Learning a Spatially Smooth Subspace for Face Recognition

Deng Cai[†] Xiaofei He[‡] Yuxiao Hu[†] Jiawei Han[†] Thomas Huang[†]

[†]University of Illinois at Urbana-Champaign
[‡]Yahoo! Research Labs
{dengcai2, hanj}@cs.uiuc.edu, hex@yahoo-inc.com, {hu3, huang}@ifp.uiuc.edu

Abstract

Subspace learning based face recognition methods have attracted considerable interests in recently years, including Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Locality Preserving Projection (LPP), Neighborhood Preserving Embedding (NPE), Marginal Fisher Analysis (MFA) and Local Discriminant Embedding (LDE). These methods consider an $n_1 \times n_2$ image as a vector in $\mathbb{R}^{n_1 \times n_2}$ and the pixels of each image are considered as independent. While an image represented in the plane is intrinsically a matrix. The pixels spatially close to each other may be correlated. Even though we have $n_1 \times n_2$ pixels per image, this spatial correlation suggests the real number of freedom is far less. In this paper, we introduce a regularized subspace learning model using a Laplacian penalty to constrain the coefficients to be spatially smooth. All these existing subspace learning algorithms can fit into this model and produce a spatially smooth subspace which is better for image representation than their original version. Recognition, clustering and retrieval can be then performed in the image subspace. Experimental results on face recognition demonstrate the effectiveness of our method.

1. Introduction

Many face recognition techniques have been developed over the past few decades. One of the most successful and well-studied techniques to face recognition is the appearance-based method [20, 24]. When using appearance-based methods, we usually represent an image of size $n_1 \times n_2$ pixels by a vector in an $n_1 \times n_2$-dimensional space. In practice, however, this $n_1 \times n_2$-dimensional space is too large to allow robust and fast face recognition. Previous works have demonstrated that the face recognition performance can be improved significantly in lower dimensional linear subspaces [1, 6, 16, 24]. Two of the most popular appearance-based face recognition methods are *Eigenface* [24] and *Fisherface* [1]. Eigenface is based on Principal Component Analysis (PCA) [9]. PCA projects the face images along the directions of maximal variances and aims to preserve the Euclidean distances between face images. Fisherface is based on Linear Discriminant Analysis (LDA) [9]. Unlike PCA which is unsupervised, LDA is supervised. When the class information is available, LDA can be used to find a linear subspace which is optimal for discrimination.

Recently there are considerable interest in geometrically motivated approaches to visual analysis. Various researchers (see [2], [22], [23]) have considered the case when the data lives on or close to a low dimensional submanifold of the high dimensional ambient space. One hopes then to estimate geometrical and topological properties of the submanifold from random points ("scattered data") lying on this unknown submanifold. Along this direction, many subspace learning algorithms have been proposed for face recognition. Some popular ones include Locality Preserving Projection (LPP) [15], Neighborhood Preserving Embedding (NPE) [14], Marginal Fisher Analysis (MFA) [26] and Local Discriminant Embedding (LDE) [7].

All the above methods consider a face image as a high dimensional vector. They do not take advantage of the spatial correlation of pixels in the image, and the pixels are considered as independent pieces of information. However, a $n_1 \times n_2$ face image represented in the plane is intrinsically a matrix, or 2-order tensor. Even though we have $n_1 \times n_2$ pixels per image, this spatial correlation suggests the real number of freedom is far less. Recently there have been a lot of interest in tensor based approaches to data analysis in high dimensional spaces. Vasilescu and Terzopoulos have proposed a novel face representation algorithm called Tensorface [25]. Tensorface represents the set of face images by a higher-order tensor and extends Singular Value Decomposition (SVD) to higher-order tensor data. Some other researchers have also shown how to extend PCA, LDA, LPP, MFA and LDE to higher order tensor data [5, 7, 13, 26, 27]. Some experimental results have showed the superiority of these tensor approaches over their corresponding vector approaches. However, our analysis later will show that these tensor approaches only consider the relationship between pixels in the same row (column) and fail to fully explorer the spatial information of images. The embedding functions of tensor approaches will still be spatially rough.

In this paper, we introduce a Spatially Smooth Subspace Learning (SSSL) model using a Laplacian penalty to constrain the coefficients to be spatially smooth. Instead of considering the basis function as a $n_1 \times n_2$-dimensional vector, we consider it as a matrix, or a discrete function defined on a $n_1 \times n_2$ lattice. Thus, the discretized Laplacian can be applied to the basis functions to measure their smoothness along horizontal and vertical directions. The discretized Laplacian operator is a finite difference approximation to the second derivative operator, summed over all directions. The choice of Laplacian penalty allows us to incorporate the prior information that neighboring pixels are correlated. Once we obtain compact representations of the images, classification and clustering can be performed in the lower dimensional subspace.

