Knowledge-Based Systems 31 (2012) 119–127

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Face recognition using discriminant sparsity neighborhood preserving embedding

Gui-Fu Lu [a,b,*], Zhong Jin [a], Jian Zou [a,b]

[a] School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China
[b] School of Computer Science and Information, AnHui Polytechnic University, WuHu, AnHui 241000, China

ARTICLE INFO

Article history:
Received 3 April 2011
Received in revised form 30 January 2012
Accepted 27 February 2012
Available online 4 March 2012

Keywords:
Sparse representation
Dimensionality reduction
Graph embedding
Feature extraction
Face recognition

ABSTRACT

In this paper, we propose an effective supervised dimensionality reduction technique, namely discriminant sparsity neighborhood preserving embedding (DSNPE), for face recognition. DSNPE constructs graph and corresponding edge weights simultaneously through sparse representation (SR). DSNPE explicitly takes into account the within-neighboring information and between-neighboring information. Further, by taking the advantage of the maximum margin criterion (MMC), the discriminating power of DSNPE is further boosted. Experiments on the ORL, Yale, AR and FERET face databases show the effectiveness of the proposed DSNPE.

1. Introduction

Face recognition has attracted many researchers in the area of pattern recognition, machine learning, and computer vision because of its immense application potential. Numerous methods have been proposed in the last two decades [1–10]. One of the most successful and well-studied techniques to face recognition is the appearance-based method. In an appearance-based technique, a face image with size $w \times h$ is perceived as a point in a $w \times h$ dimensional image space. In practice, however, these $w \times h$-dimensional spaces are too large to allow robust and fast recognition. Dimensionality reduction is an effective approach to deal with this problem. The most well-know dimensionality reduction methods are principal component analysis (PCA) [11] and linear discriminant analysis (LDA) [12].

PCA is based on the computation of low-dimensional representation of high-dimensional data that maximizes the total scatter. However, PCA does not utilize the class label information. LDA aims to better discriminate patterns of different classes by searching the projection axes on which the data points of different classes are far from each other, while constraining the data points of the same class to be as close to each other as possible. Unfortunately, LDA cannot be applied directly to small size sample (SSS) problem because the within-class scatter matrix is singular [13]. To avoid the singularity problem of LDA, Li et al. [14] used the difference of both between-class scatter and within-class scatter as discriminant criterion, called maximum margin criterion (MMC). MMC has the advantages of effectiveness and simplicity.

Both PCA and LDA have been successfully applied to some linear data. However, they fail to explore the essential structure of the data with non-linear distribution. Kernel trick is one commonly used approach to handle non-linearity structure in data. The key idea of kernel methods is to map the original data to a higher-dimensional feature space where the inner products can be computed by a kernel function without knowing the non-linear mapping function explicitly [15,16]. The widely used kernel techniques are kernel principal component analysis (KPCA) [17] and kernel Fisher discriminant analysis (KFD) [18], which can be viewed as the kernel version of PCA and LDA, respectively. However, how to select kernel and assign optimal kernel parameter is generally difficult. In most of the cases, experience still plays an important role. In [19–23] some methods based on Gabor filters have been introduced. These methods are robust to illumination changes and varying pose.

Recently, a number of manifold learning algorithms have been developed. Representative ones include locally linear embedding (LLE) [24], Isomap [25], Laplacian eigenmaps (LE) [26], and local tangent space alignment (LTSA) [27]. Based on the assumption of the local linearity, LLE first constitutes local coordinates with the least constructed cost and then maps them to a global one. Isomap determines a low-dimensional representation of the data set that aims to preserve geodesic distances between pairs of data points. LE preserves proximity relationships by manipulations on an undirected weighted graph, which indicates neighbor relations of pairwise measurements. LTSA uses the local tangent space to represent the local geometry of the essential manifold structure.

* Corresponding author at: School of Computer Science and Information, AnHui Polytechnic University, WuHu, AnHui 241000, China.
E-mail address: luguifu_jsj@163.com (G.-F. Lu).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2012.02.014

Unfortunately, all of these algorithms are plagued by the out-of-sample problem [28]. One common response to this problem is to apply a linearization procedure to construct explicit maps over new measurements. For example, LPP [29] is a linearization version of LE; neighborhood preserving embedding (NPE) [30] is a linearization version of LLE; isometric projection (IsoProjection) [31] can be seen as a linearized Isomap; and linear local tangent space alignment (LLTSA) [32] is a linearization of LTSA.

Recently, Yan et al. [33] introduced a general framework for dimensionality reduction, called graph embedding, where a large number of popular dimensionality reduction, e.g., PCA, LDA, Isomap, LLE, and Laplacian Eigenmap, could be considered as special cases within the framework. Based on the graph embedding, some discriminant manifold learning methods, e.g. marginal Fisher analysis (MFA) [33], neighborhood preserving discriminant embedding (NPDE) [34] and locality discriminanting projection (LDP) [35], have been proposed to improve the recognition performance. As a result, graph becomes the heart of the most dimensionality reduction methods. However, the way to establish high-quality graphs is still an open problem [36]. At present, there exist two popular ways for graph construction, one of which is the k-nearest-neighbor, and the other is the ε-ball based method. Motivated by the recent development of sparse representation (SR) [37,38], some research works, e.g. sparsity preserving projections (SPP) [39] and L1-graph [40], attempted to construct graph and corresponding edge weights simultaneously through SR. SPP firstly constructs an "adjacent" weight matrix of the data set based on SR, and then evaluate the low-dimensional embedding of the data to best preserve such weight matrix. Although SPP is effective in many domains, it is unsupervised and its unsupervised nature restricts its discriminating capability.

In this paper, we propose an effective supervised manifold learning algorithm, called discriminant sparsity neighborhood preserving embedding (DSNPE). The proposed DSNPE incorporates graph embedding and MMC for data analysis. Similar to SPP, DSNPE constructs graph and corresponding edge weights simultaneously through SR. More importantly, DSNPE explicitly takes into account the within-neighboring information, which is modeled by the sparse reconstruction weights of the samples from the same class, and between-neighboring information, which is modeled by the sparse reconstruction weights of the neighboring samples from different classes.

The organization of the rest of this paper is as follows. In Section 2, we review briefly NPE, sparse representation and SPP. In Section 3, we propose the discriminant sparsity neighborhood preserving embedding (DSNPE) and describe the proposed method in detail. In Section 4, we compare DSNPE with some related works. The experimental results are presented in Section 5. Conclusions are made in Section 6.

2. Related works

In this section, we will firstly review briefly principal component analysis since PCA is the most popular dimensionality reduction method and it is often used to transform the original image into a lower dimensional subspace to avoid SSS problem. Then we will review briefly neighborhood preserving embedding (NPE), sparse representation (SR) and sparsity preserving projections (SPP) since our proposed DSNPE are stemmed from these there methods.

2.1. Principal component analysis (PCA)

Let $\{\mathbf{x}_i \in \mathbb{R}^m | i = 1, \ldots, n\}$ represents the input data as an m-dimensional data point in an Euclidean vector space, and the projection vector is $\{\mathbf{y}_i \in \mathbb{R}^d | i = 1, \ldots, n\}$, where $d \ll m$. Assume that the original data in $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ is partitioned into c classes as $X = [X_1, X_2, \ldots, X_c]$, where $X_i \in \mathbb{R}^{m \times n_i}$ contains data points from the ith class and $\sum_{i=1}^{c} n_i = n$. The objective function of PCA is defined as follows:

$$\max_{\|\mathbf{a}\|=1} \sum_{i=1}^{n} (\mathbf{y}_i - \bar{\mathbf{y}})^2 \tag{1}$$

where $\mathbf{y} = \mathbf{a}^T \mathbf{x}_i$ and $\bar{\mathbf{y}}$ is the mean of $\{\bar{\mathbf{y}}\}_{i=1}^{n}$. Eq. (1) can be rewritten as

$$\max_{\|\mathbf{a}\|=1} \mathbf{a}^T \Sigma \mathbf{a} \tag{2}$$

where $\Sigma = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$ is the sample covariance matrix, here $\bar{\mathbf{x}}$ is the mean of all training samples. The eigenvectors of Σ corresponding to the largest d eigenvalues span the optimal subspace of PCA.

