Extracting the optimal dimensionality for local tensor discriminant analysis

Feiping Nie*, Shiming Xiang, Yangqiu Song, Changshui Zhang

Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 2 November 2007
Received in revised form 3 March 2008
Accepted 5 March 2008

Keywords:
Optimal dimensionality
Local scatter
Tensor discriminant analysis
Alternating optimization

ABSTRACT

Supervised dimensionality reduction with tensor representation has attracted great interest in recent years. It has been successfully applied to problems with tensor data, such as image and video recognition tasks. However, in the tensor-based methods, how to select the suitable dimensions is a very important problem. Since the number of possible dimension combinations exponentially increases with respect to the order of tensor, manually selecting the suitable dimensions becomes an impossible task in the case of high-order tensor. In this paper, we aim at solving this important problem and propose an algorithm to extract the optimal dimensionality for local tensor discriminant analysis. Experimental results on a toy example and real-world data validate the effectiveness of the proposed method.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Dimensionality reduction is an important issue when facing high-dimensional data, and supervised dimensionality reduction methods have been proven to be more suitable for classification tasks than those unsupervised methods. Linear discriminant analysis (LDA) is one of the most popular supervised methods and has been widely applied in many high-dimensional classification tasks.

Traditional LDA treats data as vectors, while in many real-world applications, data are represented by tensor form. For example, images can be seen as second-order tensors and videos can be seen as third-order tensors. Treating the tensor data as vectors will result in two obvious disadvantages. First, the underlying spatial structure in tensor data is destroyed. Second, the dimension of the vectors is extremely high, which usually results in the small sample size (SSS) problem [1] and heavy computation burden. Although many approaches have been proposed to solve the SSS problem [2–4], these variants usually discard a subspace such that some important discriminative information may be lost. Comparative study on these approaches to attack the SSS problem can be found in Refs. [5,6].

Recently, a number of algorithms of discriminant analysis with tensor representation have been proposed and attracted great interest [7–11]. Tensor-based discriminant analysis directly treats the data as tensor, and thus effectively avoids the problems derived from treating data as vectors.

However, there are some drawbacks in the tensor-based discriminant analysis. As it is a direct extension of LDA, the limitation of Gaussian assumption in LDA still exists. It may fail to find the optimal discriminative projections if the class distribution is more complex than Gaussian. Many algorithms have been proposed recently to overcome this drawback in LDA [12–16]. The main idea in these methods is to construct the local scatter matrices to replace the traditional scatter matrices, and the limitation of Gaussian distribution in LDA can be effectively removed by using these local scatter matrices.

For the dimension reduction algorithm, how to select the suitable dimensions is a very important problem. In the vector-based methods, several algorithms have been proposed recently to solve this problem [17,18]. However, for the tensor-based methods, this problem becomes more important. The reason is that the number of possible dimension combinations will exponentially increase with respect to the order of tensor, manually selecting the suitable dimensions becomes an impossible task in the case of high-order tensor. Therefore, automatically estimating the optimal dimensions for tensor discriminant analysis is more desired in real applications.

In this paper, we combine the benefits of local scatters and tensor representation and propose the local tensor discriminant analysis for supervised dimensionality reduction. The algorithm effectively overcomes the limitation of Gaussian assumption and the SSS problem. Meanwhile, the optimal dimensions are automatically extracted by algorithm itself to maximize the optimal value of a well-defined criterion.

The rest of this paper is organized as follows: We introduce some operators in tensor algebra in Section 2, and then define a criterion for the proposed algorithm in Section 3. In Section 4 we propose an optimization method to maximize this criterion. A theoretical
2. Tensor algebra

Tensor is a multidimensional array of numbers; the order of a tensor \( A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m} \) is \( m \), and an element of \( A \) is denoted by \( A_{i_1 \ldots i_m} \). We introduce the definitions of tensor operations relevant to this paper as follows [19]:

**Definition 1 (Inner Product).** The inner product of two tensors \( A, B \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m} \) with the same dimension is defined by

\[
\langle A, B \rangle = \sum_{i_1 \ldots i_m} A_{i_1 \ldots i_m} B_{i_1 \ldots i_m}
\]

The norm of tensor \( A \) is defined as \( \| A \| = \sqrt{\langle A, A \rangle} \), and the distance between \( A \) and \( B \) is defined as \( \| A - B \| \). In the case of second-order tensor, the norm is called Frobenius norm and written as \( \| A \|_F \).

