marginFace: A novel face recognition method by average neighborhood margin maximization

Fei Wang[a,*], Xin Wang[b], Daoqiang Zhang[c], Changshui Zhang[a], Tao Li[b]

[a]State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, China
[b]School of Computing and Information Sciences, Florida International University, FL 33174, USA
[c]Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

ARTICLE INFO

ABSTRACT

We propose a novel appearance-based face recognition method called the *marginFace* approach. By using *average neighborhood margin maximization* (ANMM), the face images are mapped into a face subspace for analysis. Different from *principal component analysis* (PCA) and *linear discriminant analysis* (LDA) which effectively see only the global Euclidean structure of face space, ANMM aims at discriminating face images of different people based on local information. More concretely, for each face image, it pulls the neighboring images of the same person towards it as near as possible, while simultaneously pushing the neighboring images of different people away from it as far as possible. Moreover, we propose an automatic approach for determining the optimal dimensionality of the embedded subspace. The *kernelized* (*nonlinear*) and *tensorized* (*multilinear*) form of ANMM are also derived in this paper. Finally the experimental results of applying *marginFace* to face recognition are presented to show the effectiveness of our method.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Recently, the *appearance based* face recognition methods have aroused considerable interests in image processing and computer vision fields [17,24]. Generally, these approaches treat each face image of size $n_1 \times n_2$ as a vector in $\mathbb{R}^{n_1 \times n_2}$, which brings many problems for practical face recognition, e.g. (1) the curse of high dimensionality is usually a major cause of limitations of many practical technologies; (2) the large quantities of features may even degrade the performances of the classifiers when the size of the training set is small compared to the number of features [10]. A common way to solve this problem is to apply dimensionality reduction methods, among which *principal component analysis* (PCA) [11] and *linear discriminant analysis* (LDA) are two of the most popular ones [6].

PCA is a popular unsupervised method which aims at extracting a subspace in which the variance of the projected data is maximized (or, equivalently, the reconstruction error is minimized). For linearly embedded manifolds, PCA is guaranteed to discover the intrinsic dimensionality of the manifold and produces a compact representation. The famous *Eigenface* method [24] is just based on PCA, which uses a set of basis functions obtained by PCA to describe face images.

However, PCA does not take the class information into account and thus may not be reliable for classification tasks.

LDA is a supervised technique which has been shown to be more effective than PCA in many applications. It aims to maximize the between-class scatter and simultaneously minimize the within-class scatter. Unfortunately, it has also been pointed out that there are still some drawbacks existed in LDA [6], such as (1) it usually suffers from the *small sample size* (SSS) problem [1] which makes the within-class scatter matrix singular; (2) it is only optimal for the case where the distribution of the data in each class is a *Gaussian* with an identical covariance matrix; (3) LDA can only extract at most $c - 1$ features (where c is the number of different classes), which is suboptimal for many applications.

Another limitation of PCA and LDA is that they effectively see only the linear global Euclidean structure. However, some recent research shows that the face images may reside on a nonlinear submanifold [21,18], which makes PCA and LDA inefficient. One way to solve this problem is to apply the *kernel based techniques* [19] to develop the nonlinear manifold learning forms of those methods [20,14]. The other is to adopt some nonlinear methods (which are usually based on local analysis of the data sets) directly to approximate the intrinsic face manifold [18,9,4].

Finally, PCA and LDA take their inputs as vectorial data, but in many real-world vision problems, the data are more naturally represented as higher-order tensors. For example, a captured image is

---

* Corresponding author. Tel.: +86 10 62796872.
  E-mail address: feiwang03@gmail.com (F. Wang).

a second-order tensor, i.e. matrix, and the sequential data, such as a video sequence for event analysis, is in the form of third-order tensor. Thus it is necessary to derive the *multilinear* forms of these traditional linear feature extraction methods to handle the data as tensors directly. Recently this research topic has received a lot of attention from the image processing and computer vision community [3,29,23], and the proposed methods have been shown to be much more efficient than the traditional vectorial methods.

In this paper, we propose a novel supervised face recognition method called *marginFace* based on a feature extraction method *average neighborhood margin maximization* (ANMM). The goal of ANMM is to find a subspace such that for each face image, it can *pull* the neighboring faces of the same person towards it as near as possible, while simultaneously *push* the neighboring faces of different persons away from it as far as possible. In such a way, each face image in the original image space is mapped into a discriminative low-dimensional face subspace, which is characterized by a set of feature images, called *marginFaces*. We also derive the *kernelized* (*nonlinear*) and *tensorized* (*multilinear*) forms of the *marginFace* method in this paper. Finally the experimental results on face recognition are presented to show the effectiveness of our method.

It is worthwhile to highlight some aspects of the *marginFace* algorithm as follows:

(1) While the goal of the *Eigenface* method is to preserve the global structure of the image space, and the goal of the *Fisherface* method is to preserve the global discriminative information, our *marginFace* method aims to explore the discriminative information locally, which is usually more important and effective for face recognition tasks.
(2) Compared to the traditional *Fisherface* method, *marginFace* can (1) find the discriminant directions without assuming the particular form of class densities, (2) avoid the *small sample size* problem as there is no matrix inversion computations involved, (3) find much more features, which is not limited to $C - 1$ as in traditional LDA (C is the number of classes).
(3) *marginFace* can determine the optimal dimensionality of the projected space automatically. To the best of our knowledge, most of the traditional subspace learning methods have to determine the dimension of the projected space by either cross-validation or exhaustive search.

The rest of this paper is organized as follows. In Section 2 we will introduce our ANMM method in detail, and its kernelized and tensorized forms will be derived in Sections 3 and 4. The experimental results on applying *marginFace* method in face recognition will be presented in Section 5. In Section 6 we will compare our method with some related approaches, followed by the conclusions and discussions in Section 7.

2. Feature extraction by average neighborhood margin maximization

In this section we will introduce our *average neighborhood margin maximization* algorithm in detail. Like other linear feature extraction methods, ANMM aims to learn a projection matrix $\mathbf{W}$ such that the data in the projected space have high within-class similarity and between-class separability. First let us introduce some notations and preliminary definitions.

2.1. Preliminaries

Throughout the paper, we will use the bold lowercase characters, e.g. $\mathbf{x}_i$, to represent the data vectors, and the italic uppercase characters, e.g. X_i, to represent the data tensors. To introduce our ANMM

algorithm, we first need to define two types of neighborhoods for each data point.

Definition 1 (*Homogeneous neighborhood*). For a data point $\mathbf{x}_i$, its ξ *nearest homogeneous neighborhood* $\mathcal{N}_i^o$ is the set of ξ most similar[1] data which are in the same class with $\mathbf{x}_i$.

Definition 2 (*Heterogeneous neighborhood*). For a data point $\mathbf{x}_i$, its ζ *nearest heterogeneous neighborhood* $\mathcal{N}_i^e$ is the set of ζ most similar data which are not in the same class with $\mathbf{x}_i$.

Based on Definitions 1 and 2, we can define the average neighborhood margin as follows.

Definition 3 (*Average neighborhood margin*). The *average neighborhood margin* γ_i for $\mathbf{x}_i$ is defined as

$$\gamma_i = \sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{\|\mathbf{x}_i - \mathbf{x}_k\|^2}{|\mathcal{N}_i^e|} - \sum_{\substack{j:\mathbf{x}_j \in \mathcal{N}_i^o \vee \\ \mathbf{x}_i \in \mathcal{N}_j^o}} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{|\mathcal{N}_i^o|},$$

where $|\cdot|$ represents the cardinality of a set.

Literally, this margin measures the difference between the average distance from $\mathbf{x}_i$ to the data points in its heterogeneous neighborhood and the average distance from it to the data points in its homogeneous neighborhood. Then the total average neighborhood margin for the whole data set is defined to be

Definition 4 (*Total average neighborhood margin*). The *total average neighborhood margin* γ for the whole data set $\mathcal{X}$ is defined as

$$\gamma = \sum_i \gamma_i$$

$$= \sum_i \left(\sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{\|\mathbf{x}_i - \mathbf{x}_k\|^2}{|\mathcal{N}_i^e|} - \sum_{j:\mathbf{x}_j \in \mathcal{N}_i^o} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{|\mathcal{N}_i^o|} \right).$$

2.2. The algorithm

As we have stated before, the goal of ANMM is to seek for a projection matrix $\mathbf{W} \in \mathbb{R}^{d \times l}$ (where d is the dimensionality of the original space, l is the dimensionality of the projected space, and usually $l \ll d$), which can project the data points into a low dimensional space such that the data from different classes can be well separated. Most of the previous researches, e.g. LDA and its variants, wanted to find a good subspace in which the different class mass can be separated in a global way. According to [26], in general it might be hard to find a subspace which has a good separability for the whole data set, which motivates us to consider local methods, since empirically the local methods may have stronger discriminative power than global methods [2,25].

Recalling that the average neighborhood margin in Definition 3 just reflects the local separability around each data point under the current distribution. The maximization of such a margin in the projected space can *push* the data points whose labels are different from $\mathbf{x}_i$ away from $\mathbf{x}_i$ while *pull* the data points having the same class label with $\mathbf{x}_i$ towards $\mathbf{x}_i$. Fig. 1 gives us an intuitive illustration of the result of maximizing the average neighborhood margin. Therefore, for the whole data set, we just need to maximize the *total average*

---

[1] In this paper two data vectors are considered to be similar if the Euclidean distance between them is small, two data tensors are considered to be similar if the Frobenius norm of their difference tensor is small.
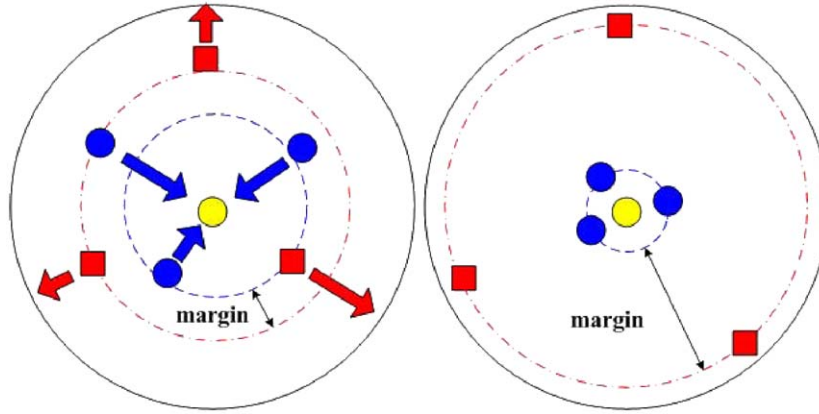
Fig. 1. An intuitive illustration of the ANMM criterion. The yellow disk in the center represents $\mathbf{x}_i$. The blue disks are the data points in the homogeneous neighborhood of $\mathbf{x}_i$, and the red squares are the data points in the heterogeneous neighborhood of $\mathbf{x}_i$. (a) shows the data distribution in the original space, (b) shows the data distribution in the projected space. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

neighborhood margin γ in Definition 4, which can maximize the over-all separability of the whole data set, and this becomes the basic idea of ANMM.