2.2. Neighborhood preserving embedding (NPE)

NPE, which is based on simple geometric intuitions, i.e., each data points and its neighbors lie on or close to a locally linear patch of some underlying manifold [24], evaluates the affinity weight matrix using local least squares approximation. The local approximation error in NPE is measured by minimizing the cost function [30]:

$$\phi(W) = \sum_i \left\| \mathbf{x}_i - \sum_j W_{ij} \mathbf{x}_j \right\|^2 \tag{3}$$

under two constraints: a sparseness constraint, i.e. $W_{ij} = 0$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are not neighbors, and an invariance constraint, i.e. $\sum_j W_{ij} = 1$. A reasonable criterion for choosing a "good" projection is minimizing the cost function [30]:

$$\phi(\mathbf{a}) = \sum_i \left\| \mathbf{a}^T \mathbf{x}_i - \sum_j W_{ij} \mathbf{a}^T \mathbf{x}_j \right\|^2 \tag{4}$$

By removing an arbitrary scaling factor, minimizing Eq. (4) leads to

$$\min_{\mathbf{a}} \frac{\mathbf{a}^T X M X^T \mathbf{a}}{\mathbf{a} X X^T \mathbf{a}} \tag{5}$$

where $M = (I - W)^T (I - W)$ is a symmetric, and semi-positive definite matrix, I is an identity matrix.

Using Lagrange multipliers and it leads to the following generalized eigenvector problem:

$$X M X^T \mathbf{a} = \lambda X X^T \mathbf{a} \tag{6}$$

Let the column vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_d$ be the solutions of Eq. (6), ordered according to their eigenvalues, $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \lambda_d$, and $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_d] \in \mathbb{R}^{m \times d}$. Thus, the embedding is as follows:

$$\mathbf{x}_i \to \mathbf{y}_i = A^T \mathbf{x}_i \tag{7}$$

The details of theoretical justification about NPE can be found in [30].

2.3. Sparse representation (SR)

Suppose we have an underdetermined system of linear equations: $\mathbf{x} = X\mathbf{s}$, where $\mathbf{x} \in \mathbb{R}^m$ is the vector to be approximated, $\mathbf{s} \in \mathbb{R}^n$ is the coefficients vector, $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ is the overcomplete dictionary with n bases. The goal of SR is to represent $\mathbf{x}$ using as few entries of X as possible, which can be obtained by solving the following optimization problem:

$$\min_{\mathbf{s}} \|\mathbf{s}\|_0, \ \text{s.t.} \quad \mathbf{x} = X\mathbf{s} \tag{8}$$

where $\| \ \|_0$ is the ℓ_0-norm which is equal to the number of non-zero components in a vector and s.t. stands for subject to. Unfortunately

it is NP-hard to find the sparsest solution of Eq. (8). Recently results [37,41] reveal that if the solution is sparse enough, the solution to the ℓ_0-minimization problem is equal to the solution to ℓ_1-minimization problem. Then the ℓ_0-minimization problem (8) is equal to the following ℓ_1-minimization problem:

$$\min_{\mathbf{s}} \|\mathbf{s}\|_1, \text{ s.t. } \mathbf{x} = X\mathbf{s} \tag{9}$$

where $\|\ \|_1$ is the ℓ_1-norm.[1] In many practical problems, the signal $\mathbf{x}$ may be noisy, and the following models can be used to estimate $\mathbf{s}$ [37]

$$\min_{s} \|\mathbf{s}\|_1, \text{ s.t. } \|\mathbf{x} - X\mathbf{s}\|_2 < \varepsilon \tag{10}$$

where ε is an error tolerance, or

$$\min_{\begin{bmatrix}\mathbf{s}\\ \zeta\end{bmatrix}} \left\| \begin{bmatrix}\mathbf{s}\\ \zeta\end{bmatrix} \right\|_1, \text{ s.t. } \mathbf{x} = [X \quad I] \begin{bmatrix}\mathbf{s}\\ \zeta\end{bmatrix} \tag{11}$$

where I is an m-order identity matrix and $\zeta \in \mathbb{R}^m$ is the noise term or error term.

2.4. Sparsity preserving projections (SPP)

SPP [39] firstly seeks a sparse reconstructive weight vector $\mathbf{s}_i$ for each $\mathbf{x}_i$ through the following modified ℓ_1 minimization problem:

$$min_{\mathbf{s}_i} \|\mathbf{s}_i\|_1, \text{ s.t. } \mathbf{x}_i = X\mathbf{s}_i, \quad 1 = \mathbf{1}^T\mathbf{s}_i \tag{12}$$

where $\mathbf{s}_i = [s_{i,1}, \ldots, s_{i,i-1}, 0, s_{i,i+1}, \ldots, s_{i,n}]^T$ is an n-dimensional vector and $\mathbf{1} = \underbrace{[1, 1, \ldots, 1]}_{n}^T \in \mathbb{R}^n$ is a vector of all ones. Note the ith element in $\mathbf{s}_i$ is zero, which implies that the $\mathbf{x}_i$ is removed from X. Then the optimal weight vector $\tilde{\mathbf{s}}_i$ obtained from Eq. (12) is used to the following objective function:

$$\min \sum_{i=1}^{n} \|\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^TX\tilde{\mathbf{s}}_i\| = \min \mathbf{a}^TX(I-S)^T(I-S)X^T\mathbf{a} \tag{13}$$

where $S = [\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \ldots, \tilde{\mathbf{s}}_n]^T$. And, the optimal $\mathbf{a}$ of SPP are the eigenvectors of the following generalized eigenvalue problem[2]:

$$X(I-S)^T(I-S)X^T\mathbf{a} = \lambda XX^T\mathbf{a} \tag{14}$$

3. Discriminant sparsity neighborhood preserving embedding (DSNPE)

In this section, we propose an effective supervised manifold learning algorithm, called discriminant sparsity neighborhood preserving embedding (DSNPE). DSNPE considers two distinct sets of sparse reconstruction weights that are computed from the face data of the same and different persons. Then, both within-neighborhood scatter and between-neighborhood scatter can be formulated based on these sparse reconstruction weights. DSNPE seeks optimal projections that maximize the difference of the trace of between-neighborhood scatter and the trace of within-neighborhood scatter, which is similar to MMC.