The points below highlight several aspects of the paper:

1. When the number-of-dimensions to sample-size ratio is too high, it is difficult for those subspace learning algorithms (LDA, LPP, NPE, *etc.*) to discover the intrinsic discriminant or geometrical structure. Since the image data generally has a large number of dimensions (pixels), the methods of regularization are needed.

2. Even if the sample size were sufficient to estimate the intrinsic geometrical structure, coefficients of spatially smooth features (pixels) tend to be spatially rough. Since we hope to interpret these coefficients, we would prefer smoother versions, especially if they do not compromise the fit.

The remainder of the paper is organized as follows. In Section 2, we provide a brief review of the various subspace learning algorithms and their tensor extensions. Section 3 introduces our proposed Spatially Smooth Subspace Learning (SSSL) model. The extensive experimental results are presented in Section 4. Finally, we provide some concluding remarks and suggestions for future work in Section 5.

2. LDA, LPP, NPE, MFA, LDE and Their Tensor Extensions

Suppose we have m $n_1 \times n_2$ face images. Let $\{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^n$ ($n = n_1 \times n_2$) denote their vector representations and $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m]$. In the past decades, many dimensionality reduction algorithms have been proposed to find a low dimensional representation of $\mathbf{x}_i$. Despite the different motivations of these algorithms, they can be nicely interpreted in a general *graph embedding* framework [3, 16, 26].

Given a graph G with m vertices, each vertex represents a data point. Let W be a symmetric $m \times m$ matrix with W_{ij} having the weight of the edge joining vertices i and j. The G and W can be defined to characterize certain statistical or geometric properties of the data set. The purpose of graph embedding is to represent each vertex of the graph as a low dimensional vector that preserves similarities between the vertex pairs, where similarity is measured by the edge weight.

Let $\mathbf{y} = [y_1, y_2, \cdots, y_m]^T$ be the map from the graph to the real line. The optimal $\mathbf{y}$ is given by minimizing

$$\sum_{i,j}(y_i - y_j)^2 W_{ij}$$

under appropriate constraint. This objective function incurs a heavy penalty if neighboring vertices i and j are mapped far apart. Therefore, minimizing it is an attempt to ensure that if vertices i and j are "close" then y_i and y_j are close as well [10]. With some simple algebraic formulations, we have

$$\sum_{i,j}(y_i - y_j)^2 W_{ij} = 2\mathbf{y}^T L\mathbf{y},$$

where $L = D - W$ is the *graph Laplacian* [8] and D is a diagonal matrix whose entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ji}$. Finally, the minimization problem reduces to find

$$\mathbf{y}^* = \underset{\mathbf{y}^T D\mathbf{y}=1}{\arg\min}\, \mathbf{y}^T L\mathbf{y} = \arg\min \frac{\mathbf{y}^T L\mathbf{y}}{\mathbf{y}^T D\mathbf{y}}.$$

The constraint $\mathbf{y}^T D\mathbf{y} = 1$ removes an arbitrary scaling factor in the embedding. Notice that $L = D - W$, it is easy to see that the above optimization problem has the following equivalent variation:

$$\mathbf{y}^* = \underset{\mathbf{y}^T D\mathbf{y}=1}{\arg\max}\, \mathbf{y}^T W\mathbf{y} = \arg\max \frac{\mathbf{y}^T W\mathbf{y}}{\mathbf{y}^T D\mathbf{y}}. \quad (1)$$

The optimal $\mathbf{y}$'s can be obtained by solving the maximum eigenvalue eigen-problem:

$$W\mathbf{y} = \lambda D\mathbf{y}. \quad (2)$$

Many recently proposed manifold learning algorithms, like ISOAMP [23], Laplacian Eigenmap [2], Locally Linear Embedding [22], can be interpreted in this framework with different choice of W. The two matrices W and D play the essential role in this graph embedding approach. The choices of these two graph matrices can be very flexible. In later discussion, we use GE(W, D) to denote the graph embedding with maximization problem of $\max(\mathbf{y}^T W\mathbf{y})/(\mathbf{y}^T D\mathbf{y})$.

