Feature Extraction by Maximizing the Average Neighborhood Margin

Fei Wang, Changshui Zhang
State Key Laboratory of Intelligent Technologies and Systems
Department of Automation, Tsinghua University, Beijing, China. 100084.

Abstract

*A novel algorithm called Average Neighborhood Margin Maximization (ANMM) is proposed for supervised linear feature extraction. 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. We will show that features extracted from ANMM can separate the data from different classes well, and it avoids the small sample size problem existed in traditional Linear Discriminant Analysis (LDA). The kernelized (nonlinear) counterpart of ANMM is also established in this paper. Moreover, as in many computer vision applications the data are more naturally represented by higher order tensors (*e.g. *images and videos), we develop a tensorized (multilinear) form of ANMM, which can directly extract features from tensors. The experimental results of applying ANMM to face recognition are presented to show the effectiveness of our method.*

1. Introduction

Feature extraction (or dimensionality reduction) is an important research topic in computer vision and pattern recognition fields, since (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 [1]. In the past several decades, many feature extraction methods have been proposed, in which the most well-known ones are *Principal Component Analysis* (*PCA*) [10] and *Linear Discriminant Analysis* (*LDA*). However, there are still some limitations for directly applying them to solve vision problems.

Firstly, although *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), it does not take the class information into account and thus may not be reliable for classification tasks. On the contrary, *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 some drawbacks existed in *LDA* [13], such as (1) it usually suffers from the *small sample size* problem [18] 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 are all linear methods. However, it is discovered that many vision problems may not be linear [7][20], which makes these linear approaches inefficient. Fortunately, *kernel based methods* [2] can handle these nonlinear cases very well. The basic idea behind those kernel based techniques is to first map the data to a high-dimensional (usually infinite-dimensional) *feature space*, and make the nonlinear problem in the original space linearly solvable in the feature space. It has been shown that *Kernelized PCA* [3] and *Kernelized LDA* [19] can improve the performances of original *PCA* and *LDA* significantly in many computer vision and pattern recognition problems.

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 a 2nd-order tensor, *i.e.* matrix, and the sequential data, such as a video sequence for event analysis, is in the form of 3rd-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 considerable interests from the computer vision and pattern recognition community [5], 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 linear feature extraction method called *Average Neighborhood Mar-*

1-4244-1180-7/07/$25.00 ©2007 IEEE

gin Maximization (*ANMM*). For each data point, *ANMM* aims to *pull* the neighboring points with the same class label towards it as near as possible, while simultaneously *push* the neighboring points with different labels away from it as far as possible. Compared with traditional *LDA*, our method has the following advantages:

1. *ANMM* avoids the *small sample size* problem [18] since it does not need to compute any matrix inverse;

2. *ANMM* can find the discriminant directions without assuming the particular form of class densities;

3. Much more feature dimensions are available in *ANMM*, which is not limited to $c-1$ as in *LDA*.

Moreover, we also derive the nonlinear and multilinear forms of *ANMM* for handling the nonlinear and tensor data. Finally the experimental results on face recognition are presented to show the effectiveness of our method.

The rest of this paper is organized as follows. In section 2 we will briefly review some methods that are closely related to *ANMM*. The algorithm details of *ANMM* will be introduced in section 3. In section 4 and section 5 we will develop the kernelized and tensorized forms of *ANMM*. The experimental results on face recognition will be presented in section 6, followed by the conclusions and discussions in section 7.

2. Related Works

In this section we will briefly review some linear feature extraction methods that are closely related to *ANMM*. First let's see some notations and problem definition.

Let $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$ be the empirical dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th datum represented by a d dimensional column vector, and $y_i \in \mathcal{L}$ is the label of $\mathbf{x}_i$, $\mathcal{L} = \{1, 2, \cdots, c\}$ is the label set. The goal of linear feature extraction is to learn a $d \times l$ projection matrix $\mathbf{W}$, which can project $\mathbf{x}_i$ to

$$\mathbf{y}_i = \mathbf{W}^T \mathbf{x}_i,$$

where $\mathbf{y}_i \in \mathbb{R}^l$ is the projected data with $l \ll d$, such that in the projected space the data from different classes can be effectively discriminated.