3.1. Formulation of between-neighborhood scatter

In order to characterize the interclass separability, for each datum $\mathbf{x}_i$ in the sample set, we only use the samples which are not in the same class as $\mathbf{x}_i$ to reconstruct $\mathbf{x}_i$. Suppose $\mathbf{x}_i$ belongs to the kth class, then, similar to Eqs. (10) and (12), the sparse between-class reconstructive weight vector $\mathbf{s}_i^b$ for each $\mathbf{x}_i$ can be achieved by solving the following ℓ_1-norm optimization problem

$$\min_{\mathbf{s}_i^b} \|\mathbf{s}_i^b\|_1, \text{ s.t. } \left\|\mathbf{x}_i - X^k\mathbf{s}_i^b\right\| < \varepsilon, \quad 1 = \mathbf{1}^T\mathbf{s}_i^b \tag{15}$$

where matrix $X^k = [X_1, X_2, \ldots, X_{k-1}, X_{k+1}, X_c] \in \mathbb{R}^{m\times(n-n_k)}$, $\mathbf{s}_i^b = \left[s_{i,1}^b, s_{i,2}^b, \ldots, s_{i,n-n_k}^b\right]^T \in \mathbb{R}^{n-n_k}$, ε is the error tolerance and $\mathbf{1} \in \mathbb{R}^n$ is a vector of all ones. Note that ε is generally fixed across various instances of the problem [37] and thus in our experiments we simply set it to 0.05 as in [37]. Let W^b denote the between-class weight matrix. Since the coefficient $\mathbf{s}_i^b = \left[s_{i,1}^b, s_{i,2}^b, \ldots, s_{i,n-n_k}^b\right]^T$ for the ℓ_1 reconstruction denotes the contribution of each sample which is not in the same class as $\mathbf{x}_i$ to reconstruct $\mathbf{x}_i$, then $\mathbf{s}_i^b = \left[s_{i,1}^b, s_{i,2}^b, \ldots, s_{i,n-n_k}^b\right]^T$ can reflect the close relation among $\mathbf{x}_i$ and the samples which are not in the same class as $\mathbf{x}_i$. So we use the coefficients $s_{i,1}^b, s_{i,2}^b, \ldots, s_{i,n-n_k}^b$ as the between-class graph weights. Note that $s_{i,1}^b, \ldots, s_{i,n_1+\cdots+n_{k-1}}^b$ reflect the contributions of samples $\mathbf{x}_1, \ldots, \mathbf{x}_{n_1+\cdots+n_{k-1}}$ to reconstruct $\mathbf{x}_i$ and $s_{i,n_1+\cdots+n_{k-1}+1}^b, \ldots, s_{i,n-n_k}^b$ reflect the contributions of samples $\mathbf{x}_{n_1+\cdots+n_{k+1}}, \ldots, \mathbf{x}_n$ to reconstruct $\mathbf{x}_i$, then W_{ij}^b can be defined as

$$W_{ij}^b = \begin{cases} s_{ij}^b, & \text{if } 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} \\ 0, & \text{if } n_1 + \cdots + n_{k-1} + 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} + n_k \\ s_{ij-n_k}^b, & \text{if } n_1 + \cdots + n_k + 1 \leqslant j \leqslant n \end{cases} \tag{16}$$

Note that the graph weights among $\mathbf{x}_i$ and the samples which are in the same class as $\mathbf{x}_i$ are all zeros since we do not use the samples which are in the same class as $\mathbf{x}_i$ to reconstruct $\mathbf{x}_i$. In order to keep the projected samples of different classes far from each other, similar to Eq. (4), we maximize the following cost function

$$\sum_{i=1}^{n} \left\| \mathbf{a}^T\mathbf{x}_i - \sum_{j=1}^{n} W_{ij}^b\mathbf{a}^T\mathbf{x}_j \right\|^2 \tag{17}$$

From Eq. (17) we can get

$$\begin{aligned} \sum_{i=1}^{n} \left\| \mathbf{a}^T\mathbf{x}_i - \sum_{j=1}^{n} W_{ij}^b\mathbf{a}^T\mathbf{x}_j \right\|^2 &= \mathbf{a}^T\left(\sum_{i=1}^{n} \left\|\mathbf{x}_i - XW_i^b\right\|^2\right)\mathbf{a} \\ &= \mathbf{a}^T\left(\sum_{i=1}^{n} \left(\mathbf{x}_i - XW_i^b\right)\left(\mathbf{x}_i - XW_i^b\right)^T\right)\mathbf{a} \end{aligned} \tag{18}$$

where W_i^b is the ith column vector of W^b. Let $\mathbf{e}_i$ be an n-dimensional unit vector with the ith element 1, 0 otherwise, then Eq. (18) can be formulated as

$$\begin{aligned} &\mathbf{a}^T\left(\sum_{i=1}^{n} \left(\mathbf{x}_i - XW_i^b\right)\left(\mathbf{x}_i - XW_i^b\right)^T\right)\mathbf{a} \\ &= \mathbf{a}^TX\left(\sum_{i=1}^{n} \left(\mathbf{e}_i - W_i^b\right)\left(\mathbf{e}_i - W_i^b\right)^T\right)X^T\mathbf{a} \\ &= \mathbf{a}^TX\left(\sum_{i=1}^{n} \mathbf{e}_i\mathbf{e}_i^T - W_i^b\mathbf{e}_i^T - \mathbf{e}_i\left(W_i^b\right)^T + W_i^b\left(W_i^b\right)^T\right)X^T\mathbf{a} \\ &= \mathbf{a}^TXM^bX^T\mathbf{a} \end{aligned} \tag{19}$$

where $M^b = (I - W^b)^T(I - W^b)$.

3.2. Formulation of within-neighborhood scatter

In order to characterize the intraclass compactness, for each datum $\mathbf{x}_i$ in the sample set, we only use the samples which are in the same class as $\mathbf{x}_i$ to reconstruct $\mathbf{x}_i$. Suppose $\mathbf{x}_i$ belongs to the kth

[1] Eq. (9) can be efficiently solved by standard linear programming using publicly available packages such as ℓ_1-magic (http://www.acm.caltech.edu/l1magic) [42].

[2] The Matlab function eig () can be used to calculate the generalized eigenvalue problem.

Table 1
The equation used in the DSNPE algorithm.

Number of equation	Description
Eq. (15)	$\min_{\mathbf{s}_i^b}\|\mathbf{s}_i^b\|_1,\ \text{s.t.}\ \ \left\|\mathbf{x}_i - X^k\mathbf{s}_i^b\right\| < \varepsilon,\ 1 = \mathbf{1}^T\mathbf{s}_i^b$
Eq. (16)	$W_{ij}^b = \begin{cases} s_{i,j}^b, & \text{if } 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} \\ 0, & \text{if } n_1 + \cdots + n_{k-1} + 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} + n_k \\ s_{i,j-n_k}^b, & \text{if } n_1 + \cdots + n_k + 1 \leqslant j \leqslant n \end{cases}$
Eq. (20)	$\min_\zeta\|\zeta\|_1\ \text{s.t.}\ \ \mathbf{x}_i = X_k\mathbf{s}_i^w + \zeta,\quad 1 = \mathbf{1}^T\mathbf{s}_i^w$
Eq. (21)	$W_{ij}^w = \begin{cases} 0, & 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} \\ s_{i,j-(n_1+\cdots+n_{k-1})}^w, & \text{if } n_1 + \cdots + n_{k-1} + 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} + n_k \\ 0, & \text{if } n_1 + \cdots + n_k + 1 \leqslant j \leqslant n \end{cases}$

class, then, similar to Eq. (12), we can obtain the sparse within-class reconstructive weight vector $\mathbf{s}_i^w$ for each $\mathbf{x}_i$ by minimizing the following object function

$$\min_\zeta\|\zeta\|_1\ \text{s.t.}\quad \mathbf{x}_i = X_k\mathbf{s}_i^w + \zeta,\quad 1 = \mathbf{1}^T\mathbf{s}_i^w \tag{20}$$

where $\mathbf{s}_i^w = \left[s_{i,1}^w, \ldots, s_{i,i-(n_1+\cdots+n_{k-1})-1}^w, 0, s_{i,i-(n_1+\cdots+n_{k-1})+1}^w, \ldots, s_{i,n_k}^w\right]^T$ is an n_k-dimensional vector in which the $(i-(n_1+\cdots+n_{k-1}))$th element is equal zero implying that $\mathbf{x}_i$ is removed from X_k, ζ is the error term. Note we no longer penalize $\mathbf{s}_i^w$ in Eq. (20), since X_k consists of only samples of class k and so $\mathbf{s}_i^w$ is no longer expected to be sparse.