The graph embedding approach described above only provides the mappings for the graph vertices in the training set. For classification purpose (*e.g.*, face recognition), a mapping for all samples, including new test samples, is required. If we choose a linear function, *i.e.*, $y_i = f(\mathbf{x}_i) = \mathbf{a}^T\mathbf{x}_i$, we have $\mathbf{y} = X^T\mathbf{a}$. Eq. (1) can be rewritten as:

$$\mathbf{a}^* = \arg\max \frac{\mathbf{y}^T W \mathbf{y}}{\mathbf{y}^T D \mathbf{y}} = \arg\max \frac{\mathbf{a}^T X W X^T \mathbf{a}}{\mathbf{a}^T X D X^T \mathbf{a}}. \quad (3)$$

The optimal $\mathbf{a}$'s are the eigenvectors corresponding to the maximum eigenvalue of eigen-problem:

$$XWX^T\mathbf{a} = \lambda XDX^T\mathbf{a}. \quad (4)$$

This approach is called *Linear extension of Graph Embedding* (LGE). With different choices of W and D, the LGE framework will lead to many popular linear dimensionality reduction algorithms, *e.g.*, LDA, LPP, NPE, MFA and LDE. We will briefly list the choices of W and D for these algorithms as follows.

LDA:

Suppose we have c classes and the t-th class have m_t samples, $m_1 + \cdots + m_c = m$. Define

$$W_{ij}^{LDA} = \begin{cases} 1/m_t, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong to the } t\text{-th class;} \\ 0, & \text{otherwise.} \end{cases}$$

With this W^{LDA}, it is easy to check that $D^{LDA} = I$. Please see [16] for the detailed derivation. LDA is the linear extension of graph embedding problem GE(W^{LDA}, I).

LPP:

Let $N_k(\mathbf{x}_i)$ denote the set of k nearest neighbors of $\mathbf{x}_i$.

$$W_{ij}^{LPP} = \begin{cases} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}, & \text{if } \mathbf{x}_i \in N_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_k(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases}$$

For supervised case, one can also integrate the label information into W^{LPP} by searching the k nearest neighbors of $\mathbf{x}_i$ among the points sharing the same label with $\mathbf{x}_i$. Please see [16] for the details. LPP is the linear extension of graph embedding problem GE(W^{LPP}, D^{LPP}).

NPE:

Let M be a $m \times m$ local reconstruction coefficient matrix which is defined as follows:

For i-th row of M, $M_{ij} = 0$ if $\mathbf{x}_j \notin N_k(\mathbf{x}_i)$. The other M_{ij} can be computed by minimizing the following objective function,

$$\min \|\mathbf{x}_i - \sum_{j \in N_k(\mathbf{x}_i)} M_{ij}\mathbf{x}_j\|^2, \quad \sum_{j \in N_k(\mathbf{x}_i)} M_{ij} = 1.$$

Define

$$W^{NPE} = M + M^T - M^T M.$$

It is easy to check that $D^{NPE} = I$. Please see [14, 26] the for detailed derivation. NPE is the linear extension of graph embedding problem GE(W^{NPE}, I).

MFA and LDE:

Local Discriminant Embedding (LDE, [7]) and Marginal Fisher Analysis (MFA, [26]) are essentially the same. Both of them uses two graphs to model the geometric and discriminant structure in the data.

Let $N_k^+(\mathbf{x}_i)$ denote the set of k nearest neighbors of $\mathbf{x}_i$ which share the same label with $\mathbf{x}_i$. And let $N_k^-(\mathbf{x}_i)$ denote the set of k nearest neighbors of $\mathbf{x}_i$ among the data points whose labels are different to that of $\mathbf{x}_i$. Define

$$W_{ij}^{MFA} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N_{k_1}^+(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_{k_1}^+(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

$$W_{ij}^{-} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N_{k_2}^-(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_{k_2}^-(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Let D^- be the diagonal matrix whose entries are column sums of W^- and $L^- = D^- - W^-$. The objective function of MFA (LDE) is

$$\mathbf{a}^* = \arg\min \frac{\mathbf{a}^T X L^{MFA} X^T \mathbf{a}}{\mathbf{a}^T X L^- X^T \mathbf{a}} = \arg\max \frac{\mathbf{a}^T X L^- X^T \mathbf{a}}{\mathbf{a}^T X L^{MFA} X^T \mathbf{a}}$$

It is clear that MFA (LDE) is the linear extension of graph embedding problem GE(L^-, L^{MFA}).

