Distance Adaptive Tensor Discriminative Geometry Preserving Projection for Face Recognition

Regular Paper

Ziqiang Wang*, Xia Sun and Lijun Sun

School of Information Science and Engineering, Henan University of Technology, Zhengzhou, China

* Corresponding author E-mail: wzqagent@126.com

Received 01 Jun 2012; Accepted 16 Jul 2012

DOI: 10.5772/51612

© 2012 Wang et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract There is a growing interest in dimensionality reduction techniques for face recognition, however, the traditional dimensionality reduction algorithms often transform the input face image data into vectors before embedding. Such vectorization often ignores the underlying data structure and leads to higher computational complexity. To effectively cope with these problems, a novel dimensionality reduction algorithm termed distance adaptive tensor discriminative geometry preserving projection (DATDGPP) is proposed in this paper. The key idea of DATDGPP is as follows: first, the face image data are directly encoded in high-order tensor structure so that the relationships among the face image data can be preserved; second, the data-adaptive tensor distance is adopted to model the correlation among different coordinates of tensor data; third, the transformation matrix which can preserve discrimination and local geometry information is obtained by an iteration algorithm. Experimental results on three face databases show that the proposed algorithm outperforms other representative dimensionality reduction algorithms.

Keywords face recognition; manifold learning; tensor structure; discriminative geometry preserving projection; dimensionality reduction.

1. Introduction

Over the last decade face recognition has become one of the most active research areas in multimedia information processing due to the rapidly increasing requirements in many practical applications, such as identify authentication, information security, human-computer interaction/communication and so on. A major challenge of face recognition, however, is that the captured face image data often lies in a high-dimensional feature space. For example, a face image with the resolution of 128×128 pixels is represented as a point in a 16384-dimensional face space. Learning in such high-dimensionality in many cases is almost infeasible. Thus, learnability necessitates dimensionality reduction, which aims to find the lower-dimensional feature representation of high-dimensional data with enhanced discriminatory power. Once the high-dimensional face image data is projected into lower-
dimensional feature space, traditional classification schemes can then be applied.

The most popular conventional algorithms for dimensionality reduction are principal component analysis (PCA) and linear discriminant analysis (LDA) [1]. PCA maintains the global Euclidean structure of the data in the original high-dimensional space and projects the data points into a lower-dimensional subspace, in which the sample variance is maximized. In contrast to the unsupervised method of PCA, LDA is a supervised learning approach. LDA aims to seek the project axes on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other. The optimal projection of LDA is computed by minimizing the within-class scatter and maximizing the between-class scatter simultaneously. It is generally believed that LDA-based algorithms outperform PCA-based ones, since the former optimizes the low-dimensional representation of the objects with the most discriminant information, while the latter simply achieves object reconstruction. In addition, non-negative matrix factorization (NMF) [2] has been proposed for face recognition. NMF learns the parts of objects by using non-negativity constraints, which leads to a parts-based representation of objects. However, the iterative update method for solving the NMF problem is computationally expensive. Therefore, although the above mentioned algorithms have widely been applied to face recognition, they are designed for discovering only the global Euclidean structure, whereas the local manifold structure is ignored.

Recently, a number of manifold learning algorithms have been proposed to discover the geometric property of high-dimensional data spaces and they have been successfully applied to face recognition. Manifold learning aims at discovering the geometric properties of the data space, such as its Euclidean embedding, intrinsic dimensionality, connected components, homology, etc. The desired manifold is an intrinsically lower-dimensional space hidden in the input high-dimensional space. The most well-known manifold learning algorithm include isometric feature mapping (ISOMAP) [3], locally linear embedding (LLE) [4] and Laplacian eigenmap (LE) [5]. ISOMAP, a variant of multidimensional scaling (MDS), aims to preserve the global geodesic distances between any pair of data points. LLE aims to embed data points in a low-dimensional space by finding the optimal linear reconstruction in a small neighbourhood. LE aims to preserve proximity relationships by manipulations on an undirected weighted graph, which indicates neighbour relations of pairwise data points. These nonlinear methods have achieved impressive results on some pattern classification tasks, however, the mappings derived from them are defined only on the training data points and thus, how to evaluate the maps on novel test data points remains unclear [6]. One common response to cope with this problem is to apply a linearization procedure to construct explicit maps over new samples. The representative example of this approach is locality preserving projection (LPP) [7], a linearization of LE. LPP is obtained by finding the optimal linear approximations to the eigenvectors of the Laplace Beltrami operator on the manifold. Despite this, dimensionality reduction methods are based on different embedding criteria. Yan et al. demonstrated that most of them can be mathematically unified within a general framework, called graph embedding [8]. Based on the graph embedding framework, marginal Fisher analysis (MFA) was developed to jointly consider the local manifold structure and the class label information for dimensionality reduction. However, MFA extracts discriminative information from only marginal samples, although non-marginal samples also contain the discriminative information. Recently, Song et al. has proposed discriminative geometry preserving projections (DGPP) [9] which has yielded impressive results on scene classification. DGPP is fundamentally based on manifold learning, but simultaneously considering both intraclass geometry and interclass discrimination for dimensionality reduction.