Let W^w denote the within-class weight matrix. Since the coefficient $\mathbf{s}_i^w = \left[s_{i,1}^w, \ldots, s_{i,i-(n_1+\cdots+n_{k-1})-1}^w, 0, s_{i,i-(n_1+\cdots+n_{k-1})+1}^w, \ldots, s_{i,n_k}^w\right]^T$ for the ℓ_1 reconstruction denotes the contribution of each sample which is in the same class as $\mathbf{x}_i$ to reconstruct $\mathbf{x}_i$, then $\mathbf{s}_i^w = \left[s_{i,1}^w, \ldots, s_{i,i-(n_1+\cdots+n_{k-1})-1}^w, 0, s_{i,i-(n_1+\cdots+n_{k-1})+1}^w, \ldots, s_{i,n_k}^w\right]^T$ can reflect the close relation among $\mathbf{x}_i$ and the samples which are in the same class as $\mathbf{x}_i$. So we use the coefficients $s_{i,1}^w, \ldots, s_{i,i-(n_1+\cdots+n_{k-1})-1}^w$, $0, s_{i,i-(n_1+\cdots+n_{k-1})+1}^w, \ldots, s_{i,n_k}^w$ as the within-class graph weights. W_{ij}^w is defined as

$$W_{ij}^w = \begin{cases} 0, & \text{if } 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} \\ s_{i,j-(n_1+\cdots+n_{k-1})}^w, & \text{if } n_1 + \cdots + n_{k-1} + 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} + n_k \\ 0, & \text{if } n_1 + \cdots + n_k + 1 \leqslant j \leqslant n \end{cases} \tag{21}$$

Note that the graph weights among $\mathbf{x}_i$ and the samples which are not in the same class as $\mathbf{x}_i$ are all zeros. In order to preserve the data local geometry, we minimize the following cost function:

$$\sum_{i=1}^n\left\|\mathbf{a}^T\mathbf{x}_i - \sum_{j=1}^n W_{ij}^w\mathbf{a}^T\mathbf{x}_j\right\|^2 \tag{22}$$

From Eq. (22) we can get

$$\begin{aligned}\sum_{i=1}^n\left\|\mathbf{a}^T\mathbf{x}_i - \sum_{j=1}^n W_{ij}^w\mathbf{a}^T\mathbf{x}_j\right\|^2 &= \mathbf{a}^T\left(\sum_{i=1}^n\left\|\mathbf{x}_i - XW_i^w\right\|^2\right)\mathbf{a} \\ &= \mathbf{a}^T\left(\sum_{i=1}^n(\mathbf{x}_i - XW_i^w)(\mathbf{x}_i - XW_i^w)^T\right)\mathbf{a}\end{aligned} \tag{23}$$

where W_i^w is the ith column vector of W^w. Then we can get

$$\begin{aligned}&\mathbf{a}^T\left(\sum_{i=1}^n(\mathbf{x}_i - XW_i^w)(\mathbf{x}_i - XW_i^w)^T\right)\mathbf{a} \\ &= \mathbf{a}^TX\left(\sum_{i=1}^n(\mathbf{e}_i - W_i^w)(\mathbf{e}_i - W_i^w)^T\right)X^T\mathbf{a} \\ &= \mathbf{a}^TX\left(\sum_{i=1}^n\mathbf{e}_i\mathbf{e}_i^T - W_i^w\mathbf{e}_i^T - \mathbf{e}_i(W_i^w)^T + W_i^w(W_i^w)^T\right)X^T\mathbf{a} \\ &= \mathbf{a}^TXM^wX^T\mathbf{a}\end{aligned} \tag{24}$$

where $M^w = (I - W^w)^T(I - W^w)$.

3.3. The objective function and the algorithm of DSNPE

The goal of DSNPE is to maximize Eq. (19) and minimize Eq. (24) at the same time. Motivated by the idea of MMC, the objective function of DSNPE is

$$\arg\max_{\mathbf{a}}\mathbf{a}^TXM^bX^T\mathbf{a} - \mu\mathbf{a}^TXM^wX^T\mathbf{a} \tag{25}$$

where μ is a non-negative constant which balances the two terms of the objective function. If we want to seek r discriminant vectors $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_r]$, then the objective function of DSNPE can be converted into

$$\arg\max_A tr(A^TXM^bX^TA) - \mu tr(A^TXM^wX^TA) \tag{26}$$

Fig. 1. Images of one person in ORL.

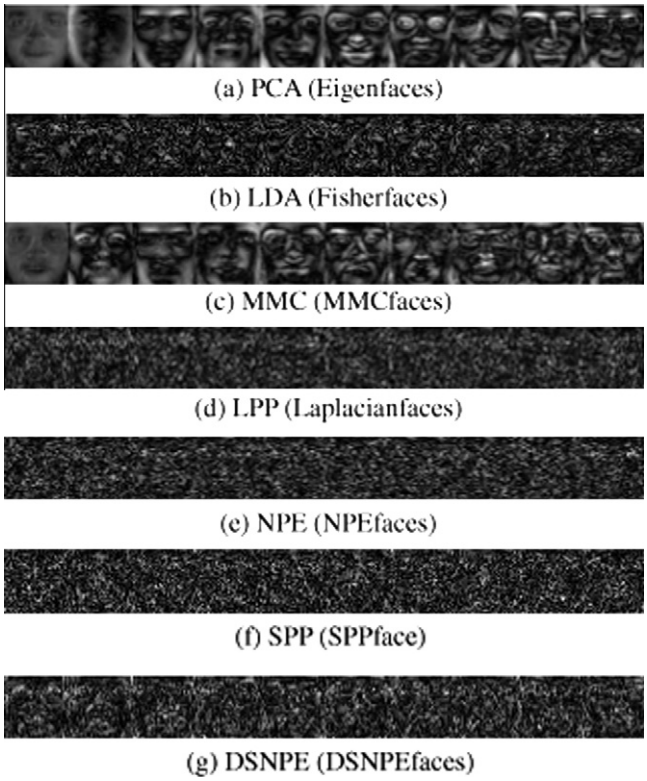

Fig. 2. The first ten basis vectors calculated by (a) PCA (Eigenfaces), (b) LDA (Fisherfaces), (c) MMC (MMCfaces), (d) LPP (Laplacianfaces), (e) NPE (NPEfaces), (f) SPP (SPPfaces), and (g) DSNPE (DSNPEfaces) using the cropped ORL database.

Table 2
Recognition accuracy (%) on ORL (mean ± std).

Sample size	PCA	LDA	MMC	LPP	NPE	SPP	DSNPE
5	86.8 ± 2.3	91.7 ± 3.6	93.5 ± 2.6	92.7 ± 1.8	92.9 ± 1.8	90.3 ± 2.2	**96.0** ± **1.4**
6	89.5 ± 2.5	93.9 ± 2.4	95.0 ± 1.9	94.8 ± 2.2	95.2 ± 2.1	91.8 ± 2.9	**97.2** ± **1.6**

The objective function Eq. (26) for minimizing $tr(A^T X M^w X^T A)$ while maximizing $tr(A^T X M^b X^T A)$ clearly has a direct solution that can be computed using the eigenvalue decomposition method. Note that the matrix M^b and M^w are symmetric. Thus the optimal projection vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_r$ can be selected as the orthonormal eigenvectors corresponding to the first r largest eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_r$, i.e. $(XM^bX^T - \mu XM^wX^T)\mathbf{a}_j = \lambda_j \mathbf{a}_j$, where $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_r$. Different from algorithms, e.g., LDA, LPP, NPE, and SPP, which lead to a generalized eigenvalue problem, DSNPE successfully avoids the matrix singularity problem since it has no inverse operation over a matrix. However, the PCA step is still recommended to reduce noise.

