Report No. UIUCDCS-R-2005-2572 **UILU-ENG-2005-1767**

Subspace Learning Based on Tensor Analysis

by

Deng Cai, Xiaofei He, and Jiawei Han

May 2005

Subspace Learning Based on Tensor Analysis*

Deng Cai† Xiaofei He‡ Jiawei Han†

† Department of Computer Science, University of Illinois at Urbana-Champaign

‡ Department of Computer Science, University of Chicago

Abstract

Linear dimensionality reduction techniques have been widely used in pattern recognition and computer vision, such as face recognition, image retrieval, etc. The typical methods include Principal Component Analysis (PCA) which is unsupervised and Linear Discriminant Analysis (LDA) which is supervised. Both of them consider an $m_1 \times m_2$ image as a high dimensional vector in $\mathbb{R}^{m_1 \times m_2}$. Such a vector representation fails to take into account the spatial locality of pixels in the image. An image is intrinsically a matrix. In this paper, we consider an image as the second order tensor in $\mathbb{R}^{m_1} \otimes \mathbb{R}^{m_2}$ and propose two novel tensor subspace learning algorithms called TensorPCA and TensorLDA. Our algorithms explicitly take into account the relationship between column vectors of the image matrix and the relationship between the row vectors of the image matrix. We compare our proposed approaches with PCA and LDA for face recognition on three standard face databases. Experimental results show that tensor analysis achieves better recognition accuracy, while being much more efficient.

* The work was supported in part by the U.S. National Science Foundation NSF IIS-02-09199/IIS-03-08215. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

1 Introduction

Real data of natural and social sciences is often very high-dimensional. However, the underlying structure can in many cases be characterized by a small number of parameters. Reducing the dimensionality of such data is beneficial for visualizing the intrinsic structure and it is also an important preprocessing step in many statistical pattern recognition problems, such as face recognition [8, 1] and image retrieval [7, 2].

One of the fundamental problems in data analysis is how to represent the data. In the most of previous works on face representation, the face is represented as a vector in high-dimensional space. However, an image is intrinsically a matrix, or the second order tensor. In vector representation, the face image has to be converted to a vector. A typical way to do this so-called "matrix-to-vector alignment" is to concatenate all the rows in the matrix together to get a single vector. In this way, an image of size $m_1 \times m_2$ is identified with a point in $\mathbb{R}^{m_1 \times m_2}$. Thus, a linear transformation in $\mathbb{R}^{m_1 \times m_2}$ is characterized by $\mathbf{y} = A^T\mathbf{x}$, where A is an $(m_1 \cdot m_2) \times (m_1 \cdot m_2)$ matrix. To acquire such linear transformation, traditional subspace learning methods (PCA and LDA) need to eigen-decomposition of some matrices with size $(m_1 \cdot m_2) \times (m_1 \cdot m_2)$, which is computational expensive. Moreover, the learning parameters in PCA and LDA (the dimension of the projection vector) are very large. These methods might not acquire good performance when the number of training samples is small.

Recently, multilinear algebra, the algebra of higher-order tensors, was applied for analyzing the multifactor structure of image ensembles [10, 11, 12]. Vasilescu and Terzopoulos have proposed a novel face representation algorithm called Tensorface [10]. Tensorface represents the set of face images by a higher-order tensor and extends traditional PCA to higher-order tensor decomposition. In this way, the multiple factors related to expression, illumination and pose can be separated from different dimensions of the tensor. However, Tensorface still considers each face image as a vector instead of 2-d tensor. Thus, Tensorface is computationally expensive. Moreover, it does not encode discriminating information, thus it is not optimal for recognition.

In this paper, we systemically study the tensor analysis for subspace learning and propose two novel algorithms called TensorPCA and TensorLDA for learning a tensor subspace. TensorPCA and TensorLDA are fundamentally based on the traditional Principle Component Analysis and Linear Discriminant Analysis algorithms. An image is represented as a matrix $X \in \mathbb{R}^{m_1} \otimes \mathbb{R}^{m_2}$ and the linear transformation in the tensor space is naturally represented as $Y = U^T XV$, where U is a $m_1 \times m_1$ matrix and V is a $m_2 \times m_2$ matrix. The new algorithm is distinct from several perspectives:

1. Comparing to traditional subspace learning algorithm such as PCA and LDA, our algorithm is much more computationally efficient. This can be simply seen from the fact that the matrices U and V are much smaller than the matrix A.

2. Comparing to recently proposed Tensorface approach [10], our algorithm explicitly model each face image as a 2-D tensor which considers spatial locality of pixels in the face image. Also, the TesorLDA incorporated the label information which can get a better performance for face recognition.

3. Our algorithm is especially suitable for the small sample issue, since the number of parameters in our algorithm is much smaller that of traditional subspace learning algorithms.