However, most previous dimensionality reduction algorithms often operate on input face image data after they have been transformed into a 1-D vector. In fact, face image data are intrinsically in the form of second- or higher-order tensors [10,11]. For example, grey-level face images represent second-order tensor data (matrices) and can be expanded to third-order tensors by representing sets of images after Gabor filtering. Therefore, such vectorization ignores the underlying data structure and often leads to the curse of the dimensionality dilemma and the small sample size problem. To address these problems, many tensor-based dimensionality reduction algorithms have been proposed. Representative algorithms include tensor principal component analysis (TPCA) [12], tensor linear discriminant analysis (TLDA) [10], tensor locality preserving projection (TLPP) [13], tensor marginal Fisher analysis (TMFA) [8] and so on. However, most of the existing tensor-based algorithms simply use the traditional Euclidean distance to measure relationships among different data points. Despite its prevalent usages, Euclidean distance often ignores the relationships among different coordinates for high-order data, such as the spatial relations of pixels in images, which has been shown in many previous studies to be very useful for improving the learning performance [14]. Therefore, it is very natural that both embedding strategies and their related distance metric should be considered in designing tensor-based dimensionality reduction algorithms. However, most of the existing

tensor learning algorithms fail to take into account the correlation among different coordinates of data with arbitrary number of orders.

In this paper, we propose a novel distance adaptive tensor manifold learning algorithm for face recognition. By using a data-adaptive tensor distance metric proposed by Liu et al. [14], we can effectively exploit the spatial correlation relationships of face image data to enhance the learning performance. We discuss how to tensorize the discriminative geometry preserving projection which gives rise to distance adaptive tensor dimensionality reduction algorithm for face recognition.

The rest of the paper is organized as follows: in Section 2, we provide a brief review of the original vector-based discriminative geometry preserving projection (DGPP) algorithm. Our distance adaptive tensor manifold learning algorithm for face recognition is introduced in Section 3. The experimental results on face recognition are presented in Section 4. Finally, we provide the concluding remarks and suggestions for future work in Section 5.

2. Brief review of DGPP

The original vector-based discriminative geometry preserving projection (DGPP) [9] is a recently proposed manifold learning algorithm for dimensionality reduction, it can precisely model both the intraclass geometry and interclass discrimination by using an average weighted adjacency graph and local linear reconstruction error. In addition, the original vector-based DGPP avoids the out of sample problem which exists in traditional manifold learning algorithms by applying a linearization procedure to construct explicit maps over new samples.

Given a set of face images \( \{ x_1, x_2, \ldots, x_n \} \subset \mathbb{R}^p \), let \( X = [x_1, x_2, \ldots, x_n] \). DGPP aims to find a linear transformation \( U \in \mathbb{R}^{p \times q} \) that maps each face image \( x_i \) to \( U^T x_i \). Given \( x_i \), we aim to find \( x_i \) such that \( y_i \) represents \( x_i \) well in terms of the discrimination preservation and local geometry preservation criterion. The optimal linear transformation of DGPP can be obtained by solving the following maximization problem:

\[
U_{\text{opt}} = \arg\max_U \sum_{i,j} h_{ij} (U^T x_i - U^T x_j)^2 - \sum_{i,j} \sum_{c_i \neq c_j} w_{ij} (U^T x_i - U^T x_j)^2 
\]

with the constraint

\[
U^T U = I 
\]

where \( I \) is an identity matrix, \( D \in \mathbb{R}^{n \times n} \) is a diagonal matrix and its \( ith \) entry is \( \sum_{j=1}^n h_{ij} \), and the weighing factor matrix \( H = [h_{ij}] \in \mathbb{R}^{n \times n} \) denotes both the distance weighing information and the class label information in terms of

\[
h_{ij} = \begin{cases} 
\exp \left( \frac{-\|x_i - x_j\|^2}{\delta^2} \right) (1/n - 1/n_i), & \text{if } c_i = c_j = l \\[1/n_i], & \text{if } c_i = c_j 
\end{cases} 
\]

where \( \delta \) is a positive constant and its setting can be referred to in [5]. \( c_i \in \{1, 2, \ldots, C\} \) denotes the class label of face image \( x_i \) and the \( l \)th class contains \( n_l \) samples satisfying \( \sum_{l=1}^C n_l = n \).

Meanwhile, \( W = [w_{ij}] \in \mathbb{R}^{n \times n} \) is the local linear reconstruction coefficient matrix, wherein \( w_{ij} \) denotes the reconstruction coefficient of \( x_i \) which can be linearly reconstructed from the samples \( x_j \) with the same class label \( c_i = c_j \). By imposing \( \sum_{j \in c_i} w_{ij} = 1 \) and \( w_{ij} = 0 \) for \( c_i \neq c_j \), \( w_{ij} \) can be obtained by solving the following reconstruction error minimization problem:

\[
w_{ij} = \arg\min_{w_{ij}} \sum_{i=1}^n \| x_i - \sum_{j \in c_i} w_{ij} x_j \|^2 
\]

Thus, we can easily obtain \( w_i = \sum_{j \in c_i} C_{i,j}^{-1} / \sum_{j \in c_i} C_{i,j}^{-1} \), wherein \( C_{p,q} = (x_i - x_p)^T (x_j - x_q) \) is the local Gram matrix and \( C_{p,q} = \delta_{c_i, c_j} \).