2.1. Tensor Extensions

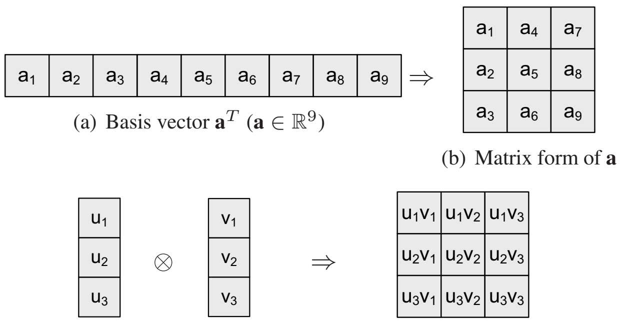
A face image represented in the plane is intrinsically a matrix, or the second order tensor. The relationship between nearby pixels of the image might be important for finding a projection. Recently there have been a lot of interest in extending the ordinary vector-based subspace learning approaches to tensor space [5, 7, 13, 26, 27].

The tensor-based approaches directly operate on the matrix representation of image data and are believed can capture the spatial relationship between the pixels. To examine what kind of spatial relationship has been captured in these tensor-based approaches, we need to examine the basis function.

Let $\{\mathbf{u}_k\}_{k=1}^{n_1}$ be an orthonormal basis of $\mathcal{R}^{n_1}$ and $\{\mathbf{v}_l\}_{l=1}^{n_2}$ be an orthonormal basis of $\mathcal{R}^{n_2}$. It can be shown that $\{\mathbf{u}_i \otimes \mathbf{v}_j\}$ forms a basis of the tensor space $\mathcal{R}^{n_1} \otimes \mathcal{R}^{n_2}$ [19]. Specifically, the projection of $T \in \mathcal{R}^{n_1} \otimes \mathcal{R}^{n_2}$ on the basis $\mathbf{u}_i \otimes \mathbf{v}_j$ can be computed as their inner product:

$$< T, \mathbf{u}_i \otimes \mathbf{v}_j >=< T, \mathbf{u}_i\mathbf{v}_j^T >= \mathbf{u}_i^T T \mathbf{v}_j$$

The ordinary vector-based approaches are linear, *i.e.*, $y_i = \mathbf{a}^T\mathbf{x}_i$ where $\mathbf{x}_i \in \mathbb{R}^n$ is the *vector* representation of the i-th image, $\mathbf{a}$ is the projection vector (basis vector) and y_i is the one-dimensional embedding on this basis. The n values in basis function $\mathbf{a}$ are independently estimated. The tensor-based approaches are multilinear, *i.e.*, $y_i = \mathbf{u}^T T_i \mathbf{v}$, where



(a) Basis vector $\mathbf{a}^T$ ($\mathbf{a} \in \mathbb{R}^9$)

(b) Matrix form of **a**

(c) Basis vector $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$

(d) $\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^T$, Basis of tensor space $\mathcal{R}^3 \otimes \mathcal{R}^3$

Figure 1. Take face images of size 3×3. The ordinary vector-based subspace learning algorithms (*e.g.* PCA and LDA) first convert the face images to 9-dimensional vectors and compute the basis vectors (projection functions). The basis vector is also 9-dimensional, as shown in (a). (b) The basis vector can be converted to the matrix form and shown as an image, which was referred as Eigenface (PCA) and Fisherface (LDA). The 9 numbers in the basis vector are independent estimated and there is no spatial relation between them. (c) The tensor-based subspace learning approaches directly take 3×3 face images as input and compute a set of 3-dimensional basis vectors **u**'s and **v**'s. (d) Each **u** and **v** form a basis $\mathbf{u} \otimes \mathbf{v}$ in tensor space which can also be shown as an image. The 9 numbers in the tensor basis only have 6 degrees of freedom and the values in the same row (column) have a common divisor. However, there is no guarantee of the spatial smoothness of the basis function.

$T_i \in \mathcal{R}^{n_1} \otimes \mathcal{R}^{n_2}$ is the *matrix* representation of the i-th image and $n = n_1 \times n_2$. The n values in a tensor basis $\mathbf{u}\mathbf{v}^T$ only have $n_1 + n_2$ degrees of freedom. In fact, the tensor-based approaches can be thought of as special cases of vector-based approaches with the following constraint:

$$a_{i+n_1(j-1)} = u_i v_j \tag{7}$$

where a_i, u_i and v_i are the i-th elements in **a**, **u** and **v** respectively.

Figure (1) gives a intuitive example. It is easy to see that there is a common divisor of the values belong to the same row (or column) in a tensor basis, which exactly the spatial relation captured by the tensor-based approaches. Intuitively, the spatial correlation of pixels in a face image would suggest the spatial smoothness of the basis function, *i.e.*, the element values in basis function would be similar if the elements are spatially near. However, the tensor-based approaches have no guarantee on this and the basis function could still be spatially rough.