**Definition 2 (k-Mode Product).** The \( k \)-mode product of a tensor \( A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m} \) and a matrix \( U \in \mathbb{R}^{k \times I_I (k = 1, 2, \ldots, m)} \) is an \( I_1 \times I_2 \times \cdots \times I_{k-1} \times I_{k+1} \times I_{k+2} \times \cdots \times I_m \) tensor denoted by \( B = A \times_k U \), where the corresponding entities are defined by

\[
B_{i_1 \ldots i_{k-1} i_k i_{k+1} \ldots i_m} = \sum_{i_{k+2} \ldots i_m} A_{i_1 \ldots i_{k-1} i_k i_{k+2} \ldots i_m} U_{i_k}^h
\]

**Definition 3 (k-Mode Unfolding).** The \( k \)-mode unfolding a tensor \( A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m} \) into a matrix \( A(k) \in \mathbb{R}^{k \times \prod_{j \neq k} I_j} \) is defined by

\[
A^{(k)} = A_{i_1 \ldots i_{k-1} i_k i_{k+1} \ldots i_m}, \quad j = 1 + \sum_{l=1, l \neq k}^{m} (i_l - 1) \prod_{o=1, o \neq k}^{m} I_o
\]

3. The criterion for local tensor discriminant analysis

Given \( n \) tensor data \( A_i \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m} (i = 1, \ldots, n) \), each datum is associated with a class label from \( 1, 2, \ldots, c \), \( n_i \) is the number of data points in class \( i \). The task of local tensor discriminant analysis is to find \( m \) optimal orthogonal projection matrices \( U_i \in \mathbb{R}^{k \times I_I (I_I < H_I, i = 1, 2, \ldots, m)} \) such that the separability between classes is maximized under these projections. Given these matrices, any data tensor \( A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m} \) can be transformed to \( B \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m} \) by

\[
B = A \times_k U_1 \cdots \times_m U_m
\]

One key issue for this task is how to define a criterion function to measure the separability between classes. First, we define the local within-class scatter \( s_w \) and the local between-class scatter \( s_b \) as

\[
s_w = \sum_{i,j} W_{ij}^w \| A_i - A_j \|^2
\]

\[
s_b = \sum_{i,j} W_{ij}^b \| A_i - A_j \|^2
\]

where \( W^w \) and \( W^b \) are the local weight matrices. There are a number of approaches to construct them [12–16], all of which have the common property that when \( A_i \) and \( A_j \) are distant, the \( ij \)-th entity in \( W^w \) and \( W^b \) is equal to zero. This property makes \( s_w \) and \( s_b \) focus more on the local information in data. In this paper, we adopt another approach to construct \( W^w \) and \( W^b \), which are directly extended from LDA. First, we define the \( k_w \)-neighbors and \( k_b \)-neighbors as follows:

**Definition 4 (\( k_w \)-Neighbors).** If \( A_i \) belongs to \( A_j \)'s \( k_w \) nearest neighbors with the same class label of \( A_j \), we say \( A_i \) belongs to \( A_j \)'s \( k_w \)-neighbors.

**Definition 5 (\( k_b \)-Neighbors).** If \( A_i \) belongs to \( A_j \)'s \( k_b \) nearest neighbors, whose class labels are different from that of \( A_j \), we say \( A_i \) belongs to \( A_j \)'s \( k_b \)-neighbors.

Then for each data point \( A_i \), we define the within-class neighborhood \( N_w(A_i) \) and the between-class neighborhood \( N_b(A_i) \) as follows:

**Definition 6 (Within-class neighborhood).** If \( A_i \) belongs to \( A_j \)'s \( k_w \)-neighbors or \( A_i \) belongs to \( A_j \)'s \( k_w \)-neighbors, then \( A_j \) is in \( N_w(A_i) \).

**Definition 7 (Between-class neighborhood).** If \( A_i \) belongs to \( A_j \)'s \( k_b \)-neighbors or \( A_i \) belongs to \( A_j \)'s \( k_b \)-neighbors, then \( A_j \) is in \( N_b(A_i) \).

