda_1 > \cdots > \lambda_d$. Thus, the embedding is as follows:

$$\mathbf{x}_i \rightarrow \mathbf{y}_i = A^T \mathbf{x}_i$$
$$A = (\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_d)$$

where $\mathbf{y}_i$ is a d-dimensional vector, and A is a $n \times d$ matrix.

Note that, if the number of samples (m) is less than the number of features (n), then $rank(X) \leq m$. Consequently, $rank(XD_wX^T) \leq m$ and $rank\big(X(\alpha L_b + (1-\alpha)W_w)X^T\big) \leq m$. The fact that XD_wX^T and $X(\alpha L_b + (1-\alpha)W_w)X^T$ are $n \times n$ matrices implies that both of them are singular. In this case, one may first apply Principal Component Analysis to remove the components corresponding to zero eigenvalues.

4 Kernel LSDA

LSDA is a linear algorithm. It may fail to discover the intrinsic geometry when the data manifold is highly nonlinear. In this section, we discussion how to perform LSDA in Reproducing Kernel Hilbert Space (RKHS), which gives rise to kernel LSDA.

Suppose $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\} \in \mathcal{X}$ is the training sample set. We consider the problem in a feature space $\mathcal{F}$ induced by some nonlinear mapping

$$\phi : \mathcal{X} \rightarrow \mathcal{F}$$

For a proper chosen ϕ, an inner product $\langle , \rangle$ can be defined on $\mathcal{F}$ which makes for a so-called reproducing kernel Hilbert space (RKHS). More specifically,

$$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \mathcal{K}(\mathbf{x}, \mathbf{y})$$

holds where $\mathcal{K}(.,.)$ is a positive semi-definite kernel function. Several popular kernel functions are: Gaussian kernel $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/\sigma^2)$; polynomial kernel $\mathcal{K}(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^d$; Sigmoid kernel $\mathcal{K}(\mathbf{x}, \mathbf{y}) = tanh(\langle \mathbf{x}, \mathbf{y} \rangle + \alpha)$.

Given a set of vectors $\{\mathbf{v}_i \in \mathcal{F} | i = 1, 2, \cdots, d\}$ which are orthonormal ($\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{i,j}$), the projection of $\phi(\mathbf{x}_i) \in \mathcal{F}$ to these $\mathbf{v}_1, \cdots, \mathbf{v}_d$ leads to a mapping from $\mathcal{X}$ to Euclidean space $\mathbb{R}^d$ through

$$\mathbf{y}_i = \big(\langle \mathbf{v}_1, \phi(\mathbf{x}_i) \rangle, \langle \mathbf{v}_2, \phi(\mathbf{x}_i) \rangle, \cdots, \langle \mathbf{v}_d, \phi(\mathbf{x}_i) \rangle\big)^T$$

We look for such $\{\mathbf{v}_i \in \mathcal{F} | i = 1, 2, \cdots, d\}$ that helps $\{\mathbf{y}_i | i = 1, \cdots, m\}$ preserve local geometrical and discriminant structure of the data manifold. A typical scenario is $\mathcal{X} = \mathbb{R}^n, \mathcal{F} = \mathbb{R}^\theta$ with $d << n < \theta$.

Let Φ denote the data matrix in RKHS:

$$\Phi = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_m)]$$

Now, the eigenvector problem in RKHS can be written as follows:

$$\Phi\big(\alpha L_b + (1-\alpha) W_w\big) \Phi^T \mathbf{v} = \lambda \Phi D_w \Phi^T \mathbf{v} \qquad (15)$$

Because the eigenvector of (15) are linear combinations of $\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_m)$, there exist coefficients $\alpha_i, i = 1, 2, \cdots, m$ such that

$$\mathbf{v} = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}_i) = \Phi \boldsymbol{\alpha}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_m)^T \in \mathbb{R}^m$.

Following some algebraic formulations, we get:

$$\begin{aligned}
& \Phi\big(\alpha L_b + (1-\alpha) W_w\big) \Phi^T \mathbf{v} = \lambda \Phi D_w \Phi^T \mathbf{v} \\
\Rightarrow \quad & \Phi\big(\alpha L_b + (1-\alpha) W_w\big) \Phi^T \Phi \boldsymbol{\alpha} = \lambda \Phi D_w \Phi^T \Phi \boldsymbol{\alpha} \\
\Rightarrow \quad & \Phi^T \Phi\big(\alpha L_b + (1-\alpha) W_w\big) \Phi^T \Phi \boldsymbol{\alpha} \\
& = \lambda \Phi^T \Phi D_w \Phi^T \Phi \boldsymbol{\alpha} \\
\Rightarrow \quad & K\big(\alpha L_b + (1-\alpha) W_w\big) K \boldsymbol{\alpha} = \lambda K D_w K \boldsymbol{\alpha} \qquad (16)
\end{aligned}$$