Let $\mathbf{y}_i = \mathbf{W}^T\mathbf{x}_i \in \mathbb{R}^l$ be the image of $\mathbf{x}_i$ in the projected space. Then

$$\sum_i \sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{\|\mathbf{y}_i - \mathbf{y}_k\|^2}{|\mathcal{N}_i^e|}$$

$$= \mathrm{tr}\left(\sum_i \sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{(\mathbf{y}_i - \mathbf{y}_k)(\mathbf{y}_i - \mathbf{y}_k)^T}{|\mathcal{N}_i^e|}\right)$$

$$= \mathrm{tr}\left[\mathbf{W}^T\left(\sum_i \sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{(\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_k)^T}{|\mathcal{N}_i^e|}\right)\mathbf{W}\right]$$

$$= \mathbf{W}^T\,\mathrm{tr}(\mathbf{S})\mathbf{W}, \tag{1}$$

where the matrix

$$\mathbf{S} = \sum_{\substack{i,k:\\\mathbf{x}_k \in \mathcal{N}_i^e}} \frac{(\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_k)^T}{|\mathcal{N}_i^e|} \tag{2}$$

is called the *scatterness matrix*. Similarly, we can define the *compactness matrix* as

$$\mathbf{C} = \sum_{\substack{i,j:\\\mathbf{x}_j \in \mathcal{N}_i^o}} \frac{(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T}{|\mathcal{N}_i^o|}. \tag{3}$$

Then

$$\sum_i \sum_{j:\mathbf{x}_j \in \mathcal{N}_i^o} \frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{|\mathcal{N}_i^o|} = \mathrm{tr}(\mathbf{W}^T\mathbf{C}\mathbf{W}).$$

Therefore the total average neighborhood margin of the whole data set in the projected space can be rewritten as

$$\gamma = \mathrm{tr}[\mathbf{W}^T(\mathbf{S} - \mathbf{C})\mathbf{W}]. \tag{4}$$

If we expand $\mathbf{W}$ as $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_l)$, then

$$\gamma = \sum_{k=1}^l \mathbf{w}_k^T(\mathbf{S} - \mathbf{C})\mathbf{w}_k.$$

To eliminate the freedom that we can multiply $\mathbf{W}$ with some nonzero scalar, we add the constraint

$$\mathbf{w}_k^T\mathbf{w}_k = 1,$$

i.e., we restrict $\mathbf{W}$ to be constituted of unit vectors. Thus the goal of the ANMM algorithm is just to solve the following optimization problem:

$$\max \quad \sum_{k=1}^l \mathbf{w}_k^T(\mathbf{S} - \mathbf{C})\mathbf{w}_k$$

$$\mathrm{s.t.} \quad \mathbf{w}_k^T\mathbf{w}_k = 1. \tag{5}$$

Using the *Lagrangian* method, we can easily find that the optimal $\mathbf{W}$ is composed of the l eigenvectors corresponding to the largest l eigenvalues of $\mathbf{S} - \mathbf{C}$.

2.3. Determining the optimal projection dimensionality

In the last subsection we have formulated our ANMM algorithm and show that the optimal projection matrix $\mathbf{W}$ can be obtained by eigenvalue decomposition on matrix $\mathbf{S} - \mathbf{C}$. The problem remaining is how to determine an optimal dimensionality, i.e., l, for the projected space. To achieve such a goal, we first introduce the *Ky Fan theorem*.

Theorem 1 (*Ky Fan*). *Let* $\mathbf{H}$ *be a symmetric matrix with eigenvalues*

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n,$$

and the corresponding eigenvectors $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$. *Then*

$$\sum_{i=1}^K \lambda_i = \max_{\mathbf{X}^T\mathbf{X} = \mathbf{I}_K} \mathrm{tr}(\mathbf{X}^T\mathbf{H}\mathbf{X}).$$

Based on the Ky Fan theorem, the maximum value for the total average neighborhood margin defined in Eq. (4) under the constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ is

$$\max_{\mathbf{W}^T\mathbf{W} = \mathbf{I}} \gamma = \max_{\mathbf{W}^T\mathbf{W} = \mathbf{I}} \mathrm{tr}(\mathbf{W}^T(\mathbf{S} - \mathbf{C})\mathbf{W}) = \sum_{i=1}^{l^*} \lambda_i,$$

where

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{l^*} \geq 0 \geq \lambda_{l^*+1} \geq \lambda_n$$

are the eigenvalues of the matrix $\mathbf{S} - \mathbf{C}$, i.e., the optimal dimensionality of the projected space just corresponds to the number of nonnegative eigenvalues of the matrix $\mathbf{S} - \mathbf{C}$.

To summarize, the main procedure of ANMM is shown in Table 1.

Table 1
Average neighborhood margin maximization.

Input: Training set $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, Testing set $\mathscr{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M\}$, Neighborhood size $|\mathcal{N}^o|$, $|\mathcal{N}^e|$;
Output: $l \times M$ feature matrix $\mathbf{F}$ extracted from $\mathscr{Z}$.
1. Construct the *heterogeneous neighborhood* and *homogeneous neighborhood* for each $\mathbf{x}_i$;
2. Construct the *scatterness matrix* $\mathbf{S}$ and *compactness matrix* $\mathbf{C}$ using Eq. (2) and Eq. (3) respectively;
3. Do eigenvalue decomposition on $\mathbf{S} - \mathbf{C}$, construct $d \times l$ matrix $\mathbf{W}$ whose columns are composed by the
eigenvectors of $\mathbf{S} - \mathbf{C}$ corresponding to its largest l
eigenvalues, where l is equal to the number of positive eigenvalues of $\mathbf{S} - \mathbf{C}$;
4. Output $\mathbf{F} = \mathbf{W}^T \mathbf{Z}$ with $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N]$.
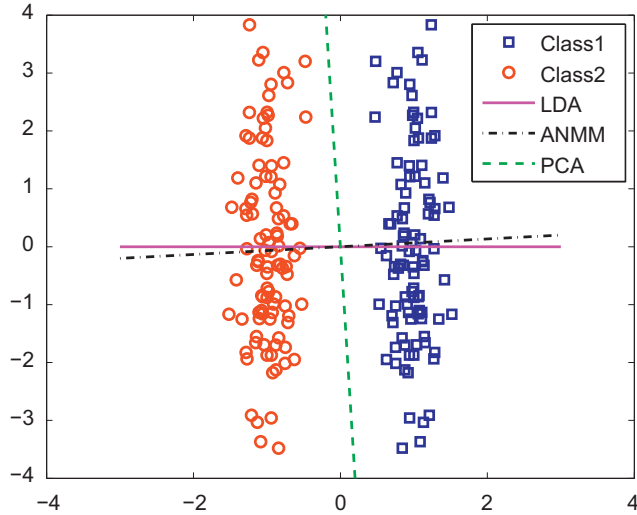

Fig. 2. Two well-separated classes of data points, in which we plot the direction which corresponds to the largest eigenvalue of the decomposed matrices for PCA, LDA, and ANMM. We can see that LDA, ANMM can find the most discriminative direction. For ANMM, the eigenvalues of the matrix $\mathbf{S} - \mathbf{C}$ is $\lambda_1 = 4001.21, \lambda_2 = 18.05$.

2.4. A distance metric learning perspective

Once the optimal projection matrix $\mathbf{W} \in \mathbb{R}^{d \times l}$ is determined, the Euclidean distance between any pair of data points in the projected space becomes

$$
\begin{aligned}
d(\mathbf{y}_i, \mathbf{y}_j) &= \|\mathbf{W}^T \mathbf{x}_i - \mathbf{W}^T \mathbf{x}_j\|^2 \\
&= (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W} \mathbf{W}^T (\mathbf{x}_i - \mathbf{x}_j) \\
&= (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) \\
&= \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{M}}^2
\end{aligned} \tag{6}
$$

which is just a *Mahalanobis distance* parameterized by $\mathbf{M} = \mathbf{W}\mathbf{W}^T \in \mathbb{R}^{d \times d}$ in the original space. Clearly, such a distance metric is low-rank since $l \ll d$, and the optimal rank can be determined by the spectral analysis of $\mathbf{S} - \mathbf{C}$ as we have introduced in the last subsection.