Denote the number of data points in \( N_w(A_i) \) by \( k_{w}(i) \) and the number of data points in \( N_b(A_i) \) by \( k_{b}(i) \). Then \( W^w \) and \( W^b \) are constructed as follows:

\[
W_{ij}^w = \begin{cases} 1/k_w(i) & A_j \in N_w(A_i) \\ 0 & \text{otherwise} \end{cases}
\]

and

\[
W_{ij}^b = \begin{cases} 1/k_w(i) + 1/k_b(i) & A_j \in N_b(A_i) \\ 1/k_w(i) + 1/k_b(i)/k_w(i) & A_j \in N_w(A_i) \\ 0 & \text{otherwise} \end{cases}
\]

From the graph view of LDA [20], the within-class scatter \( s_w \) and the between-class scatter \( s_b \) are defined as the same form as in Eqs. (6) and (7), where \( A_i \) and \( A_j \) are vectors, and \( W^w \) and \( W^b \) are defined based on the non-local other than the local structure of data, i.e.,

\[
W_{ij}^w = \begin{cases} 1/n & A_j \text{ belongs to the same class of } A_i \\ 0 & \text{otherwise} \end{cases}
\]

and

\[
W_{ij}^b = \begin{cases} 1/n & A_j \text{ belongs to the class different from } A_i \\ 1/n - 1/n_{ci} & A_j \text{ belongs to the same class of } A_i \end{cases}
\]

where \( c(i) \) denotes the class label of data point \( A_i \).

Therefore, we can see that if \( k_w(i) = n_{ci} \) and \( k_b(i) = n - n_{ci} \), the local scatters \( s_w \) and \( s_b \) defined here in the first-order tensor case are equivalent to the corresponding scatters defined in LDA.

It is straightforward that larger \( s_b \) and smaller \( s_w \) will result in more easily separating between classes. Therefore, we use the weighted difference between \( s_b \) and \( s_w \) to measure the separability between classes:

\[
J = s_b - \gamma s_w
\]

For the sake of simplicity, we denote the orthogonal projection matrices \( U_i \in \mathbb{R}^{k \times I_I (I_I < H_I, i = 1, 2, \ldots, m)} \) by \( U \). Under the projection
matrices $U$, the local within-class scatter $s^U_w$ and the local between-class scatter $s^U_b$ become:

$$s^U_w = \sum_{ij} W^U_{ij} \|B_i - B_j\|^2$$ \hspace{1cm} (13)

$$s^U_b = \sum_{ij} W^U_{ij} \|B_i - B_j\|^2$$ \hspace{1cm} (14)

where $B_i = A_i \times U_1 \times \cdots \times U_m$, then the criterion in Eq. (12) becomes:

$$\mathcal{J}(U) = s^U_b - \gamma s^U_w$$ \hspace{1cm} (15)

Therefore, based on the criterion in Eq. (15), the optimization problem for the local tensor discriminant analysis can be formulated as:

$$U^* = \arg \max_{1 \leq i \leq h_l(i) = (1, \ldots, m)} \max_{U_l} \mathcal{J}(U)$$ \hspace{1cm} (16)

where $l(i) = (1, \ldots, m)$ denotes an $L_i \times L_i$ identity matrix. Note that the dimensions $L_l(i) = (1, \ldots, m)$ in each projection matrix $U_l$ are also variables in the optimization problem to maximize the criterion in Eq. (15).

Usually the available data are limited and there are noises in data, so we can make a reasonable assumption that the performance should be improved when the dimensionality is reduced by discriminant analysis. Under this assumption, we could suppose that the criterion in Eq. (12) could be zero as the baseline when no dimensionality reduction is performed and would reach a positive value when the dimensions are reduced. Thus, we have:

$$\mathcal{J} = s_b - \gamma s_w = 0 \Rightarrow \gamma = \frac{s_b}{s_w}$$ \hspace{1cm} (17)

There are several interesting properties when the weight $\gamma$ is set to the value as (17). For example, if we substitute $s_{\gamma}(s_b + s_w)$ for $s_b$ in Eqs. (12) and (17), the corresponding criterion in Eq. (15) becomes:

$$\mathcal{J}(U) = (s_{\gamma} + s_{\gamma}) - (s_{\gamma} + s_{\gamma}) = \mathcal{J}(U)$$, and thus the solution in Eq. (16) is unchanged. This invariant property is very similar to the property of traditional LDA. In this way the $W^U_b$ defined in Eq. (9) can be replaced by:

$$W^U_b = \begin{cases} \frac{1}{k_w(l) + k_b(l)}, & A_k \in \mathcal{A} \setminus \mathcal{B}(\mathcal{A}) \cup \mathcal{B}(\mathcal{A}) \\ 0, & \text{otherwise} \end{cases}$$ \hspace{1cm} (18)

Another beneficial property is the scale invariance. Let $W^U$ and $W^b$ be scaled with an arbitrary constant, i.e., let $W^U = c_1 \times W^U$ and $W^b = c_2 \times W^b$, the corresponding criterion in Eq. (15) becomes $\mathcal{J}(U) = c_2 \times s^U_b - (c_2/c_1) \times \gamma \times (c_1 + s^U_w) - c_2 \times \mathcal{J}(U)$, which will not impact the solution in Eq. (16). This scale invariant property is desirable especially when we adopt the difference form (12) as a criterion.