The proposed DSNPE algorithmic procedure can be summarized as follows:

1. Compute the eigendecomposition of the covariance matrix $\Sigma = A_{PCA}\Sigma_{PCA}A_{PCA}^T$, where Σ_{PCA} is the eigenvalue of Σ and the columns of A_{PCA} are the orthogonal eigenvectors of corresponding eigenvalues of Σ. Then the projected training samples can be computed using $A_{PCA}^T X$. We still denote the data set in the PCA subspace by X.
2. Use Eq. (15), i.e. $\min_{s_i^b} \|\mathbf{s}_i^b\|_1$, s.t. $\left\|\mathbf{x}_i - X^k \mathbf{s}_i^b\right\| < \varepsilon, 1 = \mathbf{1}^T \mathbf{s}_i^b$, to compute the optimal sparse between-class reconstructive weight vector $\mathbf{s}_i^b$ and use Eq. (16), i.e.

$$W_{ij}^b = \begin{cases} s_{i,j}^b, & \text{if } 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} \\ 0, & \text{if } n_1 + \cdots + n_{k-1} + 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} + n_k, \\ s_{i,j-n_k}^b, & \text{if } n_1 + \cdots + n_k + 1 \leqslant j \leqslant n \end{cases}$$

to construct the between-class weight matrix W^b.
3. Use Eq. (20), i.e. $\min_\zeta \|\zeta\|_1$ s.t. $\mathbf{x}_i = X_k \mathbf{s}_i^w + \zeta,\ 1 = \mathbf{1}^T \mathbf{s}_i^w$, to compute the optimal sparse within-class reconstructive weight vector $\mathbf{s}_i^w$ and use Eq. (21), i.e.

$$W_{ij}^w = \begin{cases} 0, & \text{if } 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} \\ s_{i,j-(n_1+\cdots+n_{k-1})}^w, & \text{if } n_1 + \cdots + n_{k-1} + 1 \leqslant j \leqslant n_1 + \cdots + n_{k-1} + n_k, \\ 0, & \text{if } n_1 + \cdots + n_k + 1 \leqslant j \leqslant n \end{cases}$$

to construct the within-class weight matrix W^w.
4. Solve the eigenvalue problem $(XM^bX^T - \mu XM^wX^T)\mathbf{a}_j = \lambda_j \mathbf{a}_j$. Let $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_r$ be the r largest eigenvalues of $(XM^bX^T - \mu XM^wX^T)$ and $\mathbf{a}_1, \ldots, \mathbf{a}_r$ be the associated orthonormal eigenvectors.
5. The final projection matrix is $A = A_{PCA}A_{DSNPE}$, where $A_{DSNPE} = [\mathbf{a}_1, \ldots, \mathbf{a}_r]$.

For convenience, we present in Table 1 the equation used in the DSNPE algorithm.

3.4. Computational analysis of the DSNPE algorithm

It takes $O(mn^2)$ to perform PCA on the training samples in the Step 1 of the DSNPE algorithm. The complexity of solving the Eq. (15) using standard linear programming is $4kn^2/3 + kn(n - n_k) + O(k(n - n_k))$ [42,43], where k is the number of iterations used in the standard linear programming technique. Then it takes $n_1(4kn^2/3 + kn(n - n_1) + O(k(n - n_1))) + \cdots + n_c\ (4kn^2/3 + kn(n - n_c) + O(k(n - n_c)))$ to construct the between-class weight matrix W^b in the Step 2 of the DSNPE algorithm. Similarly, it takes $n_1(4kn^2/3 + knn_1 + O(kn_1)) + \cdots + n_c\ (4kn^2/3 + knn_c) + O(kn_c)$ to construct the within-class weight matrix W^w in the Step 3 of the DSNPE algorithm. Step 4 computes the eigendecomposition of a $(n-1) \times (n-1)$ matrix, hence, takes $O(n^3)$. The matrix multiplication in the Step 5 takes $O(mn^2)$.

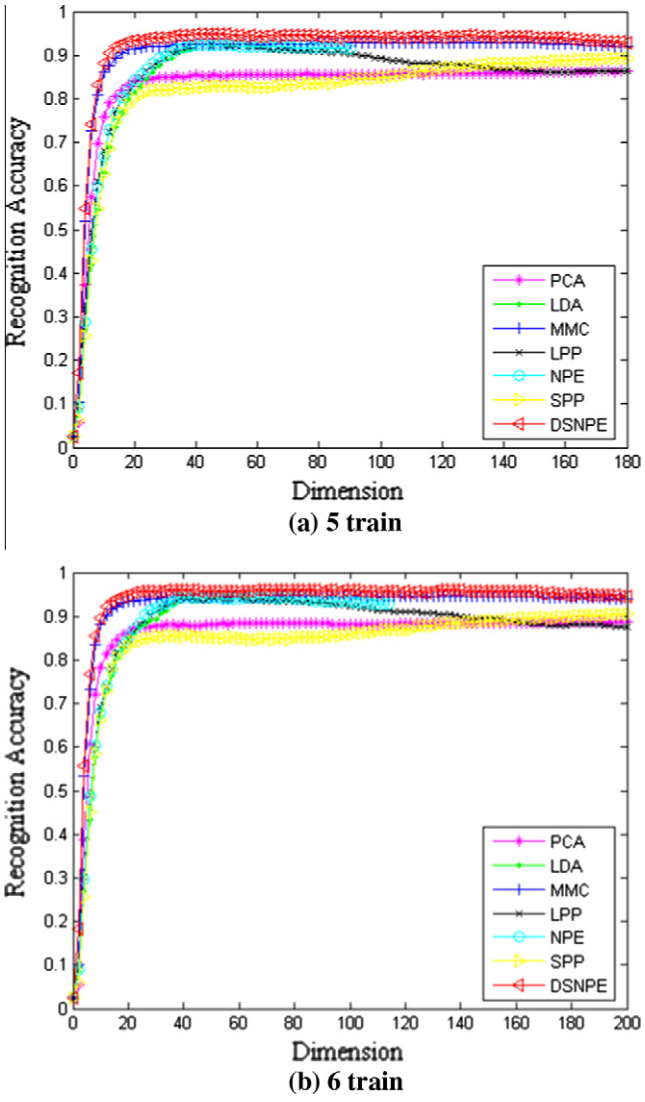

Fig. 3. Recognition rate vs. dimension of reduced space on the ORL database. (a) 5 Train, (b) 6 Train.

Fig. 4. Images of one person in Yale.

4. Experiments

In this section, we will conducts some experiments to evaluate the performances of the proposed DSNPE and some other methods including PCA [11], LDA [12], MMC [14], LPP [29], NPE [30], SPP

Table 3
Recognition accuracy (%) on Yale (mean ± std).

Sample size	PCA	LDA	MMC	LPP	NPE	SPP	DSNPE
5	57.0 ± 3.9	74.6 ± 3.6	73.4 ± 3.2	77.9 ± 2.8	78.2 ± 3.4	60.5 ± 3.6	**81.0** ± **3.7**
6	61.1 ± 6.2	78.3 ± 3.8	77.3 ± 4.1	82.3 ± 3.7	81.5 ± 3.1	66.3 ± 4.2	**85.0** ± **3.0**

[39]. Since PCA, LDA, MMC, LPP, NPE and SPP have been successfully applied to face recognition, we will also evaluate the performances of the proposed DSNPE on face image databases, i.e., ORL, Yale, AR and FERET. To make the comparison fair, for all the evaluated algorithms we first apply PCA as preprocessing step by retaining 100% energy. A nearest neighbor classifier (1-NN) with cosine distance is employed to classify in the projected feature space.