As can be seen from the above statement, DGPP aims to look for a linear transformation matrix \( U \) such that the distances between interclass samples are as large as possible, while distances between intraclass samples are as small as possible, and the local geometry of intraclass
samples is preserved as much as possible by keeping linear reconstruction error minimization. Finally, with simple matrix operations, the transformation matrix $U$ that maximizes (1) is given by the maximum eigenvalue solution to the following stand eigenproblem:

$$X (D-H^T)(D-H^T)^T - (I-W^T)(I-W^T)^T ) X^T U = \lambda U \quad (5)$$

Thus, the matrix $U$ can also be regarded as the eigenvectors of the matrix $X (D-H^T)(D-H^T)^T - (I-W^T)(I-W^T)^T ) X^T$ associated with the largest eigenvalues.

Although the original vector-based DGPP has shown promising results on scene classification, it commonly unfolds each image into a single column vector before dimensionality reduction. In fact, image objects are intrinsically in the form of second or higher-order tensors. Thus, the vectorization operation of DGPP largely increases the computational costs of image data analysis and seriously destroys the intrinsic structure relationships among different coordinates of high-order image data which has been shown in many previous studies to be very useful for improving the learning performance [14,15].

3. Distance adaptive tensor discriminative geometry preserving projection for face recognition

In order to preserve the intrinsic relationships among different coordinates for high-order face image data, a natural way is to perform dimensionality reduction in their original high-order tensor space. In this section, we will describe our distance adaptive tensor discriminative geometry preserving projection approach which is fundamentally based on discriminative geometry preserving projection, as well as tensor structure representation and its closely related adaptive tensor distance metric. We begin with a review of a few tensor operations [10,16].

3.1. Review of tensor operations

Assume that a data sample is represented as an $n$th-order tensor $X \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_n}$, which is addressed by $n$ indices $m_i, i = 1, \ldots, n$, with each $m_i$ addressing the $i$-mode of $X$. The inner product of two tensors $X \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_n}$ and $Y \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_n}$ is defined as $\langle X, Y \rangle = \sum_{l_1, \ldots, l_n} X_{l_1 \cdots l_n} Y_{l_1 \cdots l_n}$; consequently, the norm of a tensor $X$ is defined to be $\|X\| = \sqrt{\langle X, X \rangle}$ and the tensor distance between tensors $X$ and $Y$ is defined as $D(X, Y) = \|X - Y\|$. The $k$-mode product of a tensor $X$ and a matrix $U \in \mathbb{R}^{m_k \times m_k}$ is defined as $Y = X \times_k U$, where

$$Y_{i_1 \cdots i_{k-1} i_{k+1} \cdots i_n} = \sum_{j=1}^{m_k} X_{i_1 \cdots i_{k-1} j i_{k+1} \cdots i_n} U_{j, i_k}, j = 1, \ldots, m_k \quad (6)$$

The $k$-mode unfolding of a tensor $X \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_n}$ into a matrix $X^k \in \mathbb{R}^{m_k \times \prod_{i \neq k} m_i}$, that is, $X^k \leftarrow_k X$ is defined as $X^k_{i,j} = X_{i_1 \cdots i_k \cdots i_n}$ where $j = 1 + \sum_{i=1}^k (i_{i}-1) \prod_{o=1}^{i-1} m_o$. For example, if $X \in \mathbb{R}^{4 \times 5 \times 6}$ is a third-order tensor, then we can obtain $X^1 \in \mathbb{R}^{4 \times 30}$, $X^2 \in \mathbb{R}^{5 \times 24}$, and $X^3 \in \mathbb{R}^{6 \times 20}$.

3.2. Data-adaptive tensor distance

Effective distance function plays a key role in tensor-based dimensionality reduction techniques and a number of research efforts have shown that the performance of the tensor-based dimensionality reduction algorithms not only depends on embedding strategies, but is also closely related to distance metrics. The most commonly used distance metrics are Euclidean distance in tensor-based techniques. However, the orthogonality assumption of Euclidean distance ignores the relationships among different coordinates for high-order tensor data, such as the spatial relationships of pixels in images. To alleviate the orthogonality assumption and enhance the performance of the tensor-based dimensionality reduction algorithms, we adopt the data-adaptive tensor distance metric proposed by Liu et al. [14].

Given a high-order tensor data $X \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_n}$, let $x$ denote the vector form representation of $X$ . Thus, the element $X_{i_1 \cdots i_k}$, $1 \leq i_j \leq m_j$, $1 \leq j \leq N$ in $X$ is corresponding to the $l$th element in $x$, i.e., $x_l$, where

$$l = i_1 + \sum_{j=2}^N (i_j - 1) \prod_{o=1}^{j-1} m_o \quad (2 \leq l \leq N).$$

Then, following the suggestions in [14], the tensor distance between two tensors $X$ and $Y$ can be defined as follows:

$$d_{ad}(X, Y) = \sqrt{\sum_{l=1}^L g_{lm} (x_l - y_l)(x_m - y_m)} \quad (7)$$

$$= \sqrt{(x - y)^T G (x - y)}$$

where $g_{lm}$ denotes the metric efficient and $G$ denotes the metric matrix. The key issue now is the choice of $G$ so that the tensor distance induced by the data-dependent
metric matrix is motivated with respect to the intrinsic relationships between different coordinates for high-order tensor data.