4. The algorithm with alternating optimization

The optimization problem in Eq. (16) has no closed solution. Fortunately, we can use the technique of alternating optimization to obtain a local optimal solution to this problem.

Given $U_1, \ldots, U_{k-1}, U_{k+1}, \ldots, U_m$, denote $B_{j}^{-k}(i = 1, \ldots, n)$ as:

$$B_{j}^{-k} = A_i \times U_1 \times \cdots \times U_{k-1} \times U_{k+1} \times \cdots \times U_m$$ \hspace{1cm} (19)

Then, with the $k$-mode unfolding of $B_{j}^{-k}$, we get $B_{j}^{-k} = B_{j}^{-k}(k)$, and we can derive that:

$$\|B_{j}^{-k} \times U_k\| = \|(B_{j}^{-k})^T U_k\|$$ \hspace{1cm} (20)

Table 1

The algorithm for the local tensor discriminant analysis

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Initialize $U_i(k = 1, \ldots, m)$ as the arbitrary orthogonal matrices.</td>
</tr>
<tr>
<td>2.</td>
<td>Initialize $U_i(k = 1, \ldots, m)$ as the arbitrary orthogonal matrices.</td>
</tr>
<tr>
<td>3.</td>
<td>For $k = 1$ to $T$</td>
</tr>
<tr>
<td>4.</td>
<td>Output: $U_i(k = 1, \ldots, m) \in \mathbb{R}^{d_i}$</td>
</tr>
</tbody>
</table>

Therefore, we have:

$$\mathcal{J}(U) = \sum_{ij} W^U_{ij} \|B_i - B_j\|^2$$ \hspace{1cm} (22)

Substituting Eq. (21) into Eq. (22), we have:

$$\mathcal{J}(U) = \text{tr} \left\{ \left[ U_k^T \sum_{ij} W^U_{ij} \|B_i^{-k}(k) - B_j^{-k}(k)\|^2 \right] U_k \right\}$$ \hspace{1cm} (23)

Denote matrix $G_k$ as:

$$G_k = \sum_{ij} W^U_{ij} \|B_i^{-k}(k) - B_j^{-k}(k)\|^2$$ \hspace{1cm} (24)

then Eq. (12) can be rewritten as:

$$\mathcal{J}(U) = \text{tr} \left( U_k^T G_k U_k \right)$$ \hspace{1cm} (25)

Note that if $U_1, \ldots, U_{k-1}, U_{k+1}, \ldots, U_m$ are known, then $G_k$ is a constant matrix. Therefore, the optimization problem in Eq. (12) can be reformulated to:

$$U_k^* = \arg \max_{U_k \in \mathcal{H}_k} \text{tr}(U_k^T G_k U_k)$$ \hspace{1cm} (26)

It is well known from the result of Rayleigh quotient [21] that under the constraints of $U_k \in \mathbb{R}^{d_k \times k}$ and $U_k^T U_k = I_k$, the maximum of $\text{tr}(U_k^T G_k U_k)$ is the sum of the $k$ largest eigenvalues of $G_k$ when the columns of $U_k$ are the corresponding $k$ largest eigenvectors. Therefore, the optimization problem in Eq. (26) reaches its maximum when $k$ equals the number of positive eigenvalues of $G_k$.

Based on the above analysis, an iterative alternating optimization method can be used to solve the optimization problem in Eq. (15); we describe the detailed procedure in Table 1.

It is not difficult to verify that the algorithm described in Table 1 guarantees that the value of the criterion in Eq. (15) monotonically increases during iterations. Although alternating optimization theoretically can only obtain a locally optimal solution, extensive experiments on image data sets show that the algorithm in the second-order tensor case converges to the same solution (as well as the optimal dimensions of $U$) regardless of the choice of the initial $U$. 

which implies that the algorithm might always converge to the global optimum for the applications to image data sets, while in 2DLDA, the value to be optimized may sway with respect to the iteration number, which may result in the performance varying irregularly.