Traditional LDA learns $\mathbf{W}$ by maximizing the following criterion

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

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 dataset. $\mathbf{S}_w = \sum_{k=1}^{c} p_k \mathbf{S}_k$ 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 [13]. However, when the size of the dataset is small, $\mathbf{S}_w$ will become singular. Then $\mathbf{S}_w^{-1}$ does not exist and the *small sample size* (*SSS*) problem occurs. Many approaches have been proposed to solve such a problem, such as *PCA+LDA* [18], *null space LDA* [14], *direct LDA* [9], etc. *Li et al.* [6] further proposed an efficient and robust linear feature extraction method which aims to maximize the following criterion which was called a *margin* in [6]

$$\mathcal{J} = tr\left(\mathbf{W}^T(\mathbf{S}_b - \mathbf{S}_w)\mathbf{W}\right), \tag{1}$$

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. *Qiu et al.* [23] proposed a *Nonparametric Margin Maximization Criterion* for learning $\mathbf{W}$, which tries to maximize

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

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. Moreover, the stepwise procedure for maximizing $\mathcal{J}$ is time consuming.

From another point of view linear feature extraction can also be treated as learning a proper *Mahalanobis distance* between pairwise points, since

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

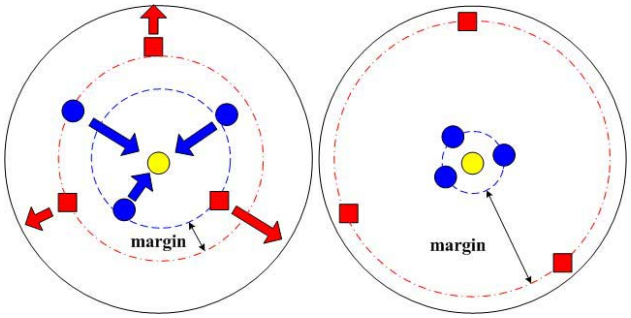
Let $\mathbf{M} = \mathbf{W}\mathbf{W}^T$, then

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

Weinberger et al. [15] 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 datasets.

3. Feature Extraction by Average Neighborhood Margin Maximization (ANMM)

In this section we will introduce our *Average Neighborhood Margin Maximization* (*ANMM*) 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. To achieve such a goal, we first introduce



(a) Neighborhood in the original space (b) Neighborhood in the projected space

Figure 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.

two types of neighborhoods:

Definition 1(*Homogeneous Neighborhoods*). 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 Neighborhoods*).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 $\mathbf{x}_i$.

Then the *average neighborhood margin* γ_i for $\mathbf{x}_i$ is defined as

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

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. The maximization of such a margin 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 *ANMM* criterion.

Therefore, the total *average neighborhood margin* can be defined as

$$\begin{aligned}\gamma &= \sum\nolimits_i \gamma_i \\ &= \sum_i \left(\sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{\|\mathbf{y}_i - \mathbf{y}_k\|^2}{|\mathcal{N}_i^e|} - \sum_{j:\mathbf{x}_j \in \mathcal{N}_i^o} \frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{|\mathcal{N}_i^o|} \right),\end{aligned}$$

and the *ANMM criterion* is to maximize γ.

Since

$$\begin{aligned}&\sum_i \sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{\|\mathbf{y}_i - \mathbf{y}_k\|^2}{|\mathcal{N}_i^e|} \\ &= tr\left(\sum_i \sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{(\mathbf{y}_i - \mathbf{y}_k)(\mathbf{y}_i - \mathbf{y}_k)^T}{|\mathcal{N}_i^e|} \right) \\ &= tr\left[\mathbf{W}^T \left(\sum_i \sum_{k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{(\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_k)^T}{|\mathcal{N}_i^e|} \right) \mathbf{W} \right] \\ &= \mathbf{W}^T tr(\mathbf{S}) \mathbf{W}, \end{aligned} \tag{3}$$

where the matrix

$$\mathbf{S} = \sum_{\substack{i,k: \\ \mathbf{x}_k \in \mathcal{N}_i^e}} \frac{(\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_k)^T}{|\mathcal{N}_i^e|}, \tag{4}$$