A more natural measurement of spatial smoothness of basis function could be the sum of the squared differences between nearby elements. In the next section, we will show how to achieve this by incorporating a 2-D discretized laplacian smoothing term in ordinary vector-based approaches.

3. Spatially Smooth Subspace Learning

In this section, we describe how to apply Laplacian penalized functional to measure the smoothness of the basis vectors of the face space, which plays the key role in our Spatially Smooth Subspace Learning (SSSL) approach . We begin with a general description of Laplacian smoothing.

3.1. Laplacian Smoothing

Let f be a function defined on a region of interest, $\Omega \subset \mathbb{R}^d$. The Laplacian operator $\mathcal{L}$ is defined as follows [18]:

$$\mathcal{L}f(\mathbf{t}) = \sum_{j=1}^{d} \frac{\partial^2 f}{\partial t_j^2} \tag{8}$$

The Laplacian penalty functional, denoted by $\mathcal{J}$, is defined by:

$$\mathcal{J}(f) = \int_{\Omega} \left[\mathcal{L}f\right]^2 d\mathbf{t} \tag{9}$$

Intuitively, $\mathcal{J}(f)$ measures the smoothness of the function f over the region Ω. In this paper, our primary interest is in image. An image is intrinsically a two-dimensional signal. Therefore, we take d to be 2 in the following.

3.2. Discretized Laplacian Smoothing

As we described previously, $n_1 \times n_2$ face images can be represented as vectors in $\mathbb{R}^n, n = n_1 \times n_2$. Let $\mathbf{a}_i \in \mathbb{R}^n$ be the basis vectors (projection functions) obtained by LPP. Without loss of generality, $\mathbf{a}_i$ can also be considered as functions defined on a $n_1 \times n_2$ lattice.

For a face image, the region of interest Ω is a two-dimensional rectangle, which for notational convenience we take to be $[0, 1]^2$. A lattice is defined on Ω as follows. Let $\mathbf{h} = (h_1, h_2)$ where $h_1 = 1/n_1$ and $h_2 = 1/n_2$. Ω_h consists of the set of two-dimensional vectors $\mathbf{t}_i = (t_{i_1}, t_{i_2})$ with $t_{i_j} = (i_j - 0.5) \cdot h_j$ for $1 \le i_j \le n_j$ and $1 \le j \le 2$. There are a total of $n = n_1 \times n_2$ grid points in this lattice. Let D_j be an $n_j \times n_j$ matrix that yields a discrete approximation to $\partial^2/\partial t_j^2$. Thus if $\mathbf{u} = (u(t_1), \cdots, u(t_{n_j}))$ is an n_j-dimensional vector which is a discretized version of a function $u(t)$, then D_j has the property that:

$$[D_j \mathbf{u}]_i \approx \frac{\partial^2 u(t_i)}{\partial t^2}$$

for $i = 1, \cdots, n_j$. There are many possible choices of D_j [4]. In this work, we apply the modified Neuman discretization [21]:

$$D_j = \frac{1}{h_j^2}\begin{pmatrix} -1 & 1 & & & & & 0 \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ 0 & & & & & 1 & -1 \end{pmatrix}$$

Give D_j, a discrete approximation for two-dimensional Laplacian $\mathcal{L}$ is the $n \times n$ matrix:

$$\Delta = D_1 \otimes I_2 + I_1 \otimes D_2 \quad (10)$$

where I_j is $n_j \times n_j$ identity matrix for $j = 1, 2$. $\otimes$ is the kronecker product [17].

For a $n_1 \times n_2$ dimensional vector $\mathbf{a}$, it is easy to check that $\|\Delta \cdot \mathbf{a}\|^2$ is proportional to the the sum of the squared differences between nearby grid points of $\mathbf{a}$ with its matrix form. It provides a measure of smoothness of $\mathbf{a}$ on the $n_1 \times n_2$ lattice.

3.3. The Algorithm

Given a pre-defined graph structure with weight matrix W, the SSSL approach is defined as the maximizer of

$$\frac{\mathbf{a}^T XWX^T\mathbf{a}}{(1-\alpha)\mathbf{a}^T XDX^T\mathbf{a} + \alpha\mathcal{J}(\mathbf{a})}, \quad (11)$$

where $\mathcal{J}$ is the discretized Laplacian regularization functional:

$$\mathcal{J}(\mathbf{a}) = \|\Delta \cdot \mathbf{a}\|^2 = \mathbf{a}^T \Delta^T \Delta \mathbf{a}. \quad (12)$$

The parameter $0 \leq \alpha \leq 1$ controls the smoothness of the estimator.