5. The theoretical bounds of the optimal dimensionality

In this section, we will analyze the bounds of the optimal dimensionality extracted in the algorithm.

Denote the column space of matrix A by \( R(A) \), the rank of matrix A by \( r(A) \), the dimension of space \( V \) by \( \dim(V) \), and the number of positive eigenvalues of matrix G by \( p(G) \). Then we have the following theorem:

**Theorem 5.1.** Suppose matrices A and B both are semi-positive, \( \gamma \) is a positive number, denote \( A - \gamma B \) by \( G \), then \( r(A) - \dim(R(A) \cap R(B)) \leq p(G) \leq r(A) \).

The proof is given in Appendix A.

Note that \( G_k(k = 1, 2, \ldots, m) \) in Eq. (24) can be rewritten as

\[
G_k = G_k^b - \gamma G_k^w
\]  

(27)

where \( G_k^b = \sum_i \|W^b_{ij}(r^k_i(k) - 1)\|B^k_i(k) - B^k_i(k)^T\| \) and \( G_k^w = \sum_i \|W^w_{ij}(r^k_i(k) - 1)\|W^w_{ij}(r^k_i(k) - 1)^T\| \). It can be easily seen that both \( G_k^b \) and \( G_k^w \) are semi-positive based on the definition of \( W^b_{ij} \) and \( W^w_{ij} \) in Eqs. (18) and (8), respectively. Therefore, according to Theorem 5.1 we know the optimal dimensionality extracted in Eq. (26) is bounded by the rank of \( G_k^b \). Obviously, the rank of \( G_k^b \) will be determined by the definition of \( W^b_{ij} \).

If \( W^b_{ij} \) is defined as in LDA (11), then in the first-order tensor case, the rank of the resulted matrix is not larger than \( c - 1 \), where \( c \) is the class number. Therefore, the extracted optimal dimensionality will not be more than \( c - 1 \). When the data of each class are Gaussian distributed with the same covariance, \( c - 1 \) dimensions are sufficient to capture the total discriminative information in data. However, when the data of each class are distributed more complex than Gaussian, \( c - 1 \) dimensions may not be enough. In contrast, if \( W^b_{ij} \) is defined based on the local structure in data as in this paper, the optimal dimensions may exceed \( c - 1 \) to capture the discriminative information when the distribution of data is more complex than Gaussian. It will be verified in the toy example.

6. Experimental results

6.1. Toy example

We present a toy example to demonstrate the effectiveness of our algorithm in the first-order tensor case (denoted as 1D LTDA). In this toy example, we generate three 10-dimensional data sets with two classes, three classes and four classes, respectively. In the first two dimensions, the classes are distributed in concentric circles, while the other eight dimensions are Gaussian noise with large variance. Thus, the optimal projections are the first two dimensions, and the real optimal dimensionality is 2 in all the three data sets. 1D LTDA fails to find the optimal projections for all the three data sets since the data distributions are highly nonlinear and more complex than Gaussian. Fig. 1 shows the results learned by our algorithm in the first-order tensor case. In all of the three data sets, 1D LTDA finds the optimal projections, and the optimal dimensionality determined by the algorithm is exactly the same as the true optimal number, namely 2, regardless of the class number \( c \), while in LDA, the optimal dimensionality determined by it is \( c - 1 \).

6.2. Real-world applications

We evaluated our algorithm in the first-order (1D LTDA) and second-order (2D LTDA) tensor cases on three applications, all of which have the data with second-order tensor representation. The algorithms are compared with two traditional discriminant analysis methods: LDA, MMC [22], two variants with local scalars: MFA [13], LFDA [14], and one two-dimensional version of LDA: 2DLDA [7].

For the first-order tensor-based algorithms, we use PCA as the preprocessing step to eliminate the null space of data covariance matrix. For LDA, MFA and LFDA, due to the singularity problem in it, we further reduce the dimension of data such that the corresponding (local) within-class scatter matrices are nonsingular.

In each experiment, we randomly select several samples per class for training and the remaining samples are used for testing. The average results are reported over 20 random splits. The classification is performed by k-nearest neighbor classifier (\( k = 1 \) in these experiments).

In the experiments, we simply set \( k_\theta \) to 20, and set \( k_w \) to \( t/2 \) for each data set, where \( t \) is the training number per class. We will discuss the influence of these two parameters on classification performance in the next subsection.