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

$$\mathbf{C} = \sum_{\substack{i,j: \\ \mathbf{x}_j \in \mathcal{N}_i^o}} \frac{(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T}{|\mathcal{N}_i^o|}. \tag{5}$$

Then

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

Therefore the average neighborhood margin can be rewritten as

$$\gamma = tr\left[\mathbf{W}^T (\mathbf{S} - \mathbf{C}) \mathbf{W}\right]. \tag{6}$$

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

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

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 our *ANMM criterion* becomes

$$\begin{aligned} \max \quad & \sum\nolimits_{k=1}^{l} \mathbf{w}_k^T (\mathbf{S} - \mathbf{C}) \mathbf{w}_k \\ s.t. \quad & \mathbf{w}_k^T \mathbf{w}_k = 1. \end{aligned} \tag{7}$$

[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.

Table 1. Average Neighborhood Margin Maximization

Input: Training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, Testing set $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M\}$, Neighborhood size $|\mathcal{N}^o|$, $|\mathcal{N}^e|$, Desired dimensionality l;
Output: $l \times M$ feature matrix $\mathbf{F}$ extracted from $\mathcal{Z}$.

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.(4) and Eq.(5) 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;
4. Output $\mathbf{F} = \mathbf{W}^T\mathbf{Z}$ with $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N]$.

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 $\mathbf{S} - \mathbf{C}$.

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

4. Nonlinearization via Kernelization

In this section, we will extend the *ANMM* algorithm to the nonlinear case via the kernel method [2]. More formally, we will first map the dataset from the original space $\mathbb{R}^d$ to a high (usually infinite) dimensional feature space $\mathcal{F}$ through a nonlinear mapping $\Phi : \mathbb{R}^d \longrightarrow \mathcal{F}$, and apply linear *ANMM* there.

In the feature space $\mathcal{F}$, 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

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

where

$$\begin{aligned}\mathbf{S}^\Phi &= \sum_{\substack{i,k:\\ \Phi(\mathbf{x}_k)\in\mathcal{N}^e_{\Phi(\mathbf{x}_i)}}} \frac{(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_k))\,(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_k))^T}{\left|\mathcal{N}^e_{\Phi(\mathbf{x}_i)}\right|} \\ \mathbf{C}^\Phi &= \sum_{\substack{i,j:\\ \Phi(\mathbf{x}_j)\in\mathcal{N}^o_{\Phi(\mathbf{x}_i)}}} \frac{(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))\,(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^T}{\left|\mathcal{N}^o_{\Phi(\mathbf{x}_i)}\right|},\end{aligned}$$

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 $\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_2), \cdots, \Phi(\mathbf{x}_N)$, *i.e.*

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

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

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

Define the matrices

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

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

then

$$\begin{aligned}\gamma^\Phi &= \sum_{k=1}^{l} \mathbf{w}_k^T(\mathbf{S}^\Phi - \mathbf{C}^\Phi)\mathbf{w}_k = \sum_{k=1}^{l} \left(\mathbf{w}_k\mathbf{S}^\Phi\mathbf{w}_k - \mathbf{w}_k\mathbf{C}^\Phi\mathbf{w}_k\right) \\ &= \sum_{k=1}^{l} (\boldsymbol{\alpha}^k)^T\left(\tilde{\mathbf{S}}^\Phi - \tilde{\mathbf{C}}^\Phi\right)\boldsymbol{\alpha}^k\end{aligned}$$

Similar to Eq.(7), we also add the constraints that $(\boldsymbol{\alpha}^k)^T(\boldsymbol{\alpha}^k) = 1\ (k = 1, 2, \cdots, l)$. 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

$$\mathbf{w}_k^T\Phi(\mathbf{z}) = \sum_{p=1}^{N} \alpha_p^k \Phi(\mathbf{x}_p)^T\Phi(\mathbf{z}) = (\boldsymbol{\alpha}^k)^T\mathbf{K}^t_{\cdot\mathbf{z}}. \tag{11}$$