In fact, on each projected direction $\mathbf{w}_i$, the total average neighborhood margin is

$$
\gamma_{\mathbf{w}_i} = \mathbf{w}_i^T (\mathbf{S} - \mathbf{C}) \mathbf{w}_i = \mathbf{w}_i^T \lambda_i \mathbf{w}_i = \lambda_i, \tag{7}
$$

where λ_i is the eigenvalue corresponding to $\mathbf{w}_i$. This equation tells us that what the eigenvalues of $\mathbf{S} - \mathbf{C}$ measures are the separabilities of the data points on the corresponding directions. The larger the λ_i, the better separability of its corresponding subspace. A negative λ_i indicates a poor separability of the projected space, in which the
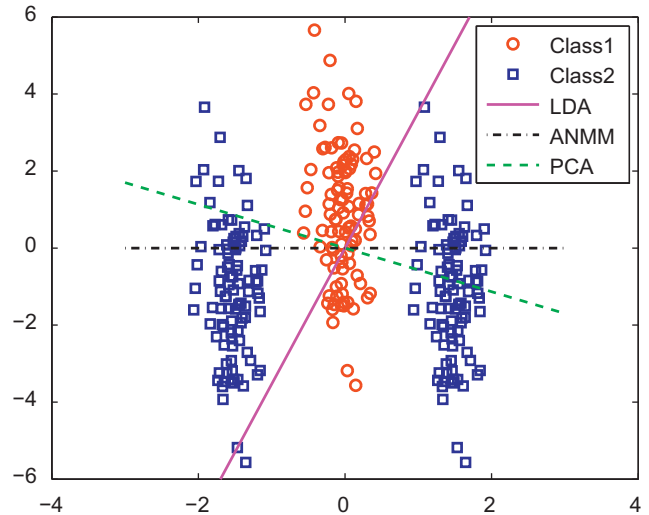

Fig. 3. An example of the multi-modal data set, in which one class is distributed as two separated Gaussians, the other class is distributed as one Gaussian. We can see that LDA is confused in this case, while the ANMM method can still find the direction of the strongest discriminative power. The eigenvalues for $\mathbf{S} - \mathbf{C}$ in this case is $\lambda_1 = 7824.06, \lambda_2 = 1933.76$.

data from different classes cannot be discriminated. Therefore, the method that we use to construct $\mathbf{W}$ in Section 2.3 is just to select the l^* projected directions with the best separabilities.

Figs. 2 and 3 show us two toy examples for comparing ANMM with the traditional PCA and LDA methods, from which we can clearly see that (1) PCA only aims to find the direction in which the data structure is maximally preserved, and there may not exist any discrimination on such direction; (2) LDA can find the best discriminative direction when the data from each class are distributed as Gaussians with equivalent covariance matrices, but it may confuse when the data distribution is more complicated; (3) ANMM can find the discriminative directions based on local analysis, and it does not make any assumptions on the distributions of the data points. Moreover, as we analyzed above, the direction corresponds to the largest eigenvalue possesses the maximum discriminative power.

3. Nonlinearization via kernelization

In this section, we will extend the ANMM algorithm to the non-linear case via the kernel method [19]. More formally, we will first map the data set from the original space $\mathbb{R}^d$ to a high (usually infinite) dimensional feature space $\mathscr{F}$ through a nonlinear mapping $\Phi : \mathbb{R}^d \longrightarrow \mathscr{F}$, and apply linear ANMM there.

In the feature space $\mathscr{F}$, the *Euclidean distance* between $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$ can be computed as

$$
\begin{aligned}
\|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\| &= \sqrt{(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^T (\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))} \\
&= \sqrt{\mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij}},
\end{aligned}
$$

where $\mathbf{K}_{ij} = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$ is the (i,j)-th entry of the *kernel matrix* $\mathbf{K}$. Thus we can use $\mathbf{K}$ to find the *heterogeneous neighborhood* and *homogeneous neighborhood* for each $\mathbf{x}_i$ in the feature space, and the *total average neighborhood margin* becomes

$$
\gamma^\Phi = \sum_{k=1}^l \mathbf{w}_k^T (\mathbf{S}^\Phi - \mathbf{C}^\Phi) \mathbf{w}_k, \tag{8}
$$

where

$$\mathbf{S}^{\Phi} = \sum_{\substack{i,k: \\ \Phi(\mathbf{x}_k) \in \mathcal{N}^e_{\Phi(\mathbf{x}_i)}}} \frac{(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_k))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_k))^T}{|\mathcal{N}^e_{\Phi(\mathbf{x}_i)}|},$$

$$\mathbf{C}^{\Phi} = \sum_{\substack{i,j: \\ \Phi(\mathbf{x}_j) \in \mathcal{N}^o_{\Phi(\mathbf{x}_i)}}} \frac{(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^T}{|\mathcal{N}^o_{\Phi(\mathbf{x}_i)}|},$$

where $\mathcal{N}^e_{\Phi(\mathbf{x}_i)}$ and $\mathcal{N}^o_{\Phi(\mathbf{x}_i)}$ are the heterogeneous and homogeneous neighborhood of $\Phi(\mathbf{x}_i)$. It is impossible to compute $\mathbf{S}^{\Phi}$ and $\mathbf{C}^{\Phi}$ directly since we usually do not know the explicit form of Φ. To avoid such a problem, we notice that each $\mathbf{w}_k$ lies in the span of $\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_2), \ldots, \Phi(\mathbf{x}_N)$, i.e.,

$$\mathbf{w}_k = \sum_{p=1}^{N} \alpha_p^k \Phi(\mathbf{x}_p).$$

Therefore

$$\mathbf{w}_k^T \Phi(\mathbf{x}_i) = \sum_{p=1}^{N} \alpha_p^k \Phi(\mathbf{x}_p)^T \Phi(\mathbf{x}_i) = (\boldsymbol{\alpha}^k)^T \mathbf{K}_{\cdot i},$$

where $\boldsymbol{\alpha}^k$ is a column vector with its p-th entry equal to α_p^k, $\mathbf{K}_{\cdot i}$ is the i-th column of $\mathbf{K}$. Thus

$$\mathbf{w}_k^T (\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^T \mathbf{w}_k = (\boldsymbol{\alpha}^k)^T (\mathbf{K}_{\cdot i} - \mathbf{K}_{\cdot j})(\mathbf{K}_{\cdot i} - \mathbf{K}_{\cdot j})^T \boldsymbol{\alpha}^k.$$

Define the matrices

$$\tilde{\mathbf{S}}^{\Phi} = \sum_{\substack{i,k: \\ \Phi(\mathbf{x}_k) \in \mathcal{N}^e_{\Phi(\mathbf{x}_i)}}} \frac{(\mathbf{K}_{\cdot i} - \mathbf{K}_{\cdot k})(\mathbf{K}_{\cdot i} - \mathbf{K}_{\cdot k})^T}{|\mathcal{N}^e_{\Phi(\mathbf{x}_i)}|}, \tag{9}$$

$$\tilde{\mathbf{C}}^{\Phi} = \sum_{\substack{i,j: \\ \Phi(\mathbf{x}_j) \in \mathcal{N}^o_{\Phi(\mathbf{x}_i)}}} \frac{(\mathbf{K}_{\cdot i} - \mathbf{K}_{\cdot j})(\mathbf{K}_{\cdot i} - \mathbf{K}_{\cdot j})^T}{|\mathcal{N}^o_{\Phi(\mathbf{x}_i)}|}, \tag{10}$$

then

$$\gamma^{\Phi} = \sum_{k=1}^{l} \mathbf{w}_k^T (\mathbf{S}^{\Phi} - \mathbf{C}^{\Phi}) \mathbf{w}_k = \sum_{k=1}^{l} (\mathbf{w}_k \mathbf{S}^{\Phi} \mathbf{w}_k - \mathbf{w}_k \mathbf{C}^{\Phi} \mathbf{w}_k)$$

$$= \sum_{k=1}^{l} (\boldsymbol{\alpha}^k)^T (\tilde{\mathbf{S}}^{\Phi} - \tilde{\mathbf{C}}^{\Phi}) \boldsymbol{\alpha}^k. \tag{11}$$

Similar to Eq. (5), we also add the constraints that $(\boldsymbol{\alpha}^k)^T(\boldsymbol{\alpha}^k)=1$ ($k= 1, 2, \ldots, l$). Then the optimal $(\boldsymbol{\alpha}^k)$'s are the eigenvectors of $\tilde{\mathbf{S}}^{\Phi} - \tilde{\mathbf{C}}^{\Phi}$ corresponding to its largest l eigenvalues. For a new test point $\mathbf{z}$, its k-th extracted feature can be computed by

$$\mathbf{w}_k^T \Phi(\mathbf{z}) = \sum_{p=1}^{N} \alpha_p^k \Phi(\mathbf{x}_p)^T \Phi(\mathbf{z}) = (\boldsymbol{\alpha}^k)^T \mathbf{K}_{\cdot \mathbf{z}}^t, \tag{12}$$

where we use $\mathbf{K}^t$ to denote the kernel matrix between the training set and the testing set.

We can apply a method similar to the one introduced in Section 2.3 to determine the optimal dimensionality of the projected space for *kernel average neighborhood margin maximization* (KANMM). More concretely, since what we want to maximize is γ_{Φ} in Eq. (11), which is equivalent to the sum of the largest l eigenvalues of $\tilde{\mathbf{S}}^{\Phi} - \tilde{\mathbf{C}}^{\Phi}$ according to the *Ky Fan* theorem. Therefore the optimal dimensionality l^* can be just set to the number of positive eigenvalues of $\tilde{\mathbf{S}}^{\Phi} - \tilde{\mathbf{C}}^{\Phi}$.

The main procedure *kernel average neighborhood margin maximization* algorithm is summarized in Table 2.

Table 2
Kernel average neighborhood margin maximization.

Input: Training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, Testing set $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M\}$, Neighborhood size
$\quad$ $|\mathcal{N}^o_{\Phi}|$, $|\mathcal{N}^e_{\Phi}|$, Kernel parameter $\boldsymbol{\theta}$;
Output: $l \times M$ feature matrix $\mathbf{F}$ extracted from $\mathcal{Z}$.
1. Construct the kernel matrix $\mathbf{K}$ on the training set;
2. Construct the *heterogeneous neighborhood* and *homogeneous neighborhood* for each $\Phi(\mathbf{x}_i)$;
3. Compute $\tilde{\mathbf{S}}^{\Phi}$ and $\tilde{\mathbf{C}}^{\Phi}$ using Eq. (9) and Eq. (10) respectively;
4. Do eigenvalue decomposition on $\tilde{\mathbf{S}}^{\Phi} - \tilde{\mathbf{C}}^{\Phi}$, store the eigenvectors $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_l\}$ corresponding to its l positive eigenvalues;
5. Construct the kernel matrix between the training set and the testing set $\mathbf{K}^t$ with its (i,j)-th entry $\mathbf{K}_{ij}^t = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{z}_j)$.
6. Output $\mathbf{F}^{\Phi}$ with $\mathbf{F}_{ij}^{\Phi} = (\boldsymbol{\alpha}^i)^T \mathbf{K}_{\cdot j}^t$.