For LPP, the Gaussian Kernel exp $(-\|x-y\|^2/\sigma^2)$ is used and parameter σ is set as $2^{(e-10)/2.5}\delta_0$, $e = 0, 1, \ldots, 20$, where δ_0 is the standard derivation of the training data set. For DSNPE, we empirically set the value of μ as 10.

4.1. Experiments on ORL database

The ORL face database consists of a total of 400 face images, of a total of 40 people (10 samples per person). For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glassed/no glassed). All the images were taken against a dark homogeneous background with the subjects in an upright, front position (with tolerance for some side movement). In our experiments, each image in ORL database was manually cropped and resized to 32×32. Some example images of one person are shown in Fig. 1.

4.1.1. Face representation using DSNPE

The projected subspaces learned by PCA, LDA, MMC, LPP, NPE, SPP, and DSNPE are different. The images in the training set are used to learn such spaces spanned by the eigenvectors of the corresponding algorithms. Using the first five image samples of each person from the ORL database as the training set, we present the first ten basis vectors of different algorithms in Fig. 2.

4.1.2. Comparison of the performance

In this subsection, we compare the performances of different algorithms on the ORL database. We randomly select i(=5,6) samples of each individual for training, and the rest of the ORL database for testing. For each giving i, we perform 20 times to randomly choose the training set and calculate the average recognition rates as well as the standard deviations. Table 2 presents the maximal average recognition and standard deviations for each method. Fig. 3 illustrates the plot of recognition rate vs. the dimension of reduced space for different methods.

4.2. Experiments on Yale database

The Yale face database contains 165 gray scale images of 15 individuals, each individual has 11 images. The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised, and wink). In our experiments, each image in Yale database was manually cropped and resized to 32×32. Fig. 4 shows sample images of one person.

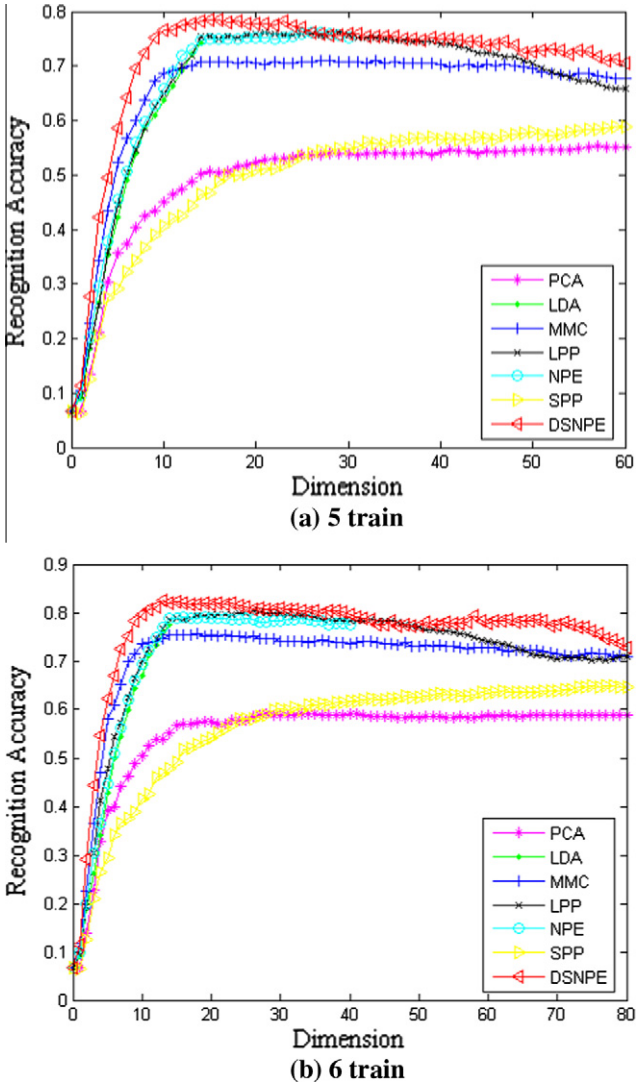

Fig. 5. Recognition rate vs. dimension of reduced space on the Yale database. (a) 5 Train, (b) 6 Train.

Fig. 6. Images of one person in AR.

4.2.1. Comparison of the performance

In this subsection, we compare the performances of different algorithms on the Yale database. We randomly select i(=5,6) samples of each individual for training, and the rest of the Yale database for testing. For each giving i, we perform 20 times to

Table 4
Recognition accuracy (%) on AR (mean ± std).

Sample size	PCA	LDA	MMC	LPP	NPE	SPP	DSNPE
5	54.3 ± 2.2	88.2 ± 1.1	92.9 ± 0.9	88.6 ± 1.0	88.4 ± 0.9	70.0 ± 6.7	**96.1** ± **0.7**

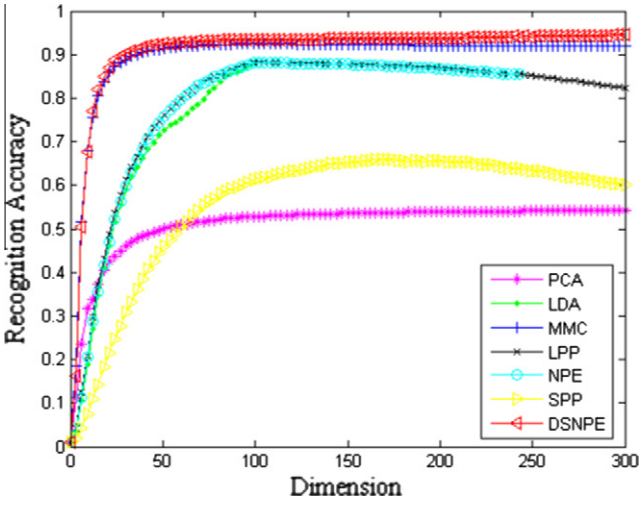

Fig. 7. Recognition rate vs. dimension of reduced space on the AR database.

Fig. 8. Images of one person in FERET.

randomly choose the training set and calculate the average recognition rates as well as the standard deviations. Table 3 presents the maximal average recognitions and standard deviations for each method. Fig. 5 illustrates the plot of recognition rate vs. the dimension of reduced space for different methods.

4.3. Experiments on AR database

The AR face database 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. For each individual, 26 pictures were taken in two sessions (separated by two weeks) and each section 13 color images. In our experiments here, we choose a subset which contains 1400 face images corresponding to 100 persons (50 men and 50 women), each individual has 14 images. The face portion of each image is manually cropped and then normalized to 33×24 pixels. The sample images of one person are shown in Fig. 6.

4.3.1. Comparison of the performance

In this subsection, we compare the performances of different algorithms on the AR database. We randomly select five samples of each individual for training, and the rest of the AR database for testing. We perform 20 times to randomly choose the training set and calculate the average recognition rates as well as the standard deviations. Table 4 presents the maximal average recognitions and standard deviations for each method. Fig. 7 illustrates the plot of recognition rate vs. the dimension of reduced space for different methods.