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

The main procedure *Kernel Average Neighborhood Margin Maximization* (*KANMM*) 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 $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M\}$, Neighborhood size $|\mathcal{N}_\Phi^o|$, $|\mathcal{N}_\Phi^e|$, Kernel parameter $\boldsymbol{\theta}$, Desired dimensionality l;

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 $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_l\}$ corresponding to the largest l eigenvalues;
5. Construct the kernel matrix between the training set and the testing set $\mathbf{K}^t$ with its (i,j)-th entry $\mathbf{K}^t_{ij} = \Phi(\mathbf{x}_i)^T\Phi(\mathbf{z}_j)$.
6. Output $\mathbf{F}^\Phi$ with $\mathbf{F}^\Phi_{ij} = (\boldsymbol{\alpha}^i)^T\mathbf{K}^t_{\cdot j}$.

5. 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's 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 $A_{i_1 i_2 \cdots i_K}$.

Definition 3 (*Scalar Product*). The *scalar product* $\langle A, B\rangle$ of two tensors $A, B \in \mathbb{R}^{d_1 \times d_2 \times \cdots d_K}$ is defined as

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

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

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

Definition 4 (f-*Mode Product*). The f-*mode product* of a tensor $A \in \mathbb{R}^{d_1 \times d_2 \times \cdots d_K}$ 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

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

Definition 5 (f-*Mode Unfolding*). Let A be a $d_1 \times \cdots \times d_K$ tensor and $(\pi_1, \cdots, \pi_{K-1})$be any permutation of the entries of the set $\{1, \cdots, f-1, f+1, \cdots, K\}$. 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 $\mathbf{A}^{(f)}_{i_f j} = A_{i_1 \cdots i_K}$ 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 $X_1, \cdots, X_N$ 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 $\mathbf{U}_i \in \mathbb{R}^{l_i \times d_i}$ $(l_i < d_i,\ i = 1, 2, \cdots, K)$, 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 iteratie scheme which was originally proposed by [12] for low-rank approximation of second-order tensors. Later [8] extended it for higher-order tensors. In the following we will adopt such an iterative scheme to solve the optimization problem.

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

$$Y_i^f = X_i \times_1 \mathbf{U}_1 \cdots \times_{f-1} \mathbf{U}_{f-1} \times_{f+1} \mathbf{U}_{f+1} \cdots \times_K \mathbf{U}_K. \tag{12}$$

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

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

Therefore we have

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

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

$$\mathbf{S} = \sum_{\substack{i,k: \\ \mathbf{x}_k \in \mathcal{N}_i^e}} \frac{\left(\mathbf{Y}_i^{(f)} - \mathbf{Y}_k^{(f)}\right)\left(\mathbf{Y}_i^{(f)} - \mathbf{Y}_k^{(f)}\right)^T}{|\mathcal{N}_i^e|}, \tag{13}$$

$$\mathbf{C} = \sum_{\substack{i,j: \\ \mathbf{x}_k \in \mathcal{N}_i^o}} \frac{\left(\mathbf{Y}_i^{(f)} - \mathbf{Y}_j^{(f)}\right)\left(\mathbf{Y}_i^{(f)} - \mathbf{Y}_j^{(f)}\right)^T}{|\mathcal{N}_i^o|}, \tag{14}$$

Table 3. Tensor Average Neighborhood Margin Maximization

Input: Training set $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^{N}$, Testing set $\mathcal{Z} = \{Z_1, Z_2, \cdots, Z_M\}$, where $X_i, Z_j \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_K}$, Neighborhood size $|\mathcal{N}^o|$, $|\mathcal{N}^e|$, Desired dimensionality $l_1, l_2, \cdots, l_K$, Iteration steps T_{max}, Difference ε;

Output: Feature tensors $\{F_i\}_{i=1}^{M}$ extracted from $\mathcal{Z}$, where $F_i \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_K}$.