The vectors $\mathbf{a}_i$ $(i = 1, \cdots, l)$ that maximize the objective function (11) are given by the maximum eigenvalue solutions to the following generalized eigenvalue problem.

$$XWX^T\mathbf{a} = \lambda\left((1-\alpha)XDX^T + \alpha\Delta^T\Delta\right)\mathbf{a}. \quad (13)$$

With the choices of different W as described in Section 2, our approach gives the spatially smooth version of LDA, LPP and NPE.

There is a slight difference for spatially smooth MFA (LDE). Based on the objective function of MFA (LDE) in Eq. (2), the objective function of spatially smooth MFA (LDE) is:

$$\arg\max \frac{\mathbf{a}^T XL^- X^T\mathbf{a}}{(1-\alpha)\mathbf{a}^T XLX^T\mathbf{a} + \alpha\mathcal{J}(\mathbf{a})}, \quad (14)$$

which leads to the maximum eigenvalue problem

$$XL^-X^T\mathbf{a} = \lambda\left((1-\alpha)XLX^T + \alpha\Delta^T\Delta\right)\mathbf{a}. \quad (15)$$

It would be important to note that the 2-D discretized laplacian smoothing term presented here has been used in Penalized Discriminant Analysis for handwritten digit recognition [11].

3.4. Model Selection

The $\alpha \in [0, 1]$ is an essential parameter in SSSL model which controls the smoothness of the estimator. When $\alpha = 0$, the SSSL model will reduce to the ordinary subspace learning approach which totally ignores the spatial relationship between pixels of an image. When $\alpha \to \infty$, the SSSL model will choose a spatially smoothest basis vector $\mathbf{a}$ and totally ignore the manifold structure of the face data. SSSL with a suitable α is a trade-off between these two extreme cases. Thus, a natural question would be how to choose the parameter α, or how to select the model.

Model selection is an essential task in most of the learning algorithms [12]. Among various kinds of methods, cross validation is probably the simplest and most widely used one. In this paper, we also use cross validation for model selection.

Table 1. Compared algorithms

Objective function	Ordinary version	Tensor extension	Smooth version
PCA	Eigenface [24]	TensorPCA [5]	–
LDA	Fisherface [1]	2DLDA [27]	S-LDA
LPP	Laplacianface [16]	TSA [13]	S-LPP
NPE	NPE [14]	TNPE	S-NPE
MFA (LDE)	MFA [26]	TMFA [26]	S-MFA

4. Experimental Results

In this section, we investigate the performance of our proposed Spatially Smooth Subspace Learning approach for face recognition. The face recognition task is handled as a multi-class classification problem – we map each test image to a low-dimensional subspace via the embedding learned from training data, and then classify the test data by the nearest neighbor criterion.

4.1. Datasets and Compared Algorithms

The Yale and AT&T face databases are used in our experiments. The Yale face database[1] contains 165 gray scale images of 15 individuals, each individual has 11 images. The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised, and wink).

The AT&T face database[2] consists of a total of 400 face images, of a total of 40 people (10 samples per person). The images were captured at different times and have different variations including expressions (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees.

All the face images are manually aligned and cropped. The size of each cropped image is 32×32 pixels, with 256 gray levels per pixel. The features (pixel values) are then

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

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

Table 2. Recognition accuracy on Yale (mean±std-dev%)

Method	G2/P9	G3/P8	G4/P7	G5/P6
Eigenface	46.0±3.4	50.0±3.5	55.7±3.5	57.7±3.8
TensorPCA	49.4±3.5	54.0±3.0	57.8±3.3	59.8±3.9
Fisherface	45.7±4.2	62.3±4.5	73.0±5.4	76.9±3.2
2DLDA	43.4±6.2	56.3±4.7	63.5±5.6	66.1±4.8
S-LDA	**57.6±4.1**	**72.3±4.4**	**77.8±3.0**	**81.7±3.2**
Laplacianface	54.5±5.2	67.2±4.1	72.7±4.2	75.8±4.6
TSA	44.3±6.5	55.8±4.5	63.2±6.0	65.7±4.6
S-LPP	**57.9±4.5**	**72.0±4.0**	**76.0±3.4**	**81.4±2.9**
NPE	52.6±4.0	66.0±4.6	73.2±5.0	76.4±4.4
TNPE	43.4±6.2	56.8±3.9	61.8±3.5	63.0±3.4
S-NPE	**57.5±4.7**	**71.9±3.9**	**77.0±3.4**	**80.9±3.5**
MFA	45.7±4.2	62.3±4.5	73.0±5.4	76.9±3.2
TMFA	43.4±6.2	56.3±4.7	63.5±5.6	66.1±4.8
S-MFA	**57.2±4.3**	**71.2±4.0**	**76.9±3.1**	**81.1±3.1**