where K is the kernel matrix, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Let the column vectors $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \cdots, \boldsymbol{\alpha}^m$ be the solutions of equation (16). For a test point $\mathbf{x}$, we compute projections onto the eigenvectors $\mathbf{v}^k$ according to

$$(\mathbf{v}^k \cdot \phi(\mathbf{x})) = \sum_{i=1}^{m} \alpha_i^k (\phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i)) = \sum_{i=1}^{m} \alpha_i^k \mathcal{K}(\mathbf{x}, \mathbf{x}_i)$$

where α_i^k is the i^{th} element of the vector $\boldsymbol{\alpha}^k$. For the original training points, the map can be obtained by $\mathbf{y} = K\boldsymbol{\alpha}$, where the i^{th} element of $\mathbf{y}$ is the one-dimensional representation of $\mathbf{x}_i$.

5 Experimental Results

In this Section, we investigate the use of LSDA on face recognition. We compare our proposed algorithm with Eigenface (PCA, [Turk and Pentland, 1991]), Fisherface (LDA, [Belhumeur *et al.*, 1997]) and Marginal Fisher Analysis (MFA, [Yan *et al.*, 2005]). We begin with a brief discussion about data preparation.

5.1 Data Preparation

Two face databases were tested. The first one is the Yale database[1], and the second one is the ORL database[2]. In all the experiments, preprocessing to locate the faces was applied. Original images were normalized (in scale and orientation) such that the two eyes were aligned at the same position. Then, the facial areas were cropped into the final image for matching. The size of each cropped image in all the experiments is 32×32 pixels, with 256 gray levels per pixel. Thus,

[1] `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`

[2] `http://www.cl.cam.ac.uk/Research/DTG/attarchive/facesataglance.html`

Figure 2: Sample face images from the Yale database. For each subject, there are 11 face images under different lighting conditions with facial expression.

Table 1: Recognition accuracy of different algorithms on the Yale database

Method	2 Train	3 Train	4 Train	5 Train
Baseline	43.4%(1024)	49.4%(1024)	52.6%(1024)	56.2%(1024)
Eigenfaces	43.4%(29)	49.4%(44)	52.6%(58)	56.2%(74)
Fisherfaces	47.2%(10)	64.9%(14)	72.9%(14)	78.8%(14)
MFA	47.7%(10)	65.7%(14)	74.1%(14)	78.9%(14)
LSDA	**56.5%(14)**	**68.5%(14)**	**74.4%(14)**	**79.0%(14)**

each image can be represented by a 1024-dimensional vector in image space. No further preprocessing is done. Different pattern classifiers have been applied for face recognition, including nearest neighbor [Turk and Pentland, 1991], Bayesian [Moghaddam, 2002], and Support Vector Machines [Phillips, 1998], etc. In this paper, we apply nearest neighbor classifier for its simplicity. In our experiments, the number of nearest neighbors (k) is taken to be 5. The parameter α is estimated by leave one out cross validation.

In short, the recognition process has three steps. First, we calculate the face subspace from the training set of face images; then the new face image to be identified is projected into d-dimensional subspace; finally, the new face image is identified by nearest neighbor classifier.

5.2 Face Recognition on Yale Database

The Yale face database is constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. The images demonstrate variations in lighting condition (left-light, center-light, right-light), facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. Figure 2 shows some sample images of one individual.

For each individual, $l(= 2, 3, 4, 5)$ images were randomly selected as training samples, and the rest were used for testing. The training set was used to learn a face subspace using the LSDA, Eigenface, and Fisherface methods. Recognition was then performed in the subspaces. We repeated this process 20 times and calculate the average recognition rate. In general, the recognition rates varies with the dimension of the face subspace. The best performance obtained by these algorithms as well as the corresponding dimensionality of the optimal subspace are shown in Table 1. For the baseline method, we simply performed face recognition in the original 1024-dimensional image space. Note that, the upper bound of the dimensionality of Fisherface is $c - 1$ where c is the number of individuals [Duda *et al.*, 2000].

As can be seen, our algorithm outperformed all other three methods. The Eigenface method performs the worst in all cases. It does not obtain any improvement over the baseline method. It would be interesting to note that, when there are only two training samples for each individual, the best performance of Fisherface is no longer obtained in a $c - 1(= 14)$ dimensional subspace, but a 10-dimensional subspace. LSDA reaches the best performance almost always at $c - 1$ dimensions. This property shows that LSDA does not suffer from the problem of dimensionality estimation which is a crucial problem for most of the subspace learning based face recognition methods.