4. Multilinearization via tensorization

Till now the ANMM method we have introduced is based on the basic assumption that the data are in vectorized representations. Therefore it is necessary to derive the *tensor form* of our ANMM method. First let us introduce some notations and definitions.

Let A be a tensor of $d_1 \times d_2 \times \cdots \times d_K$. The *order* of A is K and the f-th dimension (or *mode*) of A is of size d_f. A single entry within a tensor is denoted by $A_{i_1 i_2 \ldots i_K}$.

Definition 5 (*Scalar product*). The *scalar product* $\langle A, B \rangle$ of two tensors $A, B \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_K}$ is defined as

$$\langle A, B \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_K} A_{i_1 i_2 \ldots i_K} B^*_{i_1 i_2 \ldots i_K},$$

where $*$ denotes the *complex conjugation*. Furthermore, the *Frobenius norm* of a tensor A is defined as

$$\|A\|_F = \sqrt{\langle A, A \rangle}.$$

Definition 6 (*f-Mode product*). The f-mode product of a tensor $A \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_K}$ and a matrix $\mathbf{U} \in \mathbb{R}^{d_f \times g_f}$ is an $d_1 \times d_2 \times \cdots \times d_{f-1} \times g_f \times d_{f+1} \times \cdots \times d_K$ tensor denoted as $A \times_f \mathbf{U}$, where the corresponding entries are given by

$$(A \times_f \mathbf{U})_{i_1 \ldots i_{f-1} j_f i_{f+1} \ldots i_K} = \sum_{i_f} A_{i_1 \ldots i_{f-1} i_f i_{f+1} \ldots i_K} \mathbf{U}_{i_f j_f}.$$

Definition 7 (*f-Mode unfolding*). Let A be a $d_1 \times \cdots \times d_K$ tensor and $(\pi_1, \ldots, \pi_{K-1})$ be any permutation of the entries of the set $\{1, \ldots, f-1, f+1, \ldots, K\}$. The f-mode unfolding of the tensor A into a $d_f \times \prod_{l=1}^{K-1} d_{\pi_l}$ matrix, denoted by $\mathbf{A}^{(f)}$, is defined as

$$A \in \mathbb{R}^{d_1 \times \cdots \times d_K} \Rightarrow_f \mathbf{A}^{(f)} \in \mathbb{R}^{d_f \times \prod_{l=1}^{K-1} d_{\pi_l}},$$

where $\mathbf{A}_{i_f j}^{(f)} = A_{i_1 \ldots i_K}$ with

$$j = 1 + \sum_{l=1}^{K-1} (i_{\pi_l} - 1) \prod_{l'=1}^{l-1} d_{\pi_{l'}}.$$

The tensor based criterion for ANMM is that, given N data points $X_1, \ldots, X_N$ embedded in a tensor space $\mathbb{R}^{d_1 \times d_2 \times \cdots \times d_K}$, we want to pursue K optimal interrelated projection matrices $\mathbf{U}_i \in \mathbb{R}^{l_i \times d_i}$ ($l_i < d_i$, $i = 1, 2, \ldots, K$), which maximize the *average neighborhood margin* measured in the tensor metric. That is

$$\gamma = \sum_i \left(\sum_{j: X_j \in \mathcal{N}_i^o} \frac{\|Y_i - Y_j\|_F^2}{|\mathcal{N}_i^o|} - \sum_{k: X_k \in \mathcal{N}_i^e} \frac{\|Y_i - Y_k\|_F^2}{|\mathcal{N}_i^e|} \right),$$

where $Y_i = X_i \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times \cdots \times_K \mathbf{U}_K$. Note that directly maximizing γ is almost infeasible since it is a higher-order optimization problem. Generally such type of problems can be solved approximately by employing an iterative scheme which was originally proposed by [32] for low-rank approximation of second-order tensors. Later [27] extended it for higher-order tensors. In the following we will adopt such an iterative scheme to solve the optimization problem.

Given $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_{f-1}, \mathbf{U}_{f+1}, \ldots, \mathbf{U}_K$, let

$$Y_i^f = X_i \times_1 \mathbf{U}_1 \times \cdots \times_{f-1} \mathbf{U}_{f-1} \times_{f+1} \mathbf{U}_{f+1} \times \cdots \times_K \mathbf{U}_K. \tag{13}$$

Then, by the corresponding *f-mode unfolding*, we can get $Y_i^f \Rightarrow_f \mathbf{Y}_i^{(f)}$. Moreover, we can easily derive that

$$\|Y_i^f \times_f \mathbf{U}_f\|_F = \|(\mathbf{Y}_i^{(f)})^T \mathbf{U}_f\|_F.$$

Therefore we have

$$\begin{aligned}
\|Y_i - Y_j\|_F^2 &= \|X_i \times_1 \mathbf{U}_1 \times \cdots \times_K \mathbf{U}_K - X_j \times_1 \mathbf{U}_1 \times \cdots \times_K \mathbf{U}_K\|_F^2 \\
&= \|Y_i^f \times_f \mathbf{U}_f - Y_j^f \times_f \mathbf{U}_f\|_F^2 \\
&= \|(\mathbf{Y}_i^{(f)})^T \mathbf{U}_f - (\mathbf{Y}_j^{(f)})^T \mathbf{U}_f\|_F^2 \\
&= \mathrm{tr}[\mathbf{U}_f^T(\mathbf{Y}_i^{(f)} - \mathbf{Y}_j^{(f)})(\mathbf{Y}_i^{(f)} - \mathbf{Y}_j^{(f)})^T \mathbf{U}_f]
\end{aligned}$$

Then knowing $\mathbf{U}_1, \ldots, \mathbf{U}_{f-1}, \mathbf{U}_{f+1}, \ldots, \mathbf{U}_K$, we can rewrite the *compactness matrix* and *scatterness matrix* in *tensor* ANMM as

$$\mathbf{S} = \sum_{\substack{i,k: \\ \mathbf{x}_k \in \mathcal{N}_i^e}} \frac{(\mathbf{Y}_i^{(f)} - \mathbf{Y}_k^{(f)})(\mathbf{Y}_i^{(f)} - \mathbf{Y}_k^{(f)})^T}{|\mathcal{N}_i^e|}, \tag{14}$$

$$\mathbf{C} = \sum_{\substack{i,j: \\ \mathbf{x}_k \in \mathcal{N}_i^o}} \frac{(\mathbf{Y}_i^{(f)} - \mathbf{Y}_j^{(f)})(\mathbf{Y}_i^{(f)} - \mathbf{Y}_j^{(f)})^T}{|\mathcal{N}_i^o|}, \tag{15}$$

and our optimization problem (with respect to $\mathbf{U}_f$) becomes

$$\max_{\mathbf{U}_f} \mathrm{tr}[\mathbf{U}_f^T(\mathbf{S} - \mathbf{C})\mathbf{U}_f]. \tag{16}$$

Let us expand $\mathbf{U}_f$ as $\mathbf{U}_f = (\mathbf{u}_{f1}, \mathbf{u}_{f2}, \ldots, \mathbf{u}_{fl_f})$ with $\mathbf{u}_{fi}$ corresponding to the *i*-th column of $\mathbf{U}_f$, then Eq. (16) can be rewritten as

$$\max \sum_{i=1}^{l_f} \mathbf{u}_{fi}^T(\mathbf{S} - \mathbf{C})\mathbf{u}_{fi}. \tag{17}$$

We also add the constraint that $\mathbf{u}_{fi}^T \mathbf{u}_{fi} = 1$ to restrict the scale of $\mathbf{U}_f$. Then the above optimization problem can be efficiently solved via eigenvalue decomposition, and the optimal l_f can be determined by the number of positive eigenvalues of $\mathbf{S} - \mathbf{C}$.

The main procedure of the *tensor average neighborhood margin maximization* (TANMM) is summarized in Table 3.

5. Experiments

In this section, we will carry out a set of experiments to show the effectiveness of our *marginFace* method for face representation and recognition.

5.1. Face representation using marginFaces

As we introduced before, usually a face image of size $m \times n$ can be described as a point in the $m \times n$-dimensional image space. However, due to the unwanted variations resulting from changes in lighting, facial expression, and pose, the original image space might not be a good space for visual representation and recognition.

Table 3
Tensor average neighborhood margin maximization.

Input: Training set $\mathscr{D} = \{(X_i, y_i)\}_{i=1}^N$, Testing set $\mathscr{Z} = \{Z_1, Z_2, \ldots, Z_M\}$, where $X_i, Z_j \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_K}$, Neighborhood size $|\mathcal{N}^o|$, $|\mathcal{N}^e|$, Iteration steps T_{max}, Difference ε;
Output: Feature tensors $\{F_i\}_{i=1}^M$ extracted from $\mathscr{Z}$, where $F_i \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_K}$.
1. Initialize $\mathbf{U}_1^0 = \mathbf{I}_{d_1}, \mathbf{U}_2^0 = \mathbf{I}_{d_2}, \ldots, \mathbf{U}_K^0 = \mathbf{I}_{d_K}$, where $\mathbf{I}_{d_i}$ represents the $d_i \times d_i$
1. identity matrix;
2. For $t = 1, 2, \ldots, T_{max}$ do
  For $f = 1, 2, \ldots, K$ do
    (a). Compute Y_i^f by Eq. (13);
    (b). $Y_i^f \Rightarrow_f \mathbf{Y}_i^{(f)}$;
    (c). Compute $\mathbf{S}$ and $\mathbf{C}$ using Eq. (14) and (15);
    (d). Do eigenvalue decomposition on $\mathbf{S} - \mathbf{C}$: $(\mathbf{S} - \mathbf{C})\mathbf{U}_f^t = \mathbf{U}_f^t \Lambda_f$ with $\mathbf{U}_f^t \in \mathbb{R}^{d_f \times l_f}$, and l_f is equal to the number of positive eigenvalues of $\mathbf{S} - \mathbf{C}$;
    (f). if $\|\mathbf{U}_f^t - \mathbf{U}_f^{t-1}\| < \varepsilon$, break;
  End for.
  End for.
3. Output $F_i = Z_i \times_1 \mathbf{U}_1^t \cdots \times_K \mathbf{U}_K^t$.