Table 3. Recognition accuracy on AT&T (mean±std-dev%)

Method	G2/P8	G3/P7	G4/P6	G5/P5
Eigenface	70.7±2.7	78.9±2.3	84.2±2.1	87.9±2.5
TensorPCA	71.3±2.6	79.9±2.2	84.8±1.9	88.1±2.5
Fisherface	75.5±3.3	86.1±1.9	91.6±1.9	94.3±1.4
2DLDA	80.4±3.0	89.8±2.1	93.5±1.7	95.8±1.2
S-LDA	**85.2±2.2**	**92.3±1.7**	**95.8±1.3**	**97.2±1.3**
Laplacianface	77.6±2.5	86.0±2.0	90.3±1.7	93.0±1.9
TSA	80.4±3.2	89.8±2.1	93.4±1.6	95.7±1.3
S-LPP	**85.2±2.2**	**92.3±1.7**	**95.8±1.3**	**97.2±1.3**
NPE	77.6±2.7	85.7±1.8	90.5±1.8	93.4±1.8
TNPE	80.4±3.0	87.6±2.2	91.5±1.7	93.7±2.3
S-NPE	**84.8±2.3**	**92.3±1.7**	**95.4±1.2**	**96.9±0.9**
MFA	75.4±3.1	86.1±1.9	91.6±1.9	94.3±1.4
TMFA	80.4±3.0	89.8±2.1	93.7±1.7	95.8±1.2
S-MFA	**84.9±2.3**	**92.4±1.3**	**95.8±1.5**	**97.4±1.2**

scaled to [0,1] (divided by 256). For the vector-based approaches, the image is represented as a 1024-dimensional vector, while for the tensor-based approaches the image is represented as a (32×32)-dimensional matrix, or the second order tensor.

The image set is then partitioned into the gallery and probe set with different numbers. For ease of representation, Gm/Pn means m images per person are randomly selected for training and the remaining n images are for testing.

Table 1 summarizes the 14 algorithms compared in our experiments. These algorithms belong to five families, *i.e.*, PCA family, LDA family, LPP family, NPE family and MFA (LDE) family. For each family, we take the ordinary vector-based approaches and their tensor extensions (or 2D extensions). Finally, we implement their spatially smooth versions by using 2-D Laplacian smoothing regularization technique.

4.2. Face recognition results

The recognition accuracy of different algorithms on Yale and AT&T databases are reported on the Table (2) and (3) respectively. For each Gp/Pq, we average the results over 20 random splits and report the mean as well as the standard deviation. The cross validation in the training set was used to select the parameter α in those SSSL approaches (S-LDA, S-LPP, S-NPE and S-MFA).

A crucial problems for most of the subspace learning based face recognition methods is dimensionality estimation. The performance usually varies with the number of dimensions. We show the best results obtained by those ordinary subspace learning algorithms and their tensor extensions. Since the cross validation is needed to estimate the parameter α for those SSSL approaches, we simply set the dimensionality as $c-1$ for those SSSL approaches where c is the number of individuals.

The main observations from the performance comparisons include:

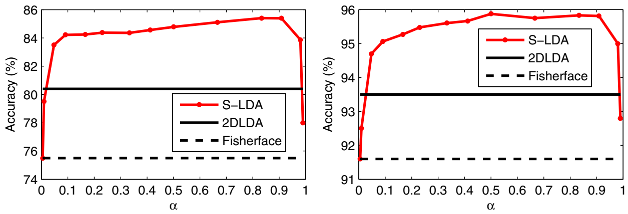
- SSSL approach significantly outperforms the corresponding ordinary subspace learning algorithm and the tensor extension with different numbers of training samples per individual in both the two databases. The reason lies SSSL explicitly takes into account the spatial relationship between the pixels in an image. The use of spatial information significantly reduces the number of degrees of freedom. Therefore, SSSL can have good performance even when there is only a small number of training samples available.

- The tensor-based algorithms show their advantages on AT&T database while failed gain improvement on Yale database. This suggests that the spatial relationship of face images considered in tensor-based approach (relation between the pixels in the same row or column) has its limitation. Compare to the tensor approaches, our SSSL approach is a more natural extension of incorporating spatial information in vector-based algorithm, which is supported by the experimental results.