The rest of this paper is organized as follows. We give a brief review of PCA and LDA in Section 2. The TensorPCA and TensorLDA algorithms are introduced in Section 3. Experimental results are shown in Section 4. Finally, we provide some Concluding remarks and suggestions for future work in Section 5.

2 A Brief review of PCA and LDA

PCA is a canonical linear dimensionality reduction algorithm. The basic idea of PCA is to project the data along the directions of maximal variances so that the reconstruction error can be minimized. Given a set of data points $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$, let $\mathbf{w}$ be the transformation vector and $y_i = \mathbf{w}^T \mathbf{x}_i$. The objective function of PCA is as follows:

$$\begin{aligned} \mathbf{w}_{opt} &= \arg\max_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \overline{y})^2 \\ &= \arg\max_{\mathbf{w}} \mathbf{w}^T C \mathbf{w} \end{aligned}$$

where $\overline{y} = \frac{1}{n}\sum y_i$ and C is the data covariance matrix. The basis functions of PCA are the eigenvectors of the data covariance matrix associated with the largest eigenvalues.

While PCA seeks directions that are efficient for representation, Linear Discriminant Analysis (LDA) seeks directions that are efficient for discrimination. Suppose we have a set of n samples $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$, belonging to k classes. The objective function of LDA is as follows:

$$\mathbf{w}_{opt} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$$

$$S_B = \sum_{i=1}^{k} n_i (\mathbf{m}^{(i)} - \mathbf{m})(\mathbf{m}^{(i)} - \mathbf{m})^T$$

$$S_W = \sum_{i=1}^{k} \left(\sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})(\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})^T \right)$$

where $\mathbf{m}$ is the total sample mean vector, n_i is the number of samples in the i-th class, $\mathbf{m}^{(i)}$ is the average vector of the i-th class, and $\mathbf{x}_j^{(i)}$ is the j-th sample in the i-th class. We call S_W the within-class scatter matrix and S_B the between-class scatter matrix.

3 Subspace Learning based on Tensor Analysis

Let $X \in \mathbb{R}^{m_1 \times m_2}$ denote an image of size $m_1 \times m_2$. Mathematically, X can be thought of as the 2^{nd} order tensor (or, 2-tensor) in the tensor space $\mathbb{R}^{m_1} \otimes \mathbb{R}^{m_2}$. Let $(\mathbf{u}_1, \cdots, \mathbf{u}_{m_1})$ be a set of orthonormal basis functions of $\mathbb{R}^{m_1}$. Let $(\mathbf{v}_1, \cdots, \mathbf{v}_{m_2})$ be a set of orthonormal basis functions of $\mathbb{R}^{m_2}$. Thus, an 2-tensor X can be uniquely written as:

$$X = \sum_{ij} (\mathbf{u}_i^T X \mathbf{v}_j) \mathbf{u}_i \mathbf{v}_j^T$$

This indicates that $\{\mathbf{u}_i \mathbf{v}_j^T\}$ forms a basis of the tensor space $\mathbb{R}^{m_1} \otimes \mathbb{R}^{m_2}$. Define two matrices $U = [\mathbf{u}_1, \cdots, \mathbf{u}_{l_1}] \in \mathbb{R}^{m_1 \times l_1}$ and $V = [\mathbf{v}_1, \cdots, \mathbf{v}_{l_2}] \in \mathbb{R}^{m_2 \times l_2}$. Let $\mathcal{U}$ be a subspace of $\mathbb{R}^{m_1}$ spanned by $\{\mathbf{u}_i\}_{i=1}^{l_1}$ and $\mathcal{V}$ be a subspace of $\mathbb{R}^{m_2}$ spanned by $\{\mathbf{v}_j\}_{j=1}^{l_2}$. Thus, the tensor product $\mathcal{U} \otimes \mathcal{V}$ is a subspace of $\mathbb{R}^{m_1} \otimes \mathbb{R}^{m_2}$. The projection of $X \in \mathbb{R}^{m_1 \times m_2}$ onto the space $\mathcal{U} \otimes \mathcal{V}$ is $Y = U^T X V \in \mathbb{R}^{l_1 \times l_2}$

Suppose we have n images, $X_1, \cdots, X_n \in \mathbb{R}^{m_1 \times m_2}$. These images belong to k categories $C_1, \cdots, C_k$. For the i-th category, there are n_i images. The mean of each category $M_i^{(X)}$ is computed by taking the average of X in category i, i.e.,

$$M_i^{(X)} = \frac{1}{n_i} \sum_{j \in C_i} X_j$$

and the global mean $M^{(X)}$ is defined as

$$M^{(X)} = \frac{1}{n} \sum_{j=1}^{n} X_j$$

Let $Y_i = U^T X_i V \in \mathbb{R}^{l_1 \times l_2}$. Likewise, we can define $M_i^{(Y)} = \frac{1}{n_i} \sum_{j \in C_i} Y_j$ and $M^{(Y)} = \frac{1}{n} \sum_{j=1}^{n} Y_j$. It is easy to check that $M_i^{(Y)} = U^T M_i^{(X)} V$ and $M^{(Y)} = U^T M^{(X)} V$.