1. Initialize $\mathbf{U}_1^0 = \mathbf{I}_{d_1}$, $\mathbf{U}_2^0 = \mathbf{I}_{d_2}$, $\cdots$, $\mathbf{U}_K^0 = \mathbf{I}_{d_K}$, where $\mathbf{I}_{d_i}$ represents the $d_i \times d_i$ identity matrix;
2. For $t = 1, 2, \cdots, T_{max}$ do
   For $f = 1, 2, \cdots, K$ do
   (a). Compute Y_i^f by Eq.(12);
   (b). $Y_i^f \Rightarrow_f \mathbf{Y}_i^{(f)}$;
   (c). Compute $\mathbf{S}$ and $\mathbf{C}$ using Eq.(13) and Eq.(14);
   (d). Do eigenvalue decomposition on $\mathbf{S} - \mathbf{C}$: $(\mathbf{S} - \mathbf{C})\mathbf{U}_f^t = \mathbf{U}_f^t \Lambda_f$ with $\mathbf{U}_f^t \in \mathbb{R}^{d_f \times l_f}$;
   (f). if $\|\mathbf{U}_f^t - \mathbf{U}_f^{t-1}\| < \varepsilon$, break;
   End for.
   End for.
3. Output $F_i = Z_i \times_1 \mathbf{U}_1^t \cdots \times_K \mathbf{U}_K^t$.

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

$$\max_{\mathbf{U}_f} tr\left[\mathbf{U}_f^T (\mathbf{S} - \mathbf{C}) \mathbf{U}_f\right] \tag{15}$$

Let's expand $\mathbf{U}_f$ as $\mathbf{U}_f = (\mathbf{u}_{f1}, \mathbf{u}_{f2}, \cdots, \mathbf{u}_{fl_f})$ with $\mathbf{u}_{fi}$ corresponding to the i-th column of $\mathbf{U}_f$, then Eq.(15) can be rewritten as

$$\max \sum_{i=1}^{l_f} \mathbf{u}_{fi}^T (\mathbf{S} - \mathbf{C}) \mathbf{u}_{fi}. \tag{16}$$

We also add the constraint that $\mathbf{u}_{fi}^T \mathbf{u}_{fi} = 1$ to restrict the scale of $\mathbf{U}_f$. The main procedure of the *Tensor Average Neighborhood Margin Maximization* (*TANMM*) is summarized in Table 3.

6. Experiments

In this section, we investigate the performance of our proposed *ANMM*, *Kernel ANMM* (*KANMM*) and *Tensor ANMM* (*TANMM*) 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* [16] method, *LDA* (*PCA+LDA*) method [18], and three *margin* based methods, namely the *Maximum Margin Criterion* (*MMC*) method [6], the *Stepwise Nonparametric Maximum MArgin Criterion* (*SNMMC*) method [23] and the *Marginal Fisher Analysis* (*MFA*) method [21];
2. *Kernel methods*. In this set of experiments, the performance of the *KANMM* method is compared with the *KPCA* and the *KDA* method [17];
3. *Tensor methods*. In this set of experiments, the performance of the *Tensor ANMM* (*TANMM*) method is compared with the *Tensor PCA* (*TPCA*) and the *Tensor LDA* (*TLDA*) methods [4].

In this study, three face dataset are used:

1. The *ORL* face dataset[2]. There are ten 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 degrees. The original images (with 256 gray levels) have size 92×112, which are resized to 32×32 for efficiency;
2. The *Yale* face dataset[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 dataset [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.

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 to 10;
2. For the *kernel methods*,we all adopt the Gaussian kernel, and the variance of the Gaussian kernel were set by cross-validation;
3. For the *tensor methods*, we require that the projected images are also square, *i.e.* of dimension $r \times r$ for some r.