In Section 2, we have discussed how to learn a good discriminative subspace based on ANMM. The eigenvectors spanning such subspace can be obtained via eigenvalue decomposition of $\mathbf{S} - \mathbf{C}$ which are defined in Eqs. (2) and (3). We can resize these eigenvectors and display them as images. Using the ORL face database (see the introduction in the next subsection) as the training set, we represent its first 10 *marginFaces* in Fig. 7(c), together with its first 10 *Eigenfaces* and *Fisherfaces*. From the figures we can clearly see that the discriminative information contained in *marginFaces* is much richer than in *Eigenfaces* and *Fisherfaces*.

5.2. Face recognition using marginFaces

In this subsection, we investigate the performance of our proposed ANMM, kernel ANMM and tensor ANMM methods for face recognition. We have done three groups of experiments to achieve this goal:

(1) *Linear methods*. In this set of experiments, the performance of original ANMM is compared with the traditional PCA [24] method, LDA (PCA + LDA) method [1], and three *margin* based methods, namely the *maximum margin criterion* (MMC) method [12], the *stepwise nonparametric maximum margin criterion* (SNMMC) method [15] and the *marginal Fisher analysis* (MFA) method [28].
(2) *Kernel methods*. In this set of experiments, the performance of the KANMM method is compared with the KPCA and the KDA method [31].
(3) *Tensor methods*. In this set of experiments, the performance of the tensor ANMM method is compared with the tensor PCA (TPCA) and the tensor LDA (TLDA) methods [3].

In this study, three face data set are used:

(1) The ORL face data set.[2] There are 10 images for each of the 40 human subjects, which were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to $20°$. The original images (with 256 gray levels) have size 92×112, which are resized to 32×32 for efficiency.

---

[2] http://www.uk.research.att.com/facedatabase.html

Fig. 4. Some sample faces of the ORL data set.

Fig. 5. Some sample faces of the YALE data set.

(2) The *Yale* face data set.[3] It contains 11 grayscale images for each of the 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. In our experiment, the images were also resized to 32×32.

(3) The *CMU PIE* face data set [22]. It contains 68 individuals with 41,368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination, and expression. In our experiments, five near frontal poses (C05, C07, C09, C27, C29) are selected under different illuminations, lighting and expressions which leaves us 170 near frontal face images for each individual, and all the images were also resized to 32×32.

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 images for recognition. Some sample face images of the three face databases are shown in Figs. 4, 5, and 6, respectively.

The free parameters for the tested methods were determined in the following ways:

(1) For the ANMM-series methods (including ANMM, KANMM, TANMM), the sizes of the *homogeneous* and *heterogeneous* neighborhoods for each data point are all set by fivefold cross-validation from $\{5, 10, 15, 20\}$.[4] For MFA, when constructing the intrinsic graph and penalty graph, we also set the with-in class and between-class neighborhood size by fivefold cross-validation from $\{5, 10, 15, 20\}$.

(2) For the *kernel based methods*, we all adopt the Gaussian kernel, and the variance of the Gaussian kernel were set by fivefold

cross-validation from

$$\{4^{-4}, 4^{-3}, 4^{-2}, 4^{-1}, 1, 4, 4^2, 4^3, 4^4\}.$$

(3) For the *tensor based methods*, we require that the projected images are also square, i.e., of dimension $r \times r$ with $r = \min\{l_1^*, l_2^*\}$, and l_1^*, l_2^* are the optimal dimensionalities obtained from the TANMM algorithm in Table 3.

After setting all the free parameters, we first project the face images into a low-dimensional space by different methods (for ANMM-series methods, the final dimensionalities of the embedded data are determined automatically by the methods introduced in Tables 1–3,[5] for other methods, the final dimensionalities are set by exhaustive search as in traditional approaches). For linear methods and kernel methods, we first vectorize the training images and obtain the low-dimensional embeddings of them, then the nearest-neighbor classifier is employed to perform classification, and the standard Euclidean distance is used. For tensor based methods, we treat each image as an 2D tensor and obtain the low-dimensional embeddings of the training images in an 2D space, finally the nearest neighbor classifier with Euclidean distance is also employed to perform classification.

The experimental results of the *linear methods* on the three data sets are shown in Figs. 8, 9 and 10, respectively. In all the figures, the abscissas represent the projected dimensions, and the ordinates are the average recognition accuracies of 50 independent runs. From the figures we observe that:

- The discriminative analysis based methods perform clearly better than PCA as they incorporates the label information.
- MMC and PCA + LDA usually performs similar to each other, however, MMC can extract more features than LDA as it avoids the singularity problem.
- The methods that explore the local information contained in the data set (ANMM, MFA and SNMMC) usually outperforms the global methods (MMC, LDA).
- ANMM performs similarly with MFA and SNMMC when the training data set is small, since in this case the margin may not be accurately estimated. When the training set size grows, ANMM performs much better.
- The discrimination power of ANMM will be enhanced with the increase of final projected dimensionality, but it will not increase all the time. When the final dimensionality is higher than some threshold, the final classification accuracy will stand still.

Table 4 shows the experimental results of all the methods on three data sets, where the value in each entry represents the average recognition accuracy (in percentages) of 50 independent trials, and the number in brackets is the corresponding projected dimension. The table shows that the ANMM-series methods can perform better than those traditional methods on the three data sets.

In summary, we can see that on these three standard benchmark data sets, our algorithm all outperforms other competitors, in spite of the variations on the poses and lightening conditions. One possible reason is because our ANMM algorithm can capture the local discriminality of the data set very well.

Besides, we also conduct an experiment on testing how will our algorithm performs with the increasing of subjects. We use the *CMU*

---

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

[4] The fivefold cross-validation proceeds like this, we first split the whole data set into a training set and a testing set, then we take the training set and split it into five folds. During the cross-validation, we take four folds for training and the other four folds for testing, and repeat the process 4 times and choose the parameter settings with the highest average accuracy. Then the parameter will be used to learn the projection direction from the whole training set and classify the testing set.

[5] Specifically, for ANMM, the dimensionality equals the number of nonnegative eigenvalues of $\mathbf{S} - \mathbf{C}$ defined in Eqs. (2) and (3); for KANMM, the dimensionality equals the number of nonnegative eigenvalues of $\tilde{S}^\Phi - \tilde{C}^\Phi$ defined in Eqs. (9) and (10); for TANMM, the optimal dimensionality of each projection matrix is determined by the nonnegative eigenvalues of $S - C$ defined in Eqs. (14) and (15).
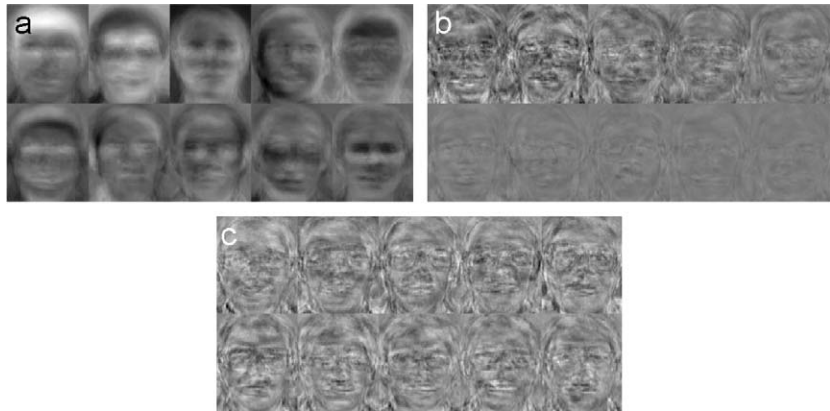
Fig. 6. Some sample faces of the PIE data set.