The tensor subspace learning problem aims at finding the $(l_1 \times l_2)$-dimensional space $\mathcal{U} \otimes \mathcal{V}$ based on the specific objective functions. Particularly, we will introduce two novel algorithms called TensorPCA and TensorLDA in this section.

It would be important to note that, the number of parameters in the tensor subspace learning problem is much smaller than that in the original subspace learning problem like PCA and LDA. Therefore, tensor analysis should be particularly applicable for the small sample issue.

3.1 Tensor PCA

TensorPCA is fundamentally based on PCA. It tries to project the data to the tensor subspace of maximal variances so that the reconstruction error can be minimized. The objective function of TensorPCA can be described as follows:

$$\max_{U,V} \sum_i \|Y_i - M^{(Y)}\|^2 \tag{1}$$

Note that we use tensor norm of the difference of two tensors to measure the distance of two images. Since order two tensor is essentially matrix, we use Frobenius norm of a matrix as our 2-d tensor norm.

Since $\|A\|^2 = \mathrm{tr}(AA^T)$, we have

$$\begin{aligned}
\sum_{i=1}^{n} \|Y_i - M^{(Y)}\|^2 &= \sum_{i=1}^{n} \mathrm{tr}\left((Y_i - M^{(Y)})(Y_i - M^{(Y)})^T\right) \\
&= \sum_{i=1}^{n} \mathrm{tr}\left(U^T(X_i - M^{(X)})VV^T(X_i - M^{(X)})^T U\right) \\
&= \mathrm{tr}\left(U^T\left(\sum_{i=1}^{n}\left((X_i - M^{(X)})VV^T(X_i - M^{(X)})^T\right)\right)U\right) \\
&\doteq \mathrm{tr}\left(U^T M_V U\right)
\end{aligned}$$

where $M_V = \sum_{i=1}^{n}\left((X_i - M^{(X)})VV^T(X_i - M^{(X)})^T\right)$. Similarly, $\|A\|^2 = \mathrm{tr}(A^TA)$, so we also have

$$\begin{aligned}
\sum_{i} \|Y_i - M^{(Y)}\|^2 &= \mathrm{tr}\Big(V^T\Big(\sum_{i=1}^{n}\left((X_i - M^{(X)})^T UU^T(X_i - M^{(X)})\right)\Big)V\Big) \\
&\doteq \mathrm{tr}\left(V^T M_U V\right)
\end{aligned}$$

where $M_U = \sum_{i=1}^{n}\left((X_i - M^{(X)})^T UU^T(X_i - M^{(X)})\right)$. Thus, the optimal projection U should be the eigenvectors of M_V and the optimal projection V should be the eigenvectors of M_U.

One might notice that U and V can not be computed independently. In our algorithm, we try to find an optimal coordinate system of $\mathbb{R}^{m_1} \otimes \mathbb{R}^{m_2}$. That is, we assume that both U and V are orthonormal, i.e. $U^TU = UU^T = I$ and $V^TV = VV^T = I$. In such case,

$$M'_V = \sum_{i=1}^{n}\left((X_i - M^{(X)})(X_i - M^{(X)})^T\right)$$

and

$$M'_U = \sum_{i=1}^{n}\left((X_i - M^{(X)})^T(X_i - M^{(X)})\right)$$

It is clear that M'_V no longer depends on V, and M'_U no longer depends on U. Therefore, the matrix U can be simply computed as the eigenvectors of M'_V and the matrix V can be computed as the eigenvectors of M'_U. Note that, both M'_U and M'_V are symmetric, hence their eigenvectors are orthonormal. This is consistent with our assumptions. If we try to reduce the original tensor space to a $l_1 \times l_2$ tensor subspace, we choose the first l_1 column vectors in U and the first l_2 column vectors in V.