In order to model the intrinsic relationships between different coordinates, Wang et al. [15] have already proved the following conclusion: for image data, if the metric coefficients depend properly on distances of pixel locations, then the obtained distance metric can effectively reflect the spatial relationships between pixels. Following this idea, Liu et al. [14] designed the following metric matrix \( G \):

\[
g_{lm} = \frac{1}{2\sigma^2} \exp \left( -\frac{\|p_l - p_m\|^2}{2\sigma^2} \right) \tag{8}
\]

where \( \sigma \) is a positive regularization parameter and is empirically set as 1 for his simplicity, and \( \|p_l - p_m\|^2 \) denotes the location distance between \( x_l \) (corresponding to \( X_{ij \cdots k} \)) and \( x_m \) (corresponding to \( X_{ij \cdots k}^\prime \)), which can be defined as follows:

\[
\|p_l - p_m\|^2 = \sqrt{\sum_{k=1}^{N} (i_k - i_k^\prime)^2} \tag{9}
\]

Once obtaining the metric matrix \( G \) according to (8), we eventually get the following data-adaptive tensor distance metric:

\[
d_G(X,Y) = \sqrt{\sum_{i=1}^{N} g_{nm}(x_i - y_m)(x_n - y_n)} = \frac{1}{2\sigma^2} \sum_{l=1}^{N} \exp \left( -\frac{\|p_l - p_m\|^2}{2\sigma^2} \right) (x_l - y_m)(x_n - y_n) \tag{10}
\]

It is important to note that the above data-adaptive tensor distance metric can be reduced to the traditional Euclidean distance when \( G = I \). Therefore, the traditional Euclidean distance is a special case of the proposed data-adaptive tensor distance metric.

### 3.3. Distance adaptive tensor discriminative geometry preserving projection

In order to preserve the intrinsic tensor structure of high-order face image data, a natural way is to perform dimensionality reduction in their original high-order tensor space. In the following, we discuss how to tensorize the discriminative geometry preserving projection which gives rise to a distance adaptive tensor discriminative geometry preserving projection (DATDGPP) algorithm for face recognition.

Given \( n \) face image samples \( X_1, X_2, \ldots, X_n \) in the tensor space \( \mathbb{R}^{{m_1 \times m_2 \times \cdots \times m_r}} \) and each \( X_j \) (\( j = 1, \ldots, n \)) belongs to the class indexed as \( c_j \in \{1, 2, \ldots, n_\} \), DATDGPP aims to find \( N \) transformation matrices \( U_i \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_r} \) that maps each high-dimensional data point \( X_j \) (\( j = 1, \ldots, n \)) to a lower-dimensional data point \( Y_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_r} \) \((j = 1, \ldots, N)\) by \( Y_j = X_j U_1 U_2 \cdots U_N \) such that \( Y_j \) represents \( X_j \) well in terms of discrimination preservation and local geometry preservation.

Similar to the original vector-based DGPP method, to preserve both the local geometry and the discriminative information in the low-dimensional feature subspace, the optimal objective function of DATDGPP is defined as follows:

\[
\arg \max_{U \in \mathbb{R}^{mx}} \left\{ \sum_{i=1}^{N} H_i \left\|X_{li} U_1 U_2 \cdots U_N - X_{li} U_1 U_2 \cdots U_N U_i\right\|_2 \right\} \tag{11}
\]

with the constraint

\[
U_k^T U_k = I \text{ } (k = 1, 2, \ldots, N) \tag{12}
\]

where the weighting factor \( H_i \) encodes both the distance weighting information and the class label information in the high-order tensor space according to

\[
H_i = \left\{ \begin{array}{ll}
\frac{\exp(-d^2_{ij}(X_i, X_j) / \delta^2) (1/n - 1/n_i)}{1/n}, & \text{if } c_i = c_j = l \\
1/n, & \text{if } c_i = c_j
\end{array} \right. \tag{13}
\]

where \( \delta \) is a positive constant and its setting can be referred to in [5]. \( d^2_{ij}(X_i, X_j) \) denotes the tensor distance metric between two tensor \( X_i \) and \( X_j \) defined in (10), and \( W_{ij} \) is the reconstruction weight in the tensor space, which can be obtained by solving the following reconstruction error minimization problem:

\[
W_{ij} = \arg \min_{W_{ij}} \sum_{i=1}^{n} \left\|X_i - \sum_{j=1}^{N} W_{ij} X_j\right\|^2 \tag{14}
\]
Similar to the original DGPP method, by imposing
$$\sum_{j \neq i} W_{ij} = 1$$ and \( W_{ij} = 0 \) for \( c_i \neq c_j \) on the optimal problem (14), we can easily obtain
$$W_i = \sum_{p} C^{-1}_{i,p} \sum_{p,q} C^{-1}_{p,q} \sum_{p,q} C_{p,q}$$
wherein
$$C_{p,q} = (X_i - X_p)^T (X_i - X_q)$$ is the local Gram matrix in the tensor space and \( c_p = c_q = c_i \).