Fig. 7. The first 10. (a) Eigenfaces. (b) Fisherfaces. (c) marginFaces of the ORL data set.
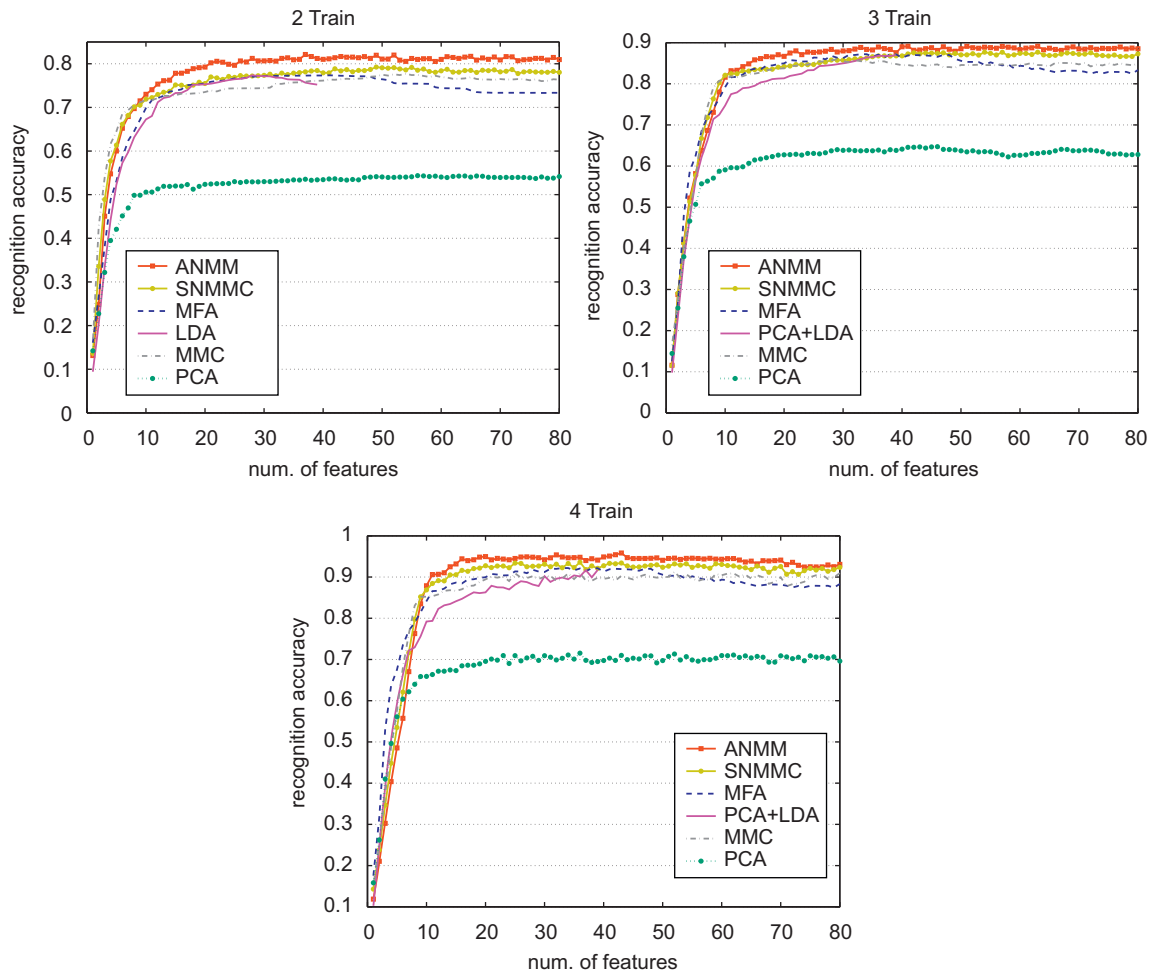

Fig. 8. Face recognition accuracies on the ORL data set with 2, 3, 4 images for each individual randomly selected for training.
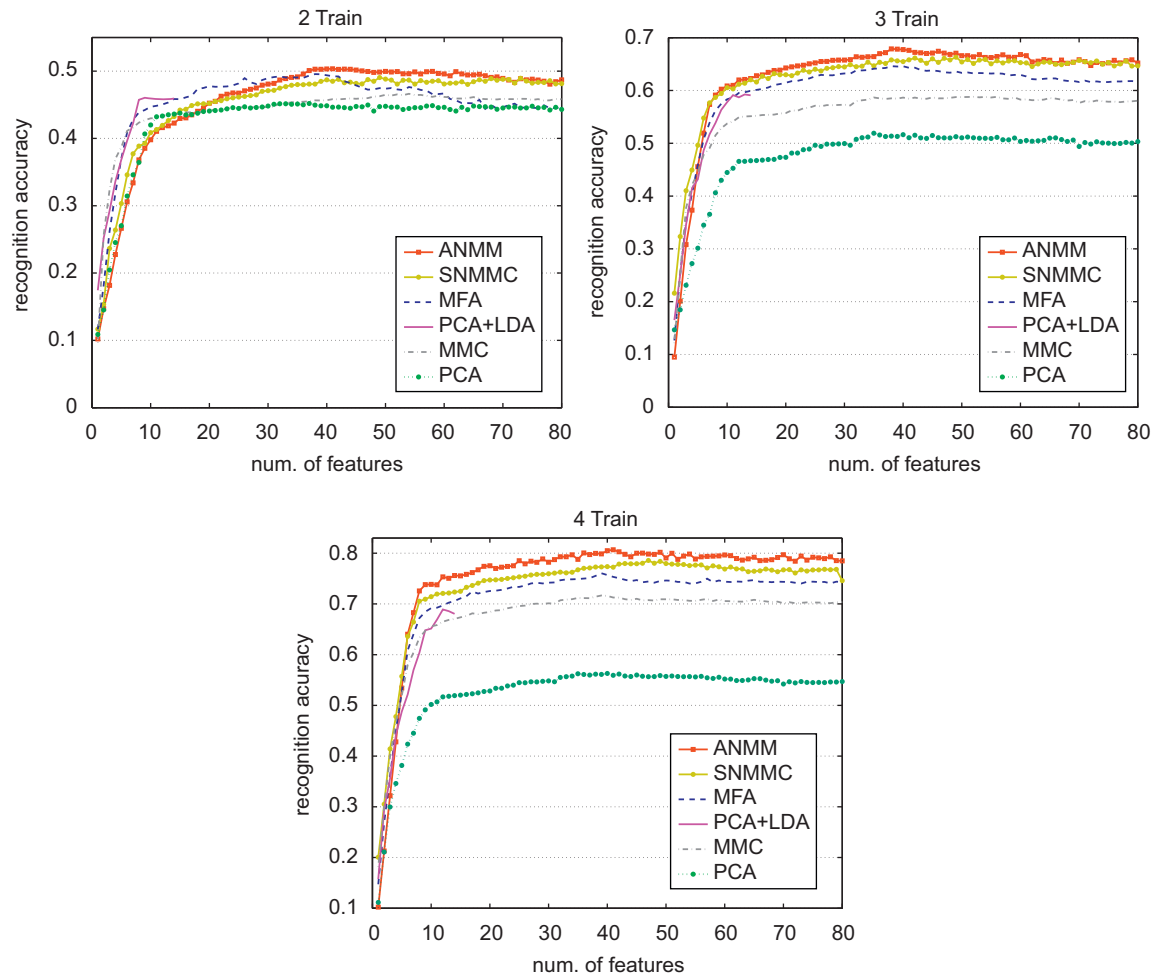
Fig. 9. Face recognition accuracies on the Yale data set with 2, 3, 4 images per individual randomly selected for training.

PIE database as our experimental data set, which contains 68 subjects as we mentioned before. In our experiment, we fist select a subset which only contains the face images of i subjects ($i = 2, 3, \ldots, 67, 68$). For each subset, 10 images per subject are randomly selected for training, and the rest of the images are used for testing. Such process is repeated 50 times and the averaged recognition accuracy is reported in Fig. 11. From the figure we can see that with the increasing of subjects, the recognition accuracy of our algorithm will decrease. The reason for this is because that when the number of subjects increases, the data patterns contained in the training set will also increase which will make the data distribution within each neighborhood more complicated. Accordingly the performance of our algorithm could be affected. Note that the range of the vertical axis is small and thus makes the figure looks bad, in fact it is consistent with the results in Fig. 10 and Table 4.

5.3. Potential applications in face verification

In the final part of the experiments, we investigate the potential of our algorithm to the face verification task. The protocol of our face verification system is the same as in [16]. In our experiments, we first split all data sets into two parts: training set and as test set, then we project the face images into the learned subspace and compute the template vector for each class (subject) from the training set (we selected the mean vector). In the third step we compute the acceptance threshold using the training images and compute the

expected ROC curves as function of the threshold value. Finally we use the testing set to plot the ROC curves.

In our experiments we just use a single threshold γ for all the subjects, such that the claimed subject will be accepted if

$$d < \gamma \cdot w,$$

where d is the Euclidean distance from the candidate pattern to the template, and w is the weight of the corresponding subject class, which is computed by the averaged Euclidean distance of all patterns in the training set from the template for any given subject class.

We conduct experiments on the ORL and Yale data sets. The ORL data set was subdivided into a training set, made up of five images per class (200 images), and a test set, made up of five images per class (200 images). In order to assess verification performances, we used all possible combinations of five images out of 10 to generate the training set (252 cases). Reported results in Fig. 12 refer to the average performances in such 252 cases.

The Yale data set was subdivided into a training set, made up of five images per class (75 images), and a test set, made up of six images per class (90 images). In order to evaluate the verification performances, we used all possible combinations of five images out of 11 images to generate the training set (330 cases). Reported results in Fig. 13 refer to the average performances in such 330 cases.

In both Figs. 12 and 13, the *x*-axis represents the *false acceptance rate* (FAR) and the *y*-axis represents the *false rejection rate* (FRR). For comparison, we also depict the results of PCA and LDA. From
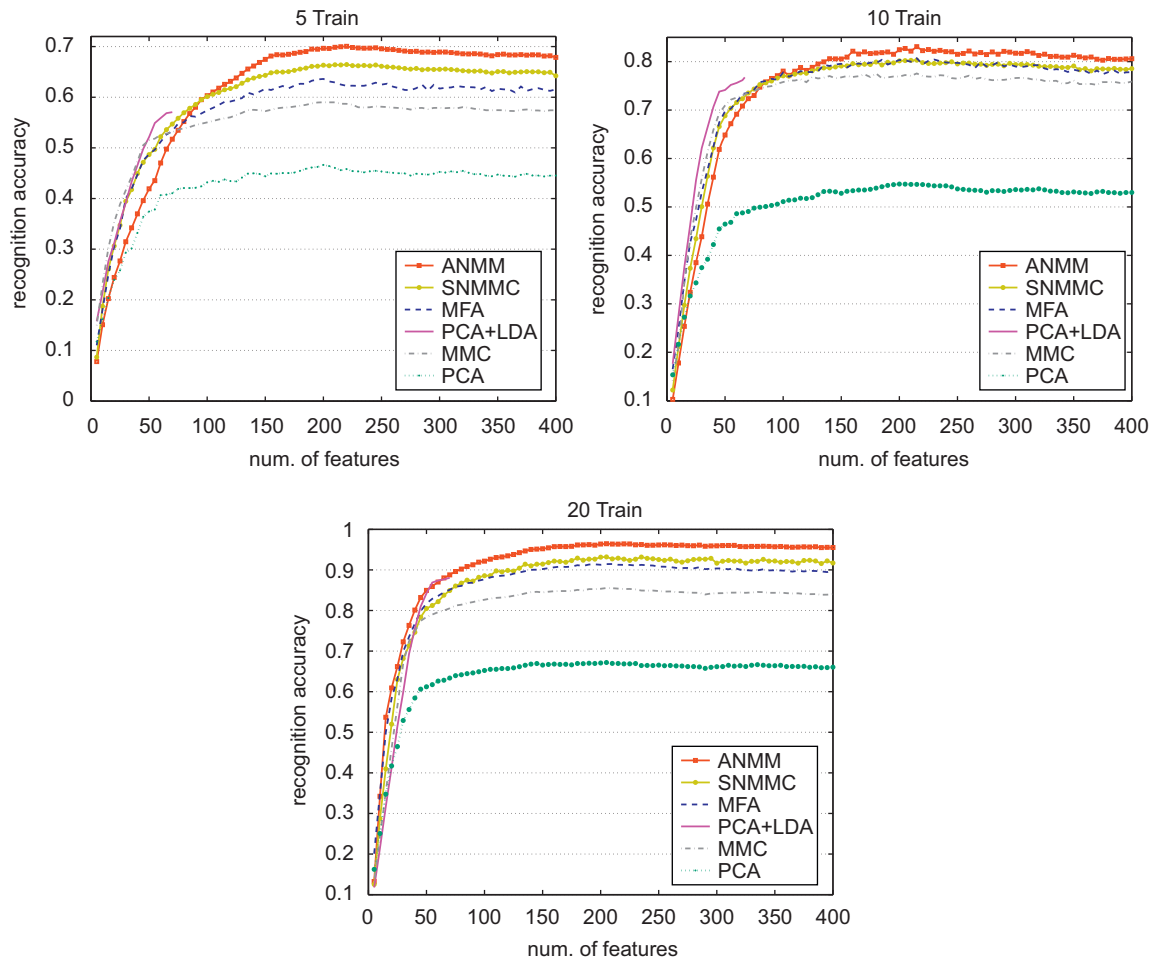
Fig. 10. Face recognition accuracies on the CMU PIE data set with 5, 10, 20 images per individual randomly selected for training.