3.2 Tensor LDA

TensorPCA is performed under unsupervised mode. Its goal is to minimize the reconstruction error. In this subsection, we introduce the TensorLDA algorithm which is performed under supervised

mode. It tries to find the most discriminative tensor subspace. By projecting the data points into the tensor subspace, the data points of different classes are far from each other while data points of the same class are close to each other. Similar to traditional Linear Discriminant Analysis, we try to maximize the following objective function:

$$\max_{U,V} \frac{\sum_{i=1}^{k} \sum_{j \in C_i} \|Y_j - M_i^{(Y)}\|^2}{\sum_{i=1}^{k} n_i \|M_i^{(Y)} - M^{(Y)}\|^2} \tag{2}$$

Since $\|A\|^2 = \mathrm{tr}(AA^T)$, we have

$$\begin{aligned}
\sum_{i=1}^{k} \sum_{j \in C_i} \|Y_j - M_i^{(Y)}\|^2 &= \sum_{i=1}^{k} \sum_{j \in C_i} \mathrm{tr}\left((Y_j - M_i^{(Y)})(Y_j - M_i^{(Y)})^T\right) \\
&= \mathrm{tr}\left(\sum_{i=1}^{k} \sum_{j \in C_i} U^T (X_j - M_i^{(X)}) V V^T (X_j - M_i^{(X)})^T U\right) \\
&\doteq \mathrm{tr}\left(U^T S_w^V U\right)
\end{aligned}$$

where $S_w^V = \sum_{i=1}^{k} \sum_{j \in C_i} (X_j - M_i^{(X)}) V V^T (X_j - M_i^{(X)})^T$. The denominator can be computed as follows:

$$\begin{aligned}
\sum_{i=1}^{k} n_i \|M_i^{(Y)} - M^{(Y)}\|^2 &= \sum_{i=1}^{k} n_i \left(\mathrm{tr}\left((M_i^{(Y)} - M^{(Y)})(M_i^{(Y)} - M^{(Y)})^T\right)\right) \\
&= \mathrm{tr}\left(\sum_{i=1}^{k} n_i U^T (M_i^{(X)} - M^{(X)}) V V^T (M_i^{(X)} - M^{(X)})^T U\right) \\
&\doteq \mathrm{tr}\left(U^T S_b^V U\right)
\end{aligned}$$

where $S_b^V = \sum_{i=1}^{k} n_i (M_i^{(X)} - M^{(X)}) V V^T (M_i^{(X)} - M^{(X)})^T$.

Similarly, $\|A\|^2 = trace(A^T A)$, so we also have

$$\sum_{i=1}^{k} \sum_{j \in C_i} \|Y_j - M_i^{(Y)}\|^2 = \mathrm{tr}\left(V^T S_w^U V\right)$$

$$S_w^U = \sum_{i=1}^{k} \sum_{j \in C_i} (X_j - M_i^{(X)})^T U U^T (X_j - M_i^{(X)})$$

and

$$\sum_{i=1}^{k} n_i \|M_i^{(Y)} - M^{(Y)}\|^2 = \mathrm{tr}\left(V^T S_b^U V\right)$$

$$S_b^U = \sum_{i=1}^{k} n_i (M_i^{(X)} - M^{(X)})^T U U^T (M_i^{(X)} - M^{(X)})$$

The optimal projections that preserve the given class structure should maximize $U^T S_w^V U / U^T S_b^V U$ and $V^T S_w^U V / V^T S_b^U V$ simultaneously. Thus U and V can be obtained by solving the following two generalized eigenvector problems:

$$S_b^V \mathbf{a} = \lambda S_w^V \mathbf{a} \tag{3}$$

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

Unfortunately, the matrices $S_b^V, S_b^U, S_w^V, S_w^U$ are not fixed while depend on U or V. Therefore, U and V can not be computed independently. To relax such a problem, we add a constraint that U and V be orthonormal. That is, $U^T U = U U^T = I$ and $V^T V = V V^T = I$. Thus,

$$\begin{aligned}
S_w^U &= \sum_{i=1}^{k} \sum_{j \in C_i} (X_j - M_i^{(X)})^T (X_j - M_i^{(X)}) \\
S_b^U &= \sum_{i=1}^{k} n_i (M_i^{(X)} - M^{(X)})^T (M_i^{(X)} - M^{(X)}) \\
S_w^V &= \sum_{i=1}^{k} \sum_{j \in C_i} (X_j - M_i^{(X)}) (X_j - M_i^{(X)})^T \\
S_b^V &= \sum_{i=1}^{k} n_i (M_i^{(X)} - M^{(X)}) (M_i^{(X)} - M^{(X)})^T
\end{aligned} \tag{5}$$

It is clear that, with the constraints, the above four matrices can be computed independently only depending on the training data points. In the following, we will describe how to compute U and V. It would be important to note that, the computations of U and V are essentially the same. Without loss of generality, we use S_w to represent S_w^U or S_w^V, and use S_b to represent S_b^U or S_b^V.