4.3. Model selection for SSSL

The $\alpha \geq 0$ is an essential parameter in our SSSL approaches which controls the smoothness of the estimator. We use cross validation on the training set to select this parameter in the previous experiments. In this subsection, we take S-LDA as an example to study the impact of parameter α on the recognition performance.

Figure (2) shows the performance of S-LDA as a function of the parameter α on AT&T database. Each figure has three lines. The curve shows the accuracy of S-LDA with respect to α. The solid line shows the accuracy of 2DLDA



(a) 2 Train (b) 4 Train

Figure 2. Model selection for S-LDA on AT&T database.

and the dashed line shows the performance of Fisherface. It is easy to see that S-LDA can achieve significantly better performance than both 2DLDA and Fisherface over a large range of α. Thus, the model selection is not a very crucial problem in S-LDA algorithm.

5. Conclusions

In this paper, we propose a new family of linear dimensionality reduction methods called Spatially Smooth Subspace Learning (SSSL). SSSL explicitly considers the spatial relationship between the pixels in images. By introducing a Laplacian penalized functional, the projection vectors obtained by SSSL can be smoother than those obtained by the ordinary subspace learning algorithm. This prior information significantly reduces the number of degrees of freedom, and hence SSSL can perform better than the corresponding ordinary subspace learning version. We developed S-LDA, S-LPP, S-NPE and S-MFA based on the SSSL model and applied to face recognition on Yale and AT&T databases. Experimental results show that our method consistently outperforms the ordinary subspace learning algorithms and their tensor extensions.

Acknowledgments

We thank the reviewers for helpful comments. The work was supported in part by the U.S. National Science Foundation NSF IIS-05-13678/06-42771 and NSF BDI-05-15813.

References

[1] P. Belhumeur, J. Hepanha, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on PAMI*, 19(7):711–720, 1997. 1, 5

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS 14*. 2001. 1, 2

[3] M. Brand. Continuous nonlinear dimensionality reduction by kernel eigenmaps. In *International Joint Conference on Artificial Intelligence*, 2003. 2

[4] B. L. Buzbee, G. H. Golub, and C. W. Nielson. On direct methods for solving poisson's equations. *SIAM Journal on Numerical Analysis*, 7(4):627–656, Dec. 1970. 4

[5] D. Cai, X. He, and J. Han. Subspace learning based on tensor analysis. Technical report, Computer Science Department, UIUC, UIUCDCS-R-2005-2572, May 2005. 1, 3, 5

[6] D. Cai, X. He, J. Han, and H.-J. Zhang. Orthogonal laplacianfaces for face recognition. *IEEE Transactions on Image Processing*, 15(11):3608–3614, 2006. 1

[7] H.-T. Chen, H.-W. Chang, and T.-L. Liu. Local discriminant embedding and its variants. In *Proc. CVPR*, 2005. 1, 3

[8] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS, 1997. 2

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2nd edition, 2000. 1

[10] S. Guattery and G. L. Miller. Graph embeddings and laplacian eigenvalues. *SIAM Journal on Matrix Analysis and Applications*, 21(3):703–723, 2000. 2

[11] T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23:73–102, 1995. 5

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001. 5

[13] X. He, D. Cai, and P. Niyogi. Tensor subspace analysis. In *NIPS 18*, 2005. 1, 3, 5

[14] X. He, D. Cai, S. Yan, and H.-J. Zhang. Neighborhood preserving embedding. In *Proc. ICCV*, 2005. 1, 3, 5

[15] X. He and P. Niyogi. Locality preserving projections. In *NIPS 16*. 2003. 1

[16] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005. 1, 2, 3, 5

[17] R. Horn and C. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991. 5

[18] J. Jost. *Riemannian Geometry and Geometric Analysis*. Springer-Verlag, 2002. 4

[19] J. M. Lee. *Introduction to Smooth Manifolds*. Springer-Verlag New York, 2002. 3

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

[21] F. O'Sullivan. Discretized laplacian smoothing by fourier methods. *Journal of the American Statistical Association*, 86(415):634–642, Sep. 1991. 4

[22] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 1, 2

[23] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 1, 2

[24] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991. 1, 5

[25] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis for image ensembles. In *Proc. CVPR*, 2003. 1

[26] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extension: A general framework for dimensionality reduction. *IEEE Transactions on PAMI*, 29(1), 2007. 1, 2, 3, 5

[27] J. Ye, R. Janardan, and Q. Li. Two-dimensional linear discriminant analysis. In *NIPS 17*, 2004. 1, 3, 5