Table 4
Face recognition results on three data sets (%).

Method	ORL			Yale			CMU PIE		
	2 Train	3 Train	4 Train	2 Train	3 Train	4 Train	5 Train	10 Train	20 Train
PCA	54.35(56)	64.71(64)	71.54(36)	45.19(37)	51.91(35)	56.30(40)	46.64(204)	54.72(213)	67.17(241)
LDA	77.36(28)	86.96(39)	91.71(39)	46.04(9)	59.25(13)	68.90(12)	57.05(62)	76.75(62)	88.06(61)
MMC	77.73(54)	85.98(29)	91.26(52)	46.64(54)	58.80(56)	71.67(39)	57.05(210)	77.56(215)	85.54(195)
SNMMC	79.23(49)	87.68(54)	93.59(36)	49.05(49)	66.31(49)	78.57(47)	66.45(223)	80.28(213)	91.20(202)
MFA	77.34(41)	87.19(33)	92.19(33)	49.56(38)	64.60(38)	76.05(39)	63.60(210)	80.69(232)	88.69(205)
ANMM	**82.13(37)**	**89.13(41)**	**95.84(43)**	**50.35(41)**	**67.87(38)**	**80.69(41)**	**70.05(222)**	**82.08(203)**	**93.46(205)**
KPCA	64.23(50)	75.25(54)	79.26(60)	49.34(45)	55.78(47)	60.72(54)	52.35(341)	60.12(384)	72.25(256)
KDA	80.29(38)	89.13(36)	93.12(38)	52.35(14)	64.89(13)	71.95(14)	62.13(67)	81.27(66)	92.11(65)
KANMM	**85.46(50)**	**92.21(39)**	**96.13(53)**	**54.62(54)**	**69.25(66)**	**80.77(62)**	**72.01(302)**	**82.41(280)**	**93.67(218)**
TPCA	59.22(10^2)	71.25(12^2)	79.86(10^2)	50.15(7^2)	57.23(11^2)	62.30(10^2)	51.17(10^2)	56.65(13^2)	69.09(11^2)
TLDA	80.68(9^2)	89.28(11^2)	93.37(8^2)	51.25(9^2)	66.19(10^2)	75.88(9^2)	60.61(12^2)	80.15(14^2)	92.75(8^2)
TANMM	**85.87(10^2)**	**92.54(9^2)**	**96.22(11^2)**	**55.31(11^2)**	**70.43(8^2)**	**81.56(10^2)**	**73.02(12^2)**	**82.78(9^2)**	**94.32(11^2)**

Figs. 12 and 13 we can see that our ANMM algorithm outperforms PCA and LDA in both data sets, which suggests its potential application in face verification task.

6. Related works

In this section we will briefly review some linear feature extraction methods that are closely related to ANMM, and discuss their relations with ANMM.

6.1. LDA and its variants

Traditional LDA [6] learns the projection matrix **W** by maximizing the following criterion:

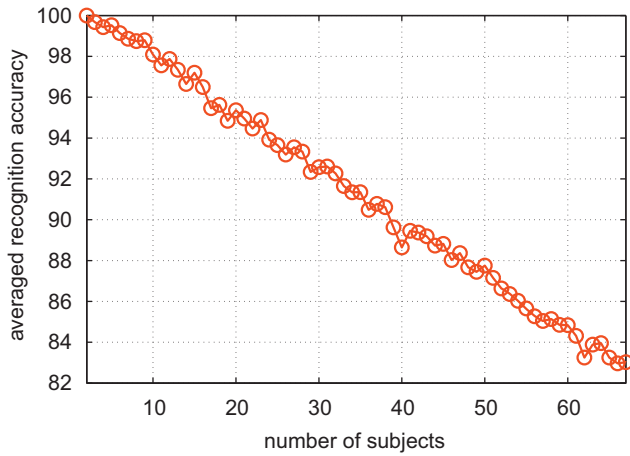$$J = \frac{|\mathbf{W}^T\mathbf{S}_b\mathbf{W}|}{|\mathbf{W}^T\mathbf{S}_w\mathbf{W}|},$$

Fig. 11. Face recognition accuracies on the CMU PIE data set with increasing subjects. The *x*-axis represents the number of subjects in the experimental data set, and the *y*-axis is the recognition accuracy averaged over 50 independent runs with 10 faces per subjects used for training.
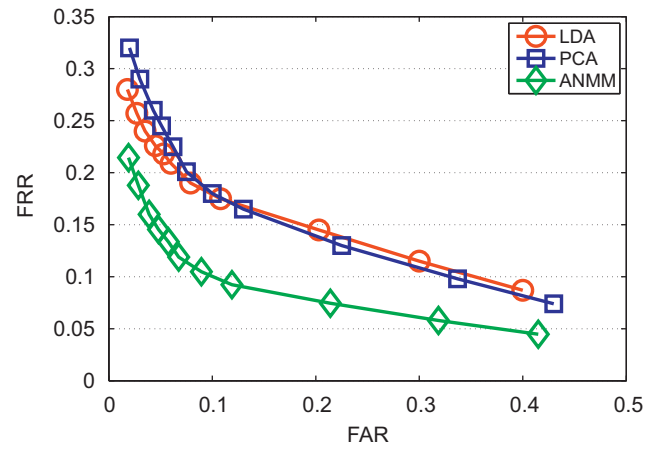

Fig. 13. For the Yale data set, the average ROC curves of the individual algorithms (PCA, LDA and ANMM). In order to assess verification performances, we used all possible combinations of five images out of 11 images to generate the training set. The ROC curve refers to the average of the 330 cases considered.
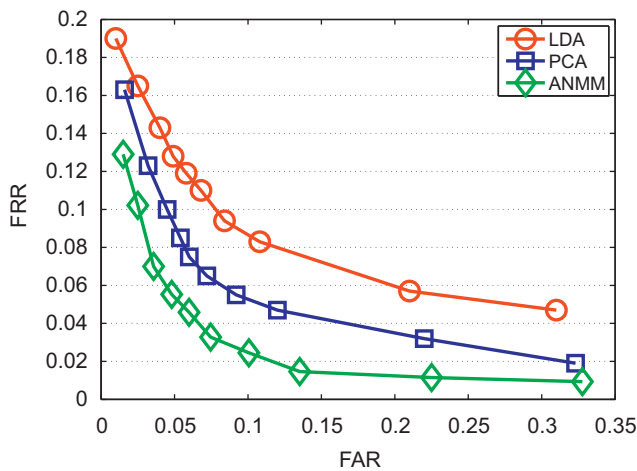

Fig. 12. For the ORL data set, the average ROC curves of the individual algorithms (PCA, LDA and ANMM). In order to assess verification performances, we used all possible combinations of five images out of 10 images to generate the training set. The ROC curve refers to the average of the 252 cases considered.

where $\mathbf{S}_b = \sum_{k=1}^c p_k(\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T$ is the *between-class scatter matrix*, where p_k and $\mathbf{m}_k$ are the prior and mean of class k, and $\mathbf{m}$ is the mean of the entire data set. $\mathbf{S}_w = \sum_{k=1}^c p_k\mathbf{S}_k$ is the *within-class scatter matrix* with $\mathbf{S}_k$ being the covariance matrix of class k.

It has been shown that J can be maximized when $\mathbf{W}$ is constituted by the eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$ corresponding to its l largest eigenvalues [6]. However, when the size of the data set is small, $\mathbf{S}_w$ will become singular. Then $\mathbf{S}_w^{-1}$ does not exist and the *small sample size* problem occurs. Many approaches have been proposed to solve such a problem, such as *PCA + LDA* [1], *null space LDA* [13], *direct LDA* [33], etc. Li et al. [12] further proposed an efficient and robust linear feature extraction method which aims to maximize the following criterion which was called a *margin* in [12]:

$$\mathscr{J} = \mathrm{tr}(\mathbf{W}^T(\mathbf{S}_b - \mathbf{S}_w)\mathbf{W}), \tag{18}$$

where tr($\cdot$) denotes the *matrix trace*. We can see that there is no need for computing any matrix inverse in optimizing the above criterion. However, such a *margin* is lack of geometric intuitions.