4.4. Experiments on FERET database

The FERET face database contains 14,126 images from 1199 individuals. In our experiments, we select a subset which contains

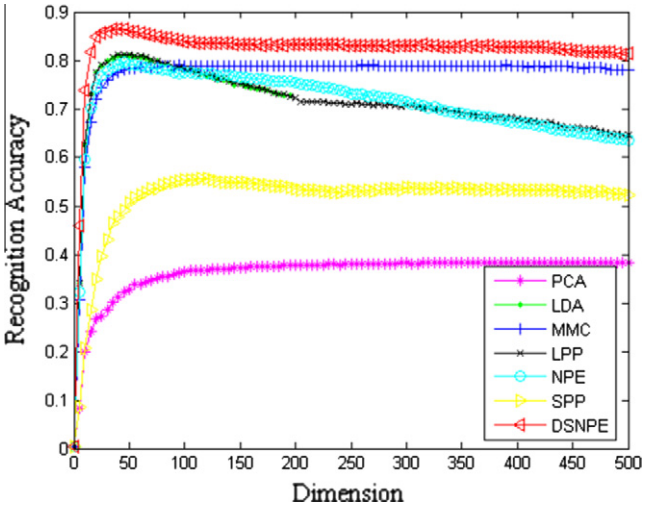

Fig. 9. Recognition rate vs. dimension of reduced space on the FERET database.

Table 6
Recognition accuracy (%) on AR database based on different classifiers using DSNPE.

Classifiers	Sunglass	Scarf
1-NN	70.0	20.5
SRC	87.0	59.5

1400 images of 200 individuals (each individual has seven images). The subset involves variations in facial expression, illumination and pose. In our experiments, each image in FERET database was manually cropped and resized to 40×40. The sample images of one person are shown in Fig. 8.

4.4.1. Comparison of the performance

We randomly select five samples of each individual for training, and the rest of the FERET database for testing. We perform 20 times to randomly choose the training set and calculate the average recognition rates as well as the standard deviations. Table 5 presents the maximal average recognitions and standard deviations for each method. Fig. 9 illustrates the plot of recognition rate vs. the dimension of reduced space for different methods.

4.5. Experiments based on sparse representation classifier

In this subsection, we further evaluate our proposed DSNPE based on sparse representation classifier (SRC) [37]. Firstly, we conduct an experiment on AR databases. In order to make the comparison fair, we partition the AR database as in [37]. A subset from the AR database consisting of 1400 images from 100 subjects, 50 male and 50 female, is used here. 800 images (about 8 samples per subject) of non-occluded frontal views with various facial expressions are used for training, while the others with sunglasses and scares are used for testing. The images are resized to 50×40. In this experiment, our proposed DSNPE are used for feature extraction. Then the nearest neighbor classifier (1-NN) with cosine distance and SRC are respectively employed to classify in the projected feature space. The results are shown in Table 6.

Secondly, we conduct an experiment on Yale database. The images are resized to 32×32. In the experiment, we use the first six images per class for training and the remaining images for

Table 5
Recognition accuracy (%) on FERET (mean ± std).

Sample size	PCA	LDA	MMC	LPP	NPE	SPP	DSNPE
5	38.8 ± 2.0	82.0 ± 1.4	79.8 ± 1.4	82.1 ± 1.7	80.5 ± 1.9	56.9 ± 1.5	**87.1 ± 1.0**

Table 7
Recognition accuracy (%) on Yale.

Classifiers	PCA	LDA	MMC	LPP	NPE	SPP	DSNPE	Gabor + ELDA	Gabor + DSNPE
1-NN	62.2	79.3	78.5	83.4	82.6	67.5	85.7	90.4	93.5
SRC	82.9	82.7	82.8	83.5	83.2	83.4	86.0	91.2	95.1

testing. Besides PCA, LDA, MMC, LPP, NPE, SPP, and DSNPE, we also evaluate the performances of two Gabor-based methods, i.e., enhanced LDA based on Gabor (Gabor + ELDA) [23] and DSNPE based on Gabor (Gabor + DSNPE), on Yale database. The parameters of Gabor filter is the same as in [23]. The results are shown in Table 7.

4.6. Discussion

From the experiments above, we notice that:

(1) The proposed DSNPE consistently outperforms PCA, LDA, MMC, LPP, NPE, and SPP in all experiments in four face databases, namely ORL, Yale, AR and FERET. The reason may be that DSNPE is a supervised method which considers not only the intraclass geometry but also discriminative information derived from the interclass samples. Moreover, DSNPE can get orthonormal projection vectors.
(2) The recognition rates significantly change on ORL, Yale, AR and FERET databases. The recognition rates on ORL and AR databases are much higher than that on Yale and FERET databases. The reason may be that Yale and FERET databases contain much more variations of pose, illumination and expression than ORL and AR databases.
(3) From Table 3 we can see that DSNPE reaches 96.1% on the AR database (without occlusion). In [37] similar results of 95.7% were reached and better results of 99.1% were reached in [20].
(4) PCA is simple to perform, but it generally performs much worse than other methods based on the nearest neighbor classifier. The performance of SPP is better than that of PCA in our experiments since SPP tends to include potential discriminant information through sparse representation.
(5) As presented in Tables 5 and 6, the classifiers also affect the recognition performance. The nearest neighbor classifier is used in our experiments due to its simplicity. If more efficient methods such as neural networks, GMM, SVM and SRC are used, the recognition accuracies will improve. For AR database, the DSNPE method reaches the same results as [37] for face recognition with occlusions with sunglasses 87% and with scarves 59.5% when the SRC method is used. Nevertheless, in [37], the SRC method was applied to subimages and then combined results improved to 97.5% with sunglasses and to 93.5% with scarves. However, SRC is very time-consuming since it uses sparse reconstruction for each test sample. On the contrary, in DSNPE, the sparse reconstruction is involved only in the training step. Once the projection vectors are obtained, they can be used for both the training and test data and then the efficiency of recognition can be effectively improved.
(6) Gabor based methods, i.e., Gabor + ELDA and Gabor + DSNPE, can get higher recognition performances than other methods. For AR database, the Gabor based methods in [20] results yields 98% with sunglasses and 99% with scarves. Additionally, in [44] results reached 80% with sunglasses and 98 with scarves. For FERET database, the DSNPE method reaches 87.1%. However, several other local feature based methods have reached higher performances over 90 [20,45–48]. Additionally, local feature based methods need only one face image for enrollment which is desired in many real situations where it is difficult to have available several images from each person.

5. Conclusions and future work

In this paper, based on sparse representation, a new algorithm called discriminant sparsity neighborhood preserving embedding is proposed for supervised dimensionality reduction. DSNPE constructs graph and corresponding edge weights simultaneously through sparse representation. Moreover, DSNPE explicitly takes into account the within-neighboring information and between-neighboring information. Experimental results on ORL, Yale, AR and FERET face databases indicate that DSNPE performs significantly better than PCA, LDA, MMC, LPP, NPE, and SPP in terms of recognition accuracy.

In this paper, we only conduct our experiments on face images. Since sparse representation has been applied to other pattern recognition problem, e.g. gene expression data [49] and handwritten numeral [50], face recognition is only one of the potential applications of our proposed method and it is possible to process these data sets by using our proposed DSNPE and we will conduct some experiments on these data sets in our future work.

Acknowledgments

This research is supported by NSFC of China (Nos. 61005008, 60873151, 60973098) and the 2010 Graduates' Research Innovation Program of Higher Education of Jiangsu Province (No. CX10B_131Z). The authors thank Cai et al. for providing the codes of LPP and NPE on their homepage, and Qiao et al. for providing the codes of SPP on their homepage.