Now our problem becomes:

$$\mathbf{a}_1 = \arg\max_{\mathbf{a}} \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_w \mathbf{a}}$$

$$\mathbf{a}_k = \arg\max_{\mathbf{a}} \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_w \mathbf{a}}$$

with the constraint

$$\mathbf{a}_k^T \mathbf{a}_1 = \mathbf{a}_k^T \mathbf{a}_2 = \cdots = \mathbf{a}_k^T \mathbf{a}_{k-1} = 0$$

Since S_w is in general nonsigular, for any $\mathbf{a}$, we can always normalize it such that $\mathbf{a}^T S_w \mathbf{a} = 1$, and the ratio of $\mathbf{a}^T S_b \mathbf{a}$ and $\mathbf{a}^T S_w \mathbf{a}$ keeps unchanged. Thus, the above minimization problem is equivalent to minimizing the value of $\mathbf{a}^T S_b \mathbf{a}$ with an additional constraint as follows,

$$\mathbf{a}^T S_w \mathbf{a} = 1$$

Note that, the above normalization is only for simplifying the computation. Once we get the optimal solutions, we can re-normalize them to get a othonormal basis vectors.

It is easy to check that $\mathbf{a}_1$ is the eigenvector of the generalized eigen-problem:

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

associated with the largest eigenvalue. Since S_w is general nonsingular, $\mathbf{a}_1$ is the eigenvector of the matrix $(S_w)^{-1}S_b$ associated with the largest eigenvalue.

In order to get the k-th basis vector, we maximize the following objective function:

$$f(\mathbf{a}_k) = \frac{\mathbf{a}_k^T S_b \mathbf{a}_k}{\mathbf{a}_k^T S_w \mathbf{a}_k} \tag{6}$$

with the constraints:

$$\mathbf{a}_k^T\mathbf{a}_1 = \mathbf{a}_k^T\mathbf{a}_2 = \cdots = \mathbf{a}_k^T\mathbf{a}_{k-1} = 0, \quad \mathbf{a}_k^T S_w \mathbf{a}_k = 1$$

We can use the Lagrange multipliers to transform the above objective function to include all the constraints

$$C^{(k)} = \mathbf{a}_k^T S_b \mathbf{a}_k - \lambda(\mathbf{a}_k^T S_w \mathbf{a}_k - 1) - \mu_1 \mathbf{a}_k^T\mathbf{a}_1 - \cdots - \mu_{k-1}\mathbf{a}_k^T\mathbf{a}_{k-1}$$

The optimization is performed by setting the partial derivative of $C^{(k)}$ with respect to $\mathbf{a}_k$ to zero:

$$\begin{aligned} &\frac{\partial C^{(k)}}{\partial \mathbf{a}_k} = 0 \\ \Rightarrow\ & 2S_b\mathbf{a}_k - 2\lambda S_w\mathbf{a}_k - \mu_1\mathbf{a}_1 \cdots - \mu_{k-1}\mathbf{a}_{k-1} = 0 \end{aligned} \tag{7}$$

Multiplying the left side of (7) by $\mathbf{a}_k^T$, we obtain

$$\begin{aligned} & 2\mathbf{a}_k^T S_b\mathbf{a}_k - 2\lambda\mathbf{a}_k^T S_w\mathbf{a}_k = 0 \\ \Rightarrow\ & \lambda = \frac{\mathbf{a}_k^T S_b\mathbf{a}_k}{\mathbf{a}_k^T S_w\mathbf{a}_k} \end{aligned} \tag{8}$$

Comparing to (6), λ exactly represents the expression to be maximized.

Multiplying the left side of (7) successively by $\mathbf{a}_1^T S_w^{-1}, \cdots, \mathbf{a}_{k-1}^T S_w^{-1}$, we now obtain a set of $k-1$ equations:

$$\begin{aligned} \mu_1\mathbf{a}_1^T S_w^{-1}\mathbf{a}_1 + \cdots + \mu_{k-1}\mathbf{a}_1^T S_w^{-1}\mathbf{a}_{k-1} &= 2\mathbf{a}_1^T S_w^{-1} S_b\mathbf{a}_k \\ \mu_1\mathbf{a}_2^T S_w^{-1}\mathbf{a}_1 + \cdots + \mu_{k-1}\mathbf{a}_2^T S_w^{-1}\mathbf{a}_{k-1} &= 2\mathbf{a}_2^T S_w^{-1} S_b\mathbf{a}_k \\ \cdots\cdots & \\ \mu_1\mathbf{a}_{k-1}^T S_w^{-1}\mathbf{a}_1 + \cdots + \mu_{k-1}\mathbf{a}_{k-1}^T S_w^{-1}\mathbf{a}_{k-1} &= 2\mathbf{a}_{k-1}^T S_w^{-1} S_b\mathbf{a}_k \end{aligned}$$