Because of the difficulty in computing the optimal \( U_1, U_2, \ldots, U_N \) simultaneously for (11), we propose an iterative algorithm inspired by previous research [10]. In this algorithm, we first initialize the transformation matrices \( U_1, U_2, \ldots, U_{k-1}, U_{k+1}, \ldots, U_N \), then the optimal transformation matrix \( U_k \) can be solved from the other projection matrices \( U_1, U_2, \ldots, U_{k-1}, U_{k+1}, \ldots, U_N \) by

$$U_k^{\ast} = \arg \max_{U_k} \left\{ \sum_{i,j} W_{ij} \| U_i X_i - U_j X_j \|^2 - \sum_{i,j} W_{ij} U_i X_i - \sum_{i,j} W_{ij} U_j X_j \right\}$$

$$= \arg \max_{U_k} \left\{ \text{Tr} \left( U_k^T X_k (D - H^T) (D - H^T)^T (X_k)^T U_k \right) - \text{Tr} \left( U_k^T X_k (I - W^T) (I - W^T)^T (X_k)^T U_k \right) \right\}$$

$$= \arg \max_{U_k} \left\{ \text{Tr} \left( U_k^T X_k (D - H^T) (D - H^T)^T (X_k)^T U_k \right) \right\}$$

with the constraint

$$U_k^T U_k = I$$

(16)

where

$$X_k^i = X_i \times_{1} U_1 \times_{k-1} U_{k-1} \times_{k+1} U_{k+1} \cdots \times_{N} U_N$$ and \( X_k^i \) is the \( k \)-mode unfolding of a tensor \( X \).

Since the optimal problem of (15) subject to (16) can be approximately solved by following the standard eigenvalue decomposition problem:

$$X_i \left( (D - H^T) (D - H^T)^T - (I - W^T) (D - W^T)^T \right) (X_i)^T U_i = \lambda_i U_i$$

(17)

then the optimal transformation matrix \( U_k \) can also be regarded as the eigenvectors of the matrix

$$X_i \left( (D - H^T) (D - H^T)^T - (I - W^T) (D - W^T)^T \right) (X_i)^T$$

associated with the largest eigenvalues. Therefore, the optimal objective function of (11) can be solved by iteratively optimizing different transformation matrices while fixing the other transformation matrices.

The resolving procedure of the above iteration algorithm can be described as follows: first, we fix \( U_2, \ldots, U_n \), and obtain the optimal transformation matrix \( U_1 \) by computing the largest eigenvectors of

$$X_1 \left( (D - H^T) (D - H^T)^T - (I - W^T) (D - W^T)^T \right) (X_1)^T.$$

Then we fix \( U_1, U_3, \ldots, U_n \), and obtain the optimal projection matrix \( U_2 \) by computing the largest eigenvectors of

$$X_2 \left( (D - H^T) (D - H^T)^T - (I - W^T) (D - W^T)^T \right) (X_2)^T.$$

The rest can be computed by analogy. Finally, we can obtain the optimal projection matrix \( U_n \) by fixing \( U_1, U_2, \ldots, U_{n-1} \). We repeat the above iteration procedure until algorithm converges. The algorithmic procedure of DATDGGP can be summarized as follows:

**Input:**
- Face image sample set \( X_1, X_2, \ldots, X_n \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_n} \) and their class labels \( c_1, c_2, \ldots, c_n \in \{1, 2, \ldots, n_i\} \),
- the maximal iteration number \( T_{\text{max}} \), and the final low-dimensions \( m'_1, m'_2, \ldots, m'_N \).

**Output:**
- Transformation matrices \( U_k \in \mathbb{R}^{m_i \times n_i} \) \( (k = 1, 2, \ldots, N) \).
- Compute the k-mode unfolding of a tensor \( X_k^i \), i.e.,

$$X_k^i = X_i \times_{1} U_1^i \times_{k-1} U_{k-1}^i \times_{k+1} U_{k+1}^i \cdots \times_{N} U_N^i;$$
- Compute

$$Q^k = X_k^i \left( (D - H^T) (D - H^T)^T - (I - W^T) (D - W^T)^T \right) (X_k^i)^T;$$
- Solve the eigenproblem: \( Q^k U_k^i = \lambda_k U_k^i \), \( U_k^i \in \mathbb{R}^{m_i \times n_i} \);
12: If \( t > 2 \) and \( \| U_k^i - U_k^{i-1} \| < m_k^i m_k \epsilon \),
\( k = 1, 2, \cdots, N \), break;
13: Out the final transformation matrices \( U_k = U_k^i \in \mathbb{R}^{m_k \times m_k^i}, k = 1, 2, \cdots, N \).

In Algorithm 1, we described in detail the procedure for learning the transformation matrices in an iterative manner. In the following, we analyse the computational complexity of DATDGPP and prove the convergence of the proposed iteration resolving algorithm.

To easily analyse the computational complexity, we assume that the high-order tensor data have the same size in each dimension, i.e., \( m_k = m, i = 1, 2, \cdots, N \). Therefore, the time complexity of the original vector-based DGPP algorithm is \( O(m^3) \). However, for our proposed DATDGPP algorithm, the time complexity in computing the scatter matrices is \( O(N \times m^{K+1}) \) and the time complexity for solving the eigenvalue decomposition is \( O(N \times m) \) for each loop, which is much lower than the original vector-based DGPP algorithm. Although the DATDGPP algorithm has no closed-form solution and many loops are needed to reach convergence, it still runs much faster than the original DGPP algorithm due to its simplicity in each iteration.