5.3 Face Recognition on ORL Database

The ORL (Olivetti Research Laboratory) face database is used in this test. It consists of a total of 400 face images, of a total of 40 people (10 samples per person). The images were captured at different times and have different variations including expressions (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees. 10 sample images of one individual are displayed in Figure 3. For each individual, $l(= 2, 3, 4, 5)$ images are randomly selected for training and the rest are used for testing.

The experimental design is the same as before. For each given l, we average the results over 20 random splits. The best result obtained in the optimal subspace and the corresponding dimensionality for each method are shown in Table 2.

As can be seen, our LSDA algorithm performed the best for all the cases. The Fisherface method performed comparatively to LSDA as the size of the training set increases. Moreover, the optimal dimensionality obtained by LSDA and Fisherface is much lower than that obtained by Eigenface.

5.4 Discussion

Several experiments on two standard face databases have been systematically performed. These experiments have revealed a number of interesting points:

1. All the three algorithms (LSDA, MFA, and Fisherface) performed better in the optimal face subspace than in the original image space. This indicates that dimensionality reduction can discover the intrinsic structure of the face manifold and hence improve the recognition rate.
2. In all the experiments, our LSDA algorithm consistently outperformed the Eigenface, Fisherface and MFA methods. Especially when the size of the training set is small, LSDA significantly outperformed Fisherface. This is probably due to the fact that Fisherface fails to accurately estimate the within-class scatter matrix from only a small number of training samples.

Figure 3: Sample face images from the ORL database. For each subject, there are 10 face images with different facial expression and details.

Table 2: Recognition accuracy of different algorithms on the ORL database

Method	2 Train	3 Train	4 Train	5 Train
Baseline	66.8%(1024)	77.0%(1024)	81.7%(1024)	86.6%(1024)
Eigenfaces	66.8%(79)	77.0%(119)	81.7%(159)	86.6%(198)
Fisherfaces	71.3%(28)	83.4%(39)	89.6%(39)	93.2%(39)
MFA	71.6%(37)	84.1%(39)	89.7%(39)	93.1%(39)
LSDA	**76.7%(39)**	**85.0%(39)**	**90.5%(39)**	**93.6%(39)**

3. Eigenface fails to gain improvement over the baseline. This is probably because that Eigneface does not encode the discriminating information.
4. In all the experiments, the optimal dimensionality obtained by LSDA is always c-1, where c is the number of classes. In practice, when the computational complexity is a major concern, one can simply project the face images into a c-1 dimensional subspace.

6 Conclusion

We have introduced a novel linear dimensionality reduction algorithm called Locality Sensitive Discriminant Analysis (LSDA). For the class of spectrally based dimensionality reduction techniques, it optimizes a fundamentally different criterion compared to classical dimensionality reduction approaches based on Fisher's criterion (LDA) or Principal Component Analysis. The most prominent property of LSDA is the complete preservation of both discriminant and local geometrical structure in the data. For LDA, on the other hand, it can only preserve the global discriminant structure, while the local geometrical structure is ignored. We have applied our algorithm to face recognition. Experiments on Yale and ORL databases have been conducted to demonstrate the effectiveness of our algorithm.

References

[Belhumeur *et al.*, 1997] Peter N. Belhumeur, J. P. Hepanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[Belkin and Niyogi, 2001] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, Cambridge, MA, 2001.

[Belkin *et al.*, 2004] M. Belkin, P. Niyogi, and V. Sindhwani. On maniold regularization. Technical report tr-2004-05, Computer Science Department, The University of Chicago, 2004.

[Chen *et al.*, 2005] Hwann-Tzong Chen, Huang-Wei Chang, and Tyng-Luh Liu. Local discriminant embedding and its variants. In *Proc. 2005 Internal Conference on Computer Vision and Pattern Recognition*, 2005.

[Chung, 1997] Fan R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS, 1997.

[Duda *et al.*, 2000] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2nd edition, 2000.

[He and Niyogi, 2003] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2003.

[Moghaddam, 2002] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6), 2002.

[Phillips, 1998] P. J. Phillips. Support vector machines applied to face recognition. *Advances in Neural Information Processing Systems*, 11:803–809, 1998.

[Roweis and Saul, 2000] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[Tenenbaum *et al.*, 2000] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[Turk and Pentland, 1991] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[Yan *et al.*, 2005] Shuicheng Yan, Dong Xu, Benyu Zhang, and Hong-Jiang Zhang. Graph embedding: A general framework for dimensionality reduction. In *Proc. 2005 Internal Conference on Computer Vision and Pattern Recognition*, 2005.

[Zhou *et al.*, 2003] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, 2003.