We define:

$$\mu^{(k-1)} = [\mu_1, \cdots, \mu_{k-1}]^T, \quad A^{(k-1)} = [\mathbf{a}_1, \cdots, \mathbf{a}_{k-1}]$$

$$B^{(k-1)} = \left[B_{ij}^{(k-1)}\right] = \left[A^{(k-1)}\right]^T S_w^{-1} A^{(k-1)}, \quad B_{ij}^{(k-1)} = \mathbf{a}_i^T S_w^{-1}\mathbf{a}_j$$

Using this simplified notation, the previous set of $k-1$ equations can be represented in a single matrix relationship

$$B^{(k-1)}\mu^{(k-1)} = 2\left[A^{(k-1)}\right]^T S_w^{-1} S_b \mathbf{a}_k$$

thus

$$\mu^{(k-1)} = 2\left[B^{(k-1)}\right]^{-1}\left[A^{(k-1)}\right]^T S_w^{-1} S_b \mathbf{a}_k \tag{9}$$

Let us now multiply the left side of (7) by S_w^{-1}

$$2S_w^{-1}S_b\mathbf{a}_k - 2\lambda\mathbf{a}_k - \mu_1 S_w^{-1}\mathbf{a}_1 - \cdots - \mu_{k-1}S_w^{-1}\mathbf{a}_{k-1} = 0$$

This can be expressed using matrix notation as

$$2S_w^{-1}S_b\mathbf{a}_k - 2\lambda\mathbf{a}_k - S_w^{-1}A^{(k-1)}\mu^{(k-1)} = 0$$

With equation (9), we obtain

$$\left\{I - S_w^{-1}A^{(k-1)}\left[B^{(k-1)}\right]^{-1}\left[A^{(k-1)}\right]^T\right\} S_w^{-1}S_b\mathbf{a}_k = \lambda\mathbf{a}_k$$

As shown in (8), λ is just the criterion to be maximized, thus $\mathbf{a}_k$ is the eigenvector of

$$M^{(k)} = \left\{I - S_w^{-1}A^{(k-1)}\left[B^{(k-1)}\right]^{-1}\left[A^{(k-1)}\right]^T\right\} S_w^{-1}S_b$$

associated with the largest eigenvalue of $M^{(k)}$.

In summery, our TensorLDA algorithm is performed as follows:

1. Compute the S_w^U, S_b^U, S_w^V, S_b^V as described in Eqn. 5
2. Calculate the orthonormal matrix $U = [\mathbf{u}_1, \cdots, \mathbf{u}_{m_1}]$ as
   (a) $\mathbf{u}_1$ is the eigenvector of $(S_w^V)^{-1}S_b^V$ corresponding to the largest eigenvalue.
   (b) $\mathbf{u}_k$ is the eigenvector of $M_V^{(k)}$ corresponding to the largest eigenvalue, where

$$\begin{aligned} M_V^{(k)} &= \left\{I - \left(S_w^V\right)^{-1}U^{(k-1)}\left[B_V^{(k-1)}\right]^{-1}\left[U^{(k-1)}\right]^T\right\}\left(S_w^V\right)^{-1}S_b^V \\ U^{(k-1)} &= [\mathbf{u}_1, \cdots, \mathbf{u}_{k-1}] \\ B_V^{(k-1)} &= \left[U^{(k-1)}\right]^T\left(S_w^V\right)^{-1}U^{(k-1)} \end{aligned}$$

3. Calculate the orthonormal matrix $V = [\mathbf{v}_1, \cdots, \mathbf{v}_{m_2}]$ as
   (a) $\mathbf{v}_1$ is the eigenvector of $(S_w^U)^{-1}S_b^U$ corresponding to the largest eigenvalue.

(b) $\mathbf{v}_k$ is the eigenvector of $M_U^{(k)}$ corresponding to the largest eigenvalue, where

$$\begin{aligned} M_U^{(k)} &= \left\{ I - \left(S_w^U\right)^{-1} V^{(k-1)} \left[B_U^{(k-1)}\right]^{-1} \left[V^{(k-1)}\right]^T \right\} \left(S_w^U\right)^{-1} S_b^U \\ V^{(k-1)} &= [\mathbf{v}_1, \cdots, \mathbf{v}_{k-1}] \\ B_U^{(k-1)} &= \left[V^{(k-1)}\right]^T \left(S_w^U\right)^{-1} V^{(k-1)} \end{aligned}$$

If we try to reduce the original tensor space to a $l_1 \times l_2$ tensor subspace, we choose the first l_1 column vectors in U and the first l_2 column vectors in V.

4 Experimental Results

In this section, we investigate the performance of our proposed TensorPCA and TensorLDA methods for face recognition. The system performance is compared with the Eigenfaces method (PCA) [9] and the Fisherfaces method (PCA+LDA) [1], two of the most popular linear methods in face recognition.