**Theorem 1.** The iterative algorithm for DATDGPP will converge.

**Proof:** to prove the convergence of the proposed iterative algorithm, we only need to prove that the objective function is non-decreasing and has an upper bound. To conveniently describe the proof procedure, we rewrite the objective function in (11) as follows:

\[
J(U_1^i, U_2^i, \cdots, U_N^i) = \arg \max \left\{ \frac{\sum_{i,j} m_i \| X_i U_j \| - \sum_{i,j} m_i \| X_i - U_j \|}{\sum_{i,j} m_i \| X_i U_j \| - \sum_{i,j} m_i \| X_i - U_j \|} \right\}
\]

In fact, in each iteration of Algorithm 1, since each update of transformation matrix \( U_k \) maximizes the current objective function while other matrices \( U_i \) \( (i = 1, 2, \cdots, k - 1, k + 1, \cdots, N) \) are fixed, the objective function in (11) is non-decreasing, i.e.,

\[
J(U_1^i, U_2^i, \cdots, U_N^i) \leq J(U_1^{i+1}, U_2^{i+1}, \cdots, U_N^{i+1}) \leq J(U_1^{i+2}, U_2^{i+2}, \cdots, U_N^{i+2}) \leq \cdots \leq J(U_1^{i+t}, U_2^{i+t}, \cdots, U_N^{i+t})
\]

In addition, an upper bound exists for the objective function in (11), i.e.,

\[
J(U_1^i, U_2^i, \cdots, U_N^i) \leq \| D - H \| (D - H')' - (I - W') (D - W')'
\]

Therefore, the objective function \( J(U_1^i, U_2^i, \cdots, U_N^i) \) is non-decreasing and has an upper bound, i.e., the proposed iterative algorithm for DATDGPP will finally converge.

It is worth noting that our proposed DATDGPP algorithm can be reduced to the tensor DGPP(TDGPP) algorithm which adopts the traditional Euclidean distance metric when the tensor distance \( d_{id}(X_i, X_j) \) in (13) adopts the traditional Euclidean distance metric, i.e., \( d_{id}(X_i, X_j) = \| X_i - X_j \| \). In fact, we have shown that the traditional Euclidean distance is a special case of our proposed data-adaptive tensor distance metric in the last paragraph of Section 3.2. In addition, since the TDGPP algorithm is similar to our proposed DATDGPP algorithm, except for the tensor distance computation in (13), we omit the detailed algorithmic procedure of TDGPP for simplicity.

### 3.4. Face recognition using DATDGPP

Once the transformation matrix of DATDGPP is obtained, we can apply it to project the face images into a low-dimensional subspace. Then the face recognition problem becomes a pattern classification task and the traditional classification algorithms can be applied in the low-dimensional subspace. In this paper, we apply the nearest-neighbour classifier because of its simplicity.

With the learned transformation matrices \( \left( U_k^i \right)_{i,k} \) in Algorithm 1, the low-dimensional feature representation of the face training sample \( X_i \) \( (i = 1, 2, \cdots, n) \), can be computed according to

\[
Y_i = X_i \times_1 U_1^i \times_2 U_2^i \cdots \times_n U_N^i .
\]  

(18)

When a new testing face image data \( X \) arrives, we can first obtain its low-dimensional feature representation according to

\[
Y = X \times_1 U_1^i \times_2 U_2^i \cdots \times_n U_N^i .
\]  

(19)

Then the class label of \( X \) is predicted to be that of a face sample whose low-dimensional feature representation is nearest to \( Y \), that is

\[
i' = \arg \min \| Y - Y_i \|
\]  

(16)
and the face image $X$ is classified to the class $c_i$, where $\|Y_i - Y\|$ denotes the tensor distance between the two tensors $Y_i$ and $Y$.

4. Experimental results

In this section we first compare our proposed DATDGPP algorithm with the original vector-based DGPP algorithm and the tensor DGPP(TDGPP) algorithm which adopts the traditional Euclidean distance metric. Then, we compare the proposed DATDGPP algorithm with tensor principal component analysis (TPCA) [12], tensor linear discriminant analysis (TLDA) [10], tensor locality preserving projection (TLPP) [13], and tensor marginal Fisher analysis (TMFA) [8] - four of the most popular tensor-based dimensionality reduction algorithms in face recognition.

4.1. Face databases

Our empirical study on face recognition was conducted on three real-world face databases: the Yale database, the Olivetti Research Laboratory (ORL) database and the PIE (pose, illumination and expression) database from CMU. In all the experiments, preprocessing to locate the faces was applied. Original images were manually aligned according to the eye position, cropped and normalized to the resolution of $112 \times 92$. In addition, in order to reduce the influence of some extreme illumination, histogram equilibrium was applied in the preprocessing step. Face images in the three databases are naturally second-order tensors.

In all the experiments the recognition process has three steps. First, we calculate the face subspace from the training samples; then the new face image to be identified is projected into $d$-dimensional subspace for the vector-based algorithms or $(d \times 1)$-dimensional subspace for the tensor-based algorithms; finally, the new face image is identified by the nearest-neighbour classifier in the low-dimensional subspace.

The Yale database (http://cvc.yale.edu/projects/yalefaces/yalefaces.html) contains 165 front view face images of 15 individuals. Eleven images were collected from each individual with varying facial expressions and configurations. Eleven sample images of one person from the Yale database are shown in Figure 1.

The ORL face database (http://www.uk.research.att.com/facedatabase.html) contains 400 images of 40 individuals. Some images were captured at different times and have different variations including expression, lighting and facial details. The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees. Ten sample images of one person from the ORL database are shown in Figure 2.

The CMU PIE face database [17] contains 41,368 face images of 68 subjects in total. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. In this work, we use a subset of five near front poses (C05, C07, C09, C27 and C29) and five illuminations (indexed as 08 and 11). Therefore, each person has ten images. Figure 3 shows ten example images of one person from the PIE database.