Another limitation of traditional LDA is its implicit *Gaussian* assumption of the class conditional densities. To solve the problem, Fukunaga et al. [6] proposed the *nonparametric discriminant analysis* (NDA) method which defines a different *between-class scatter matrix*. Qiu et al. [15] further extended the NDA method and proposed a *stepwise nonparametric margin maximization criterion* approach, which tries to maximize

$$\mathscr{J} = \sum_{i=1}^N w_i(\|\delta_i^E\|^2 - \|\delta_i^I\|^2) \tag{19}$$

in the transformed space, where $\|\delta_i^E\|$ is the distance between $\mathbf{x}_i$ and its nearest neighbor in the different class, $\|\delta_i^I\|$ is the distance between $\mathbf{x}_i$ and its furthest neighbor in the same class. The problem is that using just the nearest (or furthest) neighbor for defining the margin may cause the algorithm sensitive to outliers. The experimental results in Section 5.2 show that our ANMM method can clearly outperform the SNMMC method. Moreover, the stepwise procedure for maximizing $\mathscr{J}$ is time consuming.

6.2. Geometrical methods

Geometrical methods are another family of appearance-based face recognition methods which aim to analyze the face manifolds using the graph theory [28]. Generally, the goal of these methods is to solve the following optimization problem:

$$\min_{\mathbf{W}} \quad \mathrm{tr}(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W})$$

$$\text{s.t.} \quad \mathrm{tr}(\mathbf{W}^T\mathbf{X}\mathbf{B}\mathbf{X}\mathbf{W}^T) = \mathbf{I}, \tag{20}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ is the data matrix, $\mathbf{L}$ is some *Laplacian* matrix defined on the data graph, and $\mathbf{B}$ is the constraint matrix to avoid the trivial solutions. It can be easily observed that the solution of the above problem can be obtained by a *generalized eigenvalue decomposition* procedure. Some typical approaches include He et al.'s *Laplacianface* method [9], Cai et al.'s *orthogonal-Laplacianface* method [4], Chen et al.'s *local discriminant embedding* (LDE) approach [5] and Yan et al.'s *marginal Fisher analysis* method [28]. The only difference between those methods is that they used different Laplacians and

constraint matrices. However, all these methods still suffers from the singularity problem since they cannot guarantee that $\mathbf{XBX}^T$ is nonsingular, and usually a PCA preprocessing step is still needed.

6.3. Metric learning approaches

As we introduced in Section 2.4, from another point of view, linear feature extraction can also be treated as learning a proper *Mahalanobis distance* between pairwise points, since

$$\|\mathbf{y}_i - \mathbf{y}_j\|^2 = \|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j).$$

Let $\mathbf{M} = \mathbf{W}\mathbf{W}^T$, then

$$\|\mathbf{y}_i - \mathbf{y}_j\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j).$$

Therefore, learning the projection matrix $\mathbf{W}$ is equivalent to learn an efficient *Mahalanobis distance*, or, more concretely, learn a proper $\mathbf{M}$.

Goldberger et al. [8] proposed a probabilistic supervised metric learning method called *neighborhood component analysis*. However, their optimization criterion is nonconvex and the gradient ascent iteration is computationally rather inefficient. Globerson et al. [7] proposed the *maximally collapsing metric learning* (MCML) method to improve NCA, whose optimization criterion is convex, but from the feature extraction perspective, they require the data dimension in the projection space should be the same as the dimension of the original space, i.e., there is no dimensionality reduction in MCML. Weinberger et al. [25] proposed a large margin criterion to learn a proper $\mathbf{M}$ for *k nearest neighbor* classifier, and optimize it through a *semidefinite programming* (SDP) procedure. Unfortunately, the computational burden of *SDP* is high, which limits its potential applications in high-dimensional data sets.

7. Conclusions and discussions

In this paper we proposed a novel supervised linear feature extraction method named *average neighborhood margin maximization*. For each data point, ANMM aims at pulling the neighboring points with the same class label towards it as near as possible, while simultaneously pushing the neighboring points with different labels away from it as far as possible. Moreover, as many computer vision and pattern recognition problems are intrinsically nonlinear or multilinear, we also derive the kernelized and tensorized counterparts of ANMM. Finally the experimental results on face recognition are presented to show the effectiveness of our proposed approaches.

As we mentioned in Section 6, linear feature extraction methods can also be viewed as learning a proper *Mahalanobis distance* in the original data space. Thus ANMM can also be used for distance metric learning. From such a viewpoint, our algorithm is more efficient in that it only needs to learn the transformation matrix, but not the whole covariance matrix as in traditional metric learning algorithms [25].

Acknowledgment

References

[1] P.N. Belhumeur, J. Hespanda, D. Kiregeman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.

[2] L. Bottou, V. Vapnik, Local learning algorithms, Neural Computation 4 (1992) 888–900.

[3] D. Cai, X. He, J. Han, Subspace learning based on tensor analysis, Technical Report No. 2572, Department of Computer Science, University of Illinois at Urbana-Champaign (UIUCDCS-R-2005-2572), 2005.

[4] D. Cai, X.F. He, J.W. Han, H.J. Zhang, Orthogonal Laplacianfaces for face recognition, IEEE Transactions on Image Processing 15 (11) (2006) 3608–3614.

[5] H.-T. Chen, H.-W. Chang, T.-L. Liu, Local discriminant embedding and its variants, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 846–853.

[6] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press, New York, 1990.

[7] A. Globerson, S. Roweis, Metric learning by collapsing classes, Advances in Neural Information Processing Systems 18 (2006) 451–458.

[8] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighbourhood components analysis, Advances in Neural Information Processing Systems 17 (2005) 513–520.

[9] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using Laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.

[10] A.K. Jain, B. Chandrasekaran, Dimensionality and sample size considerations in pattern recognition practice, in: Handbook of Statistics, North-Holland, Amsterdam, 1982.

[11] I.T. Jolliffe, Principal Component Analysis, Springer, New York, 1986.

[12] H. Li, T. Jiang, K. Zhang, Efficient and robust feature extraction by maximum margin criterion, Advances in Neural Information Processing Systems 16 (2004) 97–104.

[13] K. Liu, Y. Cheng, J. Yang, A generalized optimal set of discriminant vectors, Pattern Recognition 25 (7) (1992) 731–739.

[14] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, K.-R. Müller, Fisher discriminant analysis with kernels, in: Neural Networks for Signal Processing IX, Proceedings of the 1999 IEEE Signal Processing Society Workshop, 1999, pp. 41–48.

[15] X. Qiu, L. Wu, Face recognition by stepwise nonparametric margin maximum criterion, in: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, 2005, pp. 1567–1572.

[16] G.L. Marcialis, F. Roli, Fusion of LDA and PCA for face verification, in: Proceedings of the International ECCV 2002 Workshop Copenhagen on Biometric Authentication, Lecture Notes in Computer Science, vol. 2359, pp. 30–38.

[17] H. Murase, S.K. Nayar, Visual learning and recognition of 3-D objects from appearance, International Journal of Computer Vision 14 (1) (1995) 5–24.

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

[19] B. Schölkopf, A. Smola, Learning with Kernels, The MIT Press, Cambridge, MA, London, England, 2002.

[20] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (1998) 1299–1319.

[21] H.S. Seung, D.D. Lee, The manifold ways of perception, Science 290 (5500) (2000) 2268–2269.

[22] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression database, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (12) (2003) 1615–1618.

[23] F. De la Torre, M.A.O. Vasilescu, Linear and multilinear (tensor) methods for vision, graphics, and signal processing, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Tutorial, 2006.

[24] M.A. Turk, A.P. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 71–96.

[25] K.Q. Weinberger, J. Blitzer, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, Advances in Neural Information Processing Systems 18 (2006) 1475–1482.

[26] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, Berlin, 1995.

[27] H. Wang, Q. Wu, L. Shi, Y. Yu, N. Ahuja, Out-of-core tensor approximation of multi-dimensional matrices of visual data, in: Proceedings of ACM SIGGRAPH, 2005, pp. 527–535.

[28] S. Yan, D. Xu, B. Zhang, H. Zhang, Graph embedding: a general framework for dimensionality reduction, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 830–837.

[29] S. Yan, D. Xu, L. Zhang, Q. Yang, X. Tang, H. Zhang, Multilinear discriminant analysis for face recognition, IEEE Transactions on Image Processing 16 (1) (2006) 212–220.

[31] M.-H. Yang, Kernel Eigenfaces vs. kernel Fisherfaces: face recognition using kernel methods, in: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 2002, pp. 215–220.

[32] J. Ye, Generalized low rank approximations of matrices, Machine Learning 61 (1–3) (2005) 167–191.

[33] H. Yu, J. Yang, A direct LDA algorithm for high dimensional data with application to face recognition, Pattern Recognition 34 (10) (2001) 2067–2070.

About the Author—FEI WANG is a Ph.D. candidate of grade four in Department of Automation, Tsinghua University, Beijing, China. His main research interests include machine learning, data mining and pattern recognition.

About the Author—XIN WANG is a Ph.D. candidate of grade 1 in School of Computing and Information Sciences, Florida International University, Miami, Florida. Her main research interests include machine learning, data mining and information retrieval.

About the Author—DAOQIANG ZHANG is currently a professor in the Department of Computer Science and Engineering at Nanjing University of Aeronautics and Astronautics, China. He received his B.Sc. and Ph.D. degrees in computer science from Nanjing University of Aeronautics and Astronautics, China, in 1999 and 2004, respectively. From September 2004 to December 2006, he was a postdoctoral fellow in the LAMDA group of Department of Computer Science and Technology at Nanjing University. His current research interests mainly include pattern recognition, neural computing, machine learning, data mining and image processing.

About the Author—CHANGSHUI ZHANG is a Professor in Department of Automation, Tsinghua University, Beijing, China. He currently serves as an associate editor of the Pattern Recognition Journal.

About the Author—TAO LI is an Assistant Professor in School of Computing and Information Sciences, Florida International University, Miami, Florida. His main research interests include machine learning, data mining and information retrieval.