In this study, three face databases were tested. The first one is the Yale database[1] , the second one is the ORL (Olivetti Research Laboratory) datebase[2], and the third one is the PIE (pose, illumination, and expression) database from CMU [6]. 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 matching. The size of each cropped image in Yale and PIE databases is 32×32 pixels, and 64×64 pixels for the ORL database, with 256 gray levels per pixel. For Fisherfaces and Eigenfaces, each image is represented by a $1,024$-dimensional (or $4,096$-dimensional for ORL database) vector in image space. For TensorPCA and TensorLDA, the images are represented as matrices. Different pattern classifiers have been applied for face recognition, including nearest-neighbor [1], Bayesian [4], Support Vector Machine [5], etc. In this paper, we apply the nearest-neighbor classifier for its simplicity. The Euclidean metric is used as our distance measure.

In short, the recognition process has three steps. First, we calculate the face subspace from the training set of face images; then the new face image to be identified is projected into d-dimensional subspace (PCA and LDA) or $(d \times d)$-dimensional tensor subspace (TensorPCA and TensroLDA); finally, the new face image is identified by a nearest neighbor classifier.

4.1 Yale Database

The Yale face database was constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. The images demonstrate variations in lighting

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

Figure 1: Sample face images from the Yale database. For each subject, there are 11 face images under different lighting conditions with facial expression.

Table 1: Performance comparisons on the Yale database

Method	Dims (d)	Error Rate
Baseline	1024	56.5%
Eigenfaces (PCA)	29	56.5%
Fisherfaces (PCA+LDA)	9	54.3%
TensorPCA	6	53.8%
TensorLDA	22	47.9%

condition (left-light, center-light, right-light), facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. Figure 1 show the 11 images of one individual in Yale data base. A random subset with 2 images per individual (hence, 30 images in total) was taken with labels to form the training set. The rest of the database was considered to be the testing set. 20 times of random selection for training examples were performed and the average recognition result was recorded. The training samples were used to learn the subspace. The testing samples were then projected into the low-dimensional representation subspace. Recognition was performed using a nearest-neighbor classifier. Note that, for LDA, there are at most $c-1$ nonzero generalized eigenvalues and, so, an upper bound on the dimension of the reduced space is $c-1$, where c is the number of individuals [1]. In general, the performance of the all these methods varies with the number of dimensions. We show the best results obtained by Fisherfaces, Eigenfaces, TensorPCA, TensorLDA and Baseline in Table 1. It is found that the TensorLDA method significantly outperforms both Eigenfaces and Fisherfaces methods. The error rates of Eigenfaces, Fisherfaces, TensorPCA and TensorLDA are 56.5%, 54.3%, 53.8% and 47.9%, respectively. Figure 2 shows the plots of error rate versus dimensionality reduction. It is interesting to note that when kept $c-1$ dimensions, Fisherfaces is even worse than baseline. This result is consistent with the observation in [3] that Eigenfaces can outperform Fisherfaces in small training sample.

4.2 ORL Database

ORL (Olivetti Research Laboratory) face database is a well-known dataset for face recognition. It contains 400 images of 40 individuals. Some images were captured at different times and have different variations including expression (open or closed eyes, smiling or non-smiling) and facial

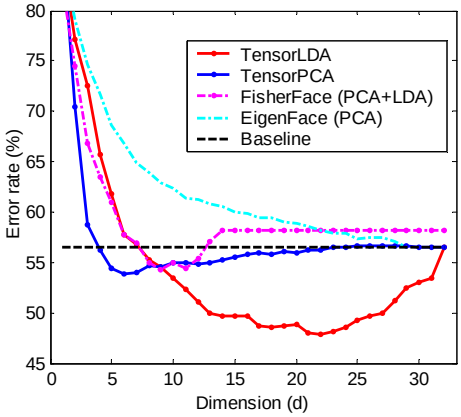

Figure 2: Error rate vs. dimensionality reduction on Yale database

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

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 images are grayscale and of size 64×64. 10 sample images of one individual in the ORL database are displayed in Figure 3. A random subset with 2 images per individual (hence, 80 images in total) was taken with labels to form the training set. The rest of the database was considered to be the testing set. 20 times of random selection for training example were performed and the average recognition result was recorded. The experimental protocol is the same as before. The recognition results are shown in Table 2. It is found that the TensorLDA method significantly outperforms both Eigenfaces and Fisherfaces methods. The error rates of Eigenfaces, Fisherfaces, TensorPCA and TensorLDA are 33.5%, 29.84%, 30.78% and 23.41%, respectively. Figure 2 shows the plots of error rate versus dimensionality reduction.