4.2. Results

We conducted two experiments on each database. In each experiment, the face image set was randomly partitioned into the training and testing set with different numbers. For ease of representation, the experiments are named as $Gm/Pn$ which means that $m$ images per person are randomly selected for training and the remaining $n$ images for testing. We repeat each experiment 20 times on randomly selected training and testing sets and report the average results.

In the first experiments we compare the recognition accuracy and running time of DGPP, TDGPP and DATDGPP algorithms under different training and testing partitions. In general, the performance of all these algorithms varies with the number of dimensions. We show the maximal average recognition accuracy as well as the optimal reduced dimension and the running time obtained by DGPP, TDGPP and DATDGPP algorithms on the three databases in Tables 1-6. In the second experiment we also compare our proposed DATDGPP algorithm with the other four representative tensor-based dimensionality reduction algorithms: TPCA, TLDA, TLPP and TMFA. Tables 1, 3, 5 report the maximal average recognition accuracies and the corresponding optimal reduced dimensions of the TPCA, TLDA, TLPP, TMFA and DATDGPP algorithms on the three databases. Due to space limitations, we omit the plots of recognition accuracy versus variation of reduced dimensions on the three databases.
Table 1. Comparisons of maximal average recognition accuracy (in percent) as well as the optimal reduced dimension on the Yale database.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>G2/P9</th>
<th>G3/P8</th>
<th>G4/P7</th>
<th>G5/P6</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGPP</td>
<td>63.5(12)</td>
<td>67.8(20)</td>
<td>74.6(15)</td>
<td>77.5(15)</td>
</tr>
<tr>
<td>TDGPP</td>
<td>86.4(14×14)</td>
<td>88.2(13×13)</td>
<td>90.3(10×10)</td>
<td>92.7(10×10)</td>
</tr>
<tr>
<td>TPCA</td>
<td>61.3(29×29)</td>
<td>69.7(27×27)</td>
<td>76.9(23×23)</td>
<td>81.2(23×23)</td>
</tr>
<tr>
<td>TLDA</td>
<td>70.4(26×26)</td>
<td>75.4(22×22)</td>
<td>82.6(18×18)</td>
<td>85.1(18×18)</td>
</tr>
<tr>
<td>TLPP</td>
<td>73.2(20×20)</td>
<td>78.6(20×20)</td>
<td>83.5(17×17)</td>
<td>87.3(16×16)</td>
</tr>
<tr>
<td>TMFA</td>
<td>84.7(16×16)</td>
<td>86.2(14×14)</td>
<td>89.3(12×12)</td>
<td>91.2(12×12)</td>
</tr>
<tr>
<td>DATDGPP</td>
<td>89.1(12×12)</td>
<td>90.5(12×12)</td>
<td>93.4(8×8)</td>
<td>94.6(8×8)</td>
</tr>
</tbody>
</table>

Table 2. Comparisons of running time (second) on the Yale database.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>G2/P9</th>
<th>G3/P8</th>
<th>G4/P7</th>
<th>G5/P6</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGPP</td>
<td>2.58</td>
<td>2.79</td>
<td>3.62</td>
<td>4.34</td>
</tr>
<tr>
<td>TDGPP</td>
<td>0.32</td>
<td>0.38</td>
<td>0.45</td>
<td>0.49</td>
</tr>
<tr>
<td>DATDGPP</td>
<td>0.31</td>
<td>0.36</td>
<td>0.42</td>
<td>0.47</td>
</tr>
</tbody>
</table>

Table 3. Comparisons of maximal average recognition accuracy (in percent) as well as the optimal reduced dimension on the ORL database.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>G2/P8</th>
<th>G3/P7</th>
<th>G4/P6</th>
<th>G5/P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGPP</td>
<td>87.2(37)</td>
<td>90.4(39)</td>
<td>92.7(46)</td>
<td>96.4(42)</td>
</tr>
<tr>
<td>TDGPP</td>
<td>93.5(8×8)</td>
<td>95.8(6×6)</td>
<td>96.5(6×6)</td>
<td>98.7(6×6)</td>
</tr>
<tr>
<td>TPCA</td>
<td>77.8(30×30)</td>
<td>86.9(27×27)</td>
<td>90.7(26×26)</td>
<td>93.2(26×26)</td>
</tr>
<tr>
<td>TLDA</td>
<td>87.2(10×10)</td>
<td>91.6(8×8)</td>
<td>94.3(8×8)</td>
<td>97.8(8×8)</td>
</tr>
<tr>
<td>TLPP</td>
<td>86.3(14×14)</td>
<td>90.7(12×12)</td>
<td>93.8(12×12)</td>
<td>97.4(12×12)</td>
</tr>
<tr>
<td>TMFA</td>
<td>90.5(12×12)</td>
<td>95.3(10×10)</td>
<td>96.4(10×10)</td>
<td>98.5(10×10)</td>
</tr>
<tr>
<td>DATDGPP</td>
<td>95.6(8×8)</td>
<td>97.5(6×6)</td>
<td>98.8(6×6)</td>
<td>99.6(6×6)</td>
</tr>
</tbody>
</table>