References

[1] P. Bermejo, L.D.L. Ossa, J.A. Gámez, J.M. Puerta, Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking, Knowledge-Based Systems 25 (1) (2012) 35–44.
[2] I.-O. Stathopoulou, E. Alepis, G.A. Tsihrintzis, M. Virvou, On assisting a visual-facial affect recognition system with keyboard-stroke pattern information, Knowledge-Based Systems 23 (4) (2010) 350–356.
[3] K.-A. Hwang, C.-H. Yang, Assessment of affective state in distance learning based on image detection by using fuzzy fusion, Knowledge-Based Systems 22 (4) (2009) 256–260.
[4] R. Chellappa, C. Wilson, S. Sirohey, Human and machine recognition of faces: a survey, Proceedings of the IEEE 83 (5) (1995) 705–740.
[5] W. Zhao, R. Chellappa, P. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Computing Surveys 35 (4) (2003) 399–458.
[6] M.E. Elalami, A novel image retrieval model based on the most relevant features, Knowledge-Based Systems 24 (1) (2011) 23–32.
[7] S.S. Kannan, N. Ramaraj, A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm, Knowledge-Based Systems 23 (6) (2010) 580–585.
[8] Y. Liu, Dimensionality reduction and main component extraction of mass spectrometry cancer data, Knowledge-Based Systems 26 (2012) 207–215.
[9] D. Soria, J.M. Garibaldi, F. Ambrogi, E.M. Biganzoli, I.O. Ellis, A 'non-parametric' version of the naive Bayes classifier, Knowledge-Based Systems 24 (2011) 775–784.
[10] Y. Gao, J. Pan, G. Ji, Z. Yang, A novel two-level nearest neighbor classification algorithm using an adaptive distance metric, Knowledge-Based Systems 26 (2012) 103–110.
[11] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd ed., Academic Press, Boston, USA, 1990.
[12] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., John Wiley & Sons, New York, 2000.

[13] S.J. Raudys, A.K. Jain, Small sample size effects in statistical pattern recognition: recommendations for practitioners, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (3) (1991) 252–264.
[14] H. Li, T. Jiang, K. Zhang, Efficient and robust feature extraction by maximum margin criterion, IEEE Transactions on Neural Networks 17 (1) (2006) 1157–1165.
[15] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[16] K.R. Müler, S. Mika, G. Räsch, K. Tsuda, B. Schökopf, An introduction to kernel-based learning algorithms, IEEE Transactions on Neural Networks 12 (3) (2001) 181–201.
[17] B. Schölkopf, A. Smola, K.R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Compution 10 (5) (1998) 1299–1319.
[18] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, A. Smola, K.-R. Muller, Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 623–628.
[19] J. Yanxia, R. Bo, Face recognition using local Gabor phase characteristics, in: 2010 International Conference on Computational Intelligence and Software Engineering (CiSE), 2010, pp. 1–4.
[20] C.A. Perez, L.A. Cament, L.E. Castillo, Methodological improvement on local Gabor face recognition based on feature selection and enhanced Borda count, Pattern Recognition 44 (2011) 951–963.
[21] L. Yu, Z. He, Q. Cao, Gabor texture representation method for face recognition using the Gamma and generalized Gaussian models, Image and Vision Computing 28 (1) (2010) 177–187.
[22] S. Xie, S. Shan, X. Chen, J. Chen, Fusing local patterns of Gabor magnitude and phase for face recognition, IEEE Transactions on Image Processing 19 (5) (2010) 1349–1361.
[23] C. Liu, H. Wechsler, Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition, IEEE Transactions on Image Processing 11 (4) (2002) 467–476.
[24] S.T. Roweis, L.K. Saul, Nonlinear dimension reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[25] J.B. Tenenbaum, V.d. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[26] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (6) (2003) 1373–1396.
[27] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimension reduction via local tangent space alignment, SIAM Journal on Scientific Computing 26 (1) (2005) 313–338.
[28] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Roux, M. Ouimet, Out-of-sample extensions for LLE, ISOMAP, MDS, Eigenmaps,and spectral clustering, in: Advances in Neural Information Processing Systems (NIPS), MIT Press, MA, Cambridge, 2004.
[29] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using Laplacian faces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.
[30] X. He, D. Cai, S. Yan, H. Zhang, Neighborhood preserving embedding, in: Proceedings in International Conference on Computer Vision (ICCV), 2005, pp. 1208–1213.
[31] D. Cai, X. He, J. Han, Isometric projection, in: Proceedings of AAAI Conference on Artificial Intelligence, 2007.
[32] T. Zhang, J. Yang, D. Zhao, X. Ge, Linear local tangent space alignment and application to face recognition, Neurocomputing 70 (2007) 1547–1553.
[33] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 40–51.
[34] P.Y. Han, A.T.B. Jin, F.S. Abas, Neighbourhood preserving discriminant embedding in face recognition, Journal of Visual Communication and Image Representation 20 (2009) 532–542.
[35] J. Hu, W. Deng, J. Guo, W. Xu, Learning a locality discriminating projection for classification, Knowledge-Based Systems 22 (8) (2009) 562–568.
[36] W. Liu, S.-F. Chang, Robust multi-class transductive learning with graphs, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[37] J. Wright, A. Yang, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.
[38] K. Huang, S. Aviyente, Sparse representation for signal classification, in: Advances in Neural Information Processing Systems (NIPS), 2006.
[39] L. Qiao, Songcan Chen, X. Tan, Sparsity preserving projections with applications to face recognition, Pattern Recognition 43 (1) (2010) 331–341.
[40] B. Cheng, J. Yang, S. Yan, Y. Fu, T.S. Huang, Learning with L1 graph for image analysis, IEEE Transactions on Image Processing 18 (4) (2010) 858–866.
[41] D. Donoho, For most large underdetermined systems of linear equations the minimal L1-norm solution is also the sparsest solution, Communications on Pure and Applied Mathematics 59 (7) (2004) 787–829.
[42] E. Candes, J. Romberg, L1-magic: recovery of sparse signals via convex programming, 2005, <http://www.acm.caltech.edu/l1magic>.
[43] D.L. Donoho, Y. Tsaig, Fast solution of L1-norm minimization problems when the solution may be sparse, IEEE Transactions on Information Theory 54 (11) (2008) 4789–4812.
[44] W. Zhang, S. Shan, W. Gao, X. Chen, H. Zhang, Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition, in: Proceedings of the Tenth IEEE International Conference on Computer Vision, Beijing, China, 2005, pp. 786–791.
[45] B. Zhang, S. Shan, X. Chen, W. Gao, Histogram of Gabor phase patterns (HGPP): a novel object representation approach for face recognition, IEEE Transactions on Image Processing 16 (1) (2007) 57–68.
[46] H.V. Nguyen, L. Bai, L. Shen, Local Gabor binary pattern whitened PCA: a novel approach for face recognition from single image per person, in: Proceedings of the Third International Conference on Advances in Biometrics, Lecture Notes in Computer Science, Alghero, Italy, 2009, pp. 269–278.
[47] J. Zou, Q. Ji, G. Nagy, A comparative study of local matching approach for face recognition, IEEE Transactions on Image Processing 16 (10) (2007) 2617–2628.
[48] S. Xie, S. Shan, X. Chen, X. Meng, W. Gao, Learned local Gabor patterns for face representation and recognition, Signal Processing 89 (12) (2009) 2333–2344.
[49] X. Hang, F.-X. Wu, Sparse representation for classification of tumors using gene expression data, Journal of Biomedicine and Biotechnology 2009 (1) (2009) 1–6.
[50] J. Yang, L. Zhang, Y. Xu, J.-y. Yang, Beyong sparsity: the role of L1-optimizer in pattern classification, Pattern Recognition 45 (3) (2012) 1104–1108.