4.3 PIE Database

The CMU PIE face database 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. We choose the five near frontal pose (C05, C07, C09, C27, C29) and use all the images under different illumination, lighting and expression which leave us 170 near frontal face images for each individual. Figure 5 shows several sample images of one individual with different poses, expressions and illuminations. We randomly choose 20 images for each individual as training data. We repeat this process 20 times and compute the average performance. Table 3

Table 2: Performance comparisons on the ORL database

Method	Dims (d)	Error Rate
Baseline	4096	33.5%
Eigenfaces (PCA)	79	33.5%
Fisherfaces (PCA+LDA)	21	29.84%
TensorPCA	10	30.78%
TensorLDA	40	23.41%

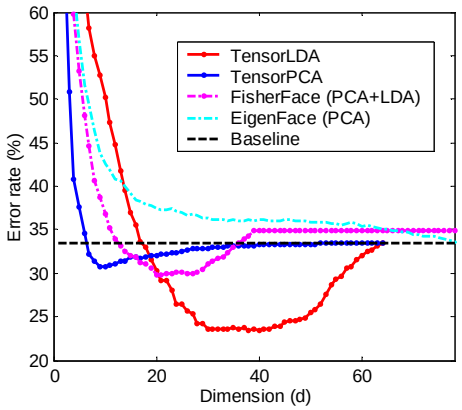

Figure 4: Error rate vs. dimensionality reduction on ORL database

shows the recognition results. As can be seen, Fisherfaces performs a bit worse than our algorithms on this database, while Eigenfaces performs poorly. The error rate for TensorLDA, TensorPCA, Fisherfaces, and Eigenfaces are 13.32%, 38.15%, 15.44%, and 38.14%, respectively. Figure 6 shows a plot of error rate versus dimensionality reduction. As can be seen, the error rate of our TensorLDA method decreases fast as the dimensionality of the face tensor subspace increases, and achieves the best result with $d = 13$ dimension. There is no significant improvement if more dimensions are used. For Fisherfaces, the dimension of the face subspace is bounded by $c - 1$, and it achieves the best result with $c - 1 = 67$ dimensions.

5 Conclusions

Tensor based subspace learning is introduced in this paper. We propose two new linear dimensionality reduction algorithm called TensorPCA and TensorLDA. Different from traditional algorithms like PCA and LDA, our algorithms are performed in the tensor space rather than the vector space. As a result they are especially suitable for face analysis which is intrinsically the second order tensor.

Figure 5: Sample face images from the CMU PIE database. For each subject, there are 170 near frontal face images under varying pose, illumination, and expression.

Table 3: Performance comparisons on the PIE database

Method	Dims (d)	Error Rate
Baseline	1024	38.15%
Eigenfaces (PCA)	889	38.14%
Fisherfaces (PCA+LDA)	67	15.44%
TensorPCA	32	38.15%
TensorLDA	13	13.32%

Several questions remain to be investigated in our future work,

1. In this paper, we applied our algorithms to face recognition in which the face images are naturally represented as the second order tensors. It is unclear if there is any other application domains in which tensor representation is essential.

2. Our algorithms are linear. Therefore, it can not capture the nonlinear structure of the data manifold. It remains unclear how to generalize our algorithms to nonlinear case.

References

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

[2] J. Dy, C. E. Brodley, A. Kak, L. S. Broderick, and A. M. Aisen. Hierarchical feature selection applied to content-based image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:373–378, 2003.

[3] A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE Trans. on PAMI*, 23(2):228–233, 2001.

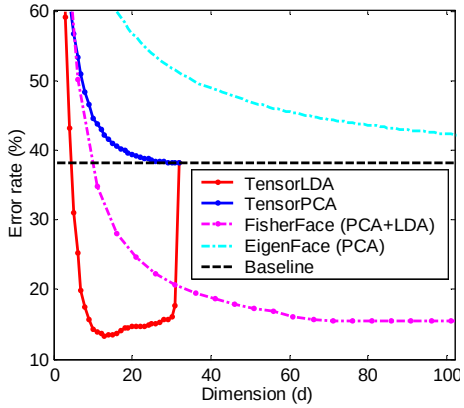

Figure 6: Error rate vs. dimensionality reduction on PIE database

[4] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. on PAMI*, 19(7):696–710, 1997.

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

[6] T. Sim, S. Baker, and M. Bsat. The CMU pose, illuminlation, and expression database. *IEEE Trans. on PAMI*, 25(12):1615–1618, 2003.

[7] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):831–836, 1996.

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

[9] M. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, Maui, Hawaii, 1991.

[10] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis for image ensembles. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[11] J. Yang, D. Zhang, A. Frangi, and J. Yang. Two-dimensional PCA: A new approach to appearance-based face representation and recogntion. *IEEE Trans. PAMI*, 26(1):131–137, Jan. 2004.

[12] J. Ye, R. Janardan, and Q. Li. Two-dimensional linear discriminant analysis. In *Advances in Neural Information Processing Systems 17*, 2004.