[2] http://www.uk.research.att.com/facedatabase.html
[3] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

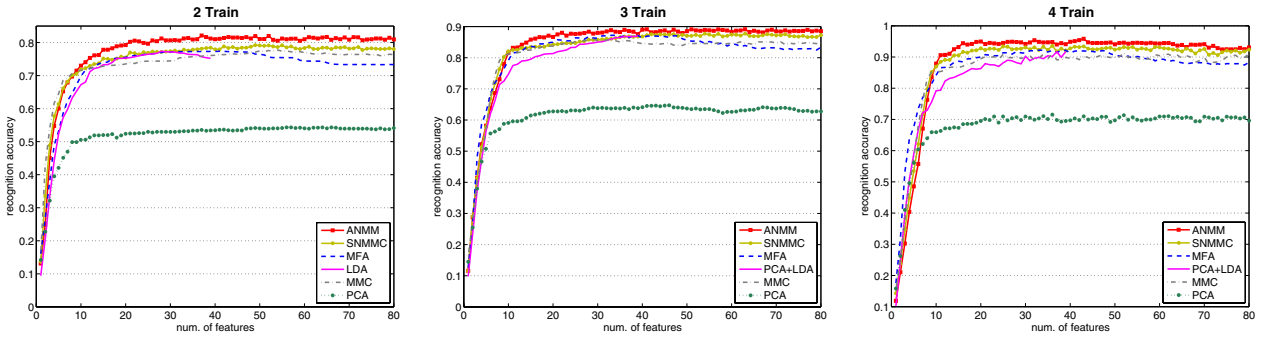


Figure 2. Face recognition accuracies on the ORL dataset with 2,3,4 images for each individual randomly selected for training.

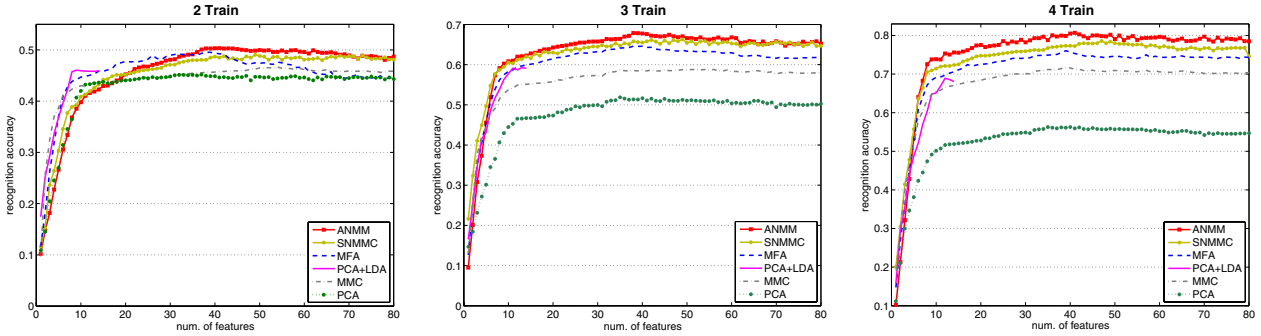


Figure 3. Face recognition accuracies on the Yale dataset with 2,3,4 images per individual randomly selected for training.

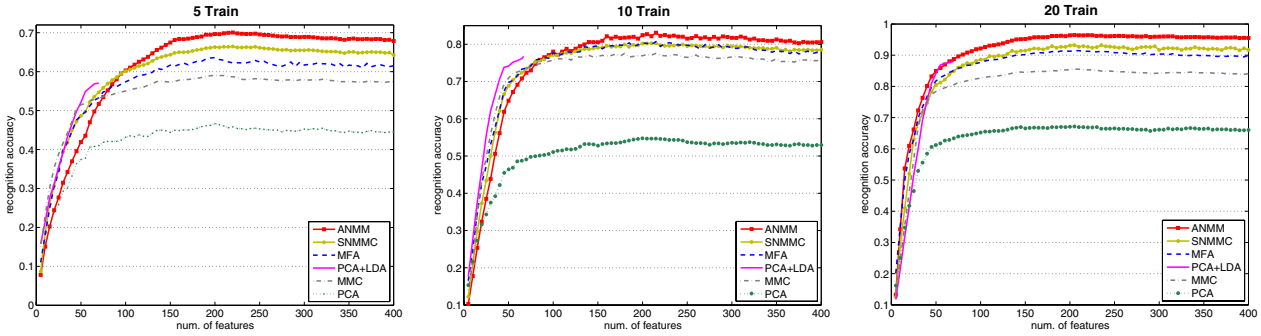


Figure 4. Face recognition accuracies on the CMU PIE dataset with 5,10,20 images per individual randomly selected for training.