Table 4. Comparisons of running time (second) on the ORL database.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>G2/P8</th>
<th>G3/P7</th>
<th>G4/P6</th>
<th>G5/P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGPP</td>
<td>2.36</td>
<td>2.65</td>
<td>3.49</td>
<td>4.25</td>
</tr>
<tr>
<td>TDGPP</td>
<td>0.29</td>
<td>0.34</td>
<td>0.42</td>
<td>0.46</td>
</tr>
<tr>
<td>DATDGPP</td>
<td>0.27</td>
<td>0.33</td>
<td>0.40</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 5. Comparisons of maximal average recognition accuracy (in percent) as well as the optimal reduced dimension on the CMU PIE database.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>G2/P8</th>
<th>G3/P7</th>
<th>G4/P6</th>
<th>G5/P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGPP</td>
<td>64.7(50)</td>
<td>72.5(54)</td>
<td>85.6(68)</td>
<td>88.3(70)</td>
</tr>
<tr>
<td>TDGPP</td>
<td>78.6(12×12)</td>
<td>83.8(10×10)</td>
<td>87.3(10×10)</td>
<td>90.5(10×10)</td>
</tr>
<tr>
<td>TPCA</td>
<td>56.5(27×27)</td>
<td>67.6(25×25)</td>
<td>75.8(23×23)</td>
<td>82.4(23×23)</td>
</tr>
<tr>
<td>TLDA</td>
<td>64.7(10×10)</td>
<td>73.4(8×8)</td>
<td>80.9(8×8)</td>
<td>86.1(8×8)</td>
</tr>
<tr>
<td>TLPP</td>
<td>65.9(16×16)</td>
<td>74.5(14×14)</td>
<td>81.7(14×14)</td>
<td>86.5(14×14)</td>
</tr>
<tr>
<td>TMFA</td>
<td>76.4(14×14)</td>
<td>81.3(12×12)</td>
<td>85.2(12×12)</td>
<td>89.3(10×10)</td>
</tr>
<tr>
<td>DATDGPP</td>
<td>82.3(10×10)</td>
<td>85.2(9×9)</td>
<td>89.6(9×9)</td>
<td>91.8(9×9)</td>
</tr>
</tbody>
</table>

Table 6. Comparisons of running time (second) on the CMU PIE database.
The main observations from the above performance comparisons include:

1) The tensor-based DGPP algorithms (i.e., TDGPP and DATDGPP) perform much better than the original vector-based DGPP algorithm, which demonstrates that the tensor structure representation can effectively use the correlation among different coordinates of face image data to enhance the recognition performance.

2) Our proposed DATDGPP algorithm consistently outperforms the traditional Euclidean distance-based TDGPP algorithm. This is because the orthogonality assumption of the traditional Euclidean distance may not reflect the real distance between two tensor representation-based face images, while our proposed data-adaptive tensor distance can effectively reflect the spatial relationships between pixels.

3) The running times of the tensor-based DGPP algorithms (i.e., TDGPP and DATDGPP) are much smaller than the original vector-based DGPP algorithm. This result is consistent with the above analysis of computational complexity, i.e., the computational complexities of the tensor-based DGPP algorithms are much lower than the original vector-based DGPP algorithm.

4) The TPCA algorithm performs the worst among the compared algorithms. A possible explanation is as follows: similar to the traditional PCA, the TPCA is also unsupervised, it simply achieves object reconstruction and is not necessarily useful for discrimination and classification tasks.

5) The average performance of TLDA and TLPP are similar. For some databases, TLDA outperforms TLPP, while TLPP is better than TLDA for other databases. This demonstrates that it is hard to evaluate whether local manifold structure or class label information is more important.

6) The TMFA algorithm performs better than TLDA and TLPP algorithms in all experiments. This observation indicates the importance of utilizing both class label information and local manifold structure, as well as describing the separability of different classes with margin criterion. However, the performance of TMFA is still inferior to DATDGPP. The reason is as follows: TMFA extracts discriminative information from only marginal points, although non-marginal points also contain the discriminative information.

7) Our proposed DATDGPP algorithm consistently outperforms TPCA, TLDA, TLPP and TMFA algorithms. The main reason could be attributed to the following fact: first, DATDGPP can precisely model both the intraclass geometry and interclass discrimination; second, the proposed data-adaptive tensor distance can effectively reflect the spatial relationships between pixels. Therefore, our proposed DATDGPP algorithm achieves the best performance among the compared algorithms by the combination of the adaptive tensor distance and discriminative geometry preserving projection strategy.

5. Conclusion and future work

In this paper, we proposed a novel distance adaptive tensor discriminative geometry preserving projection (DATDGPP) algorithm for face recognition. The advantages of DATDGPP are as follows: DATDGPP can preserve the natural tensor structure of face images, it adopts the data-adaptive tensor distance to model the correlation among different coordinates of tensor data and its transformation matrix is of discrimination preservation and local geometry preservation ability. Experiments have demonstrated the effectiveness of the proposed algorithm compared with the original DGPP algorithm and other representative tensor-based dimensionality reduction algorithms. Since the proposed DATDGPP algorithm is a general tensor dimensionality reduction algorithm for high-order tensor data, we plan to apply the algorithm to other high-order tensor data (such as video data) classification in the future.

6. Acknowledgements

This work is supported by the National Natural Science Foundation of China under grant no.70701013, the Natural Science Foundation of Henan Province under grant no.102300410020, the National Science Foundation for Post-doctoral Scientists of China under grant no. 2011M500035 and the Specialized Research Fund for the Doctoral Programme of Higher Education of China under grant no.20110023110002.

7. References