The experimental results of the *linear methods* on the three datasets are shown in Fig.2, Fig.3, Fig.4 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 clearly see that the performances of *ANMM* is better than other linear methods on all the three datasets.

Table 4 shows the experimental results of all the methods on three datasets, 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 datasets.

7. Conclusions and Discussions

In this paper we proposed a novel supervised linear feature extraction method named *Average Neighborhood Margin Maximization* (*ANMM*). 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 recogntion 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.

Table 4. Face recognition results on three datasets (%).

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)**

As we mentioned in section 2, 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[15].

References

[1] A. K. Jain, B. Chandrasekaran. Dimensionality and Sample Size Considerations in Pattern Recognition Practice. In *Handbook of Statistics*. Amsterdam, North Holland. 1982. 1

[2] B. Schölkopf, A. Smola. *Learning with Kernels*. The MIT Press. Cambridge, Massachusetts. London, England. 2002. 1, 4

[3] B. Schölkopf, A. Smola, K.-R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299-1319. 1998. 1

[4] D. Cai, X. He, J. Han. Subspace Learning Based on Tensor Analysis. *Department of Computer Science Technical Report No. 2572, University of Illinois at Urbana-Champaign (UIUCDCS-R-2005-2572)*. 2005. 6

[5] Fernando De la Torre, M. Alex O. Vasilescu. Linear and Multilinear (Tensor) Methods for Vision, Graphics, and Signal Processing. *IEEE CVPR Tutorial*. 2006. 1

[6] H. Li, T. Jiang, K. Zhang. Efficient and Robust Feature Extraction by Maximum Margin Criterion. In *NIPS 16*. 2004. 2, 6

[7] H. S. Seung, D. D. Lee. The manifold ways of perception. *Science*, 290. 2000. 1

[8] 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. 5

[9] H. Yu, J. Yang. A Direct LDA Algorithm for High Dimensional Data with Application to Face Recognition. *Pattern Recognition*. 2001. 2

[10] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York. 1986. 1

[11] J. Yang, D. Zhang, Alejandro F. Frangi, J. Yang. Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. *IEEE TPAMI*. 2004.

[12] J. Ye. Generalized Low Rank Approximations of Matrices. In *Proceedings of ICML*. 2004. 5

[13] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 2nd edtion. 1990. 1, 2

[14] K. Liu, Y. Cheng, J. Yang. A Generalized Optimal Set of Discriminant Vectors. *Pattern Recognition*. 1992. 2

[15] K. Q. Weinberger, J. Blitzer, L. K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification In *NIPS 18*. 2006. 2, 8

[16] M. A. Turk and A. P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1): 71-96, 1991. 6

[17] 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. 6

[18] P.N. Belhumeur, J. Hespanda, D. Kiregeman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Trans. on PAMI*. 1997. 1, 2, 6

[19] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, K.-R. Müller. Fisher Discriminant Analysis with Kernels. *Neural Networks for Signal Processing IX*, IEEE. 1999. 1

[20] S. T. Roweis, L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290. 2000. 1

[21] S. Yan, D. Xu, B. Zhang and H. Zhang. Graph Embedding: A General Framework for Dimensionality Reduction. In *Proceedings of IEEE CVPR*. 2005. 6

[22] T. Sim, S. Baker, and M. Bsat. The CMU pose, illuminlation, and expression database. *IEEE Trans. on PAMI*. 2003. 6

[23] X. Qiu, L. Wu. Face Recognition by Stepwise Nonparametric Margin Maximum Criterion. In *Proc. ICCV*. 2005. 2, 6