The experimental results are reported in Table 2. The bold value indicates the best result among the methods. For LDA and MMC, the projection dimensionality is set to the rank of the between-class scatter matrix. For MFA and LFDA, the results are recorded under different dimensions from 1 to \( h \times w \), where \( h \) and \( w \) are the height and width of the image, respectively. The best results are reported in Table 2. For 2DLDA, as it is hard to select the total dimension combinations for the two projection matrices, we let the two dimensions be the same number. The results are recorded under different dimensions from \( 1^2 \) to \( \min(h, w)^2 \), where \( h \) and \( w \) are the height and width of the image, respectively, and the best result is reported in Table 2, where the ‘dim’ is the corresponding dimensions. For 1D LTDA and 2D LTDA, the dimensions are automatically determined, and the ‘dim’ in Table 2 is the average value of \( m \) and \( \hat{r} \) over 20 random splits, respectively.

6.2.1. Face recognition

The UMIST repository is a multiview face database, consisting of 575 images of 20 people, each covering a wide range of poses from profile to frontal views. The size of each cropped image is \( 112 \times 92 \) with 256 gray levels per pixel [23]. Some images are shown in Fig. 2. We down-sample the size of each image to \( 28 \times 23 \) and no other preprocessing is performed. Two, four or six samples per class are randomly selected for training and the remaining samples for testing.

On this data set, our algorithm with the second-order (2D LTDA) tensor shows the best performances, and 1D LTDA also demonstrates the competitive performances.

6.2.2. Object recognition

The COIL-20 data set [24] consists of images of 20 objects viewed from varying angles at the interval of 5°, resulting in 72 images per object. Some images are shown in Fig. 3. Each image is down-sampled to the size of \( 32 \times 32 \) and we randomly select four, six or eight samples per class for training and the remaining samples for testing.

On this data set, our algorithm with the first-order (1D LTDA) and second-order (2D LTDA) tensor demonstrate the best performances. The 2D LTDA shows excellent results in all the cases and 1D LTDA also demonstrates better results than those of LDA, MMC, MFA, LFDA and 2DLDA.

6.2.3. Music genre classification

In these experiments, we use the ISMIR2004 Audio Description Contest [25] data set for testing our proposed algorithm. There are
Fig. 1. In the first row are the first two dimensions of the original data sets, in the second row are the corresponding results learned by 1D LTDA, and in the third row are the optimal value of criterion (15) under different reduced dimensions by 1D LTDA. Traditional LDA fails to find the optimal subspace in all the three cases and the results are not displayed here.

Table 2
Experimental results in each data set

<table>
<thead>
<tr>
<th>Data set</th>
<th>Method 2 train</th>
<th>4 train</th>
<th>6 train</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc. (%)</td>
<td>Dev. (%)</td>
<td>dim</td>
</tr>
<tr>
<td>UMIST</td>
<td>LDA 62.5</td>
<td>4.1</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>MMC 64.5</td>
<td>4.2</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>MFA 67.6</td>
<td>5.1</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>LFDA 66.8</td>
<td>4.6</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2DLDA 65.5</td>
<td>4.5</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>1D LTDA 67.9</td>
<td>4.2</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>2D LTDA 70.8</td>
<td>4.1</td>
<td>112.4</td>
</tr>
<tr>
<td></td>
<td>4 train</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acc. (%)</td>
<td>Dev. (%)</td>
<td>dim</td>
</tr>
<tr>
<td>Coil20</td>
<td>LDA 77.8</td>
<td>1.9</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>MMC 82.9</td>
<td>1.8</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>MFA 79.6</td>
<td>2.0</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>LFDA 79.1</td>
<td>1.9</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>2DLDA 80.3</td>
<td>2.3</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>1D LTDA 83.5</td>
<td>1.6</td>
<td>24.2</td>
</tr>
<tr>
<td></td>
<td>2D LTDA 90.1</td>
<td>1.8</td>
<td>96.5</td>
</tr>
<tr>
<td></td>
<td>4 train</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acc. (%)</td>
<td>Dev. (%)</td>
<td>dim</td>
</tr>
<tr>
<td>Music</td>
<td>LDA 46.3</td>
<td>5.2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>MMC 51.9</td>
<td>4.1</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>MFA 48.2</td>
<td>4.6</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>LFDA 47.9</td>
<td>5.5</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>2DLDA 48.1</td>
<td>5.6</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>1D LTDA 55.5</td>
<td>3.7</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>2D LTDA 52.1</td>
<td>3.6</td>
<td>68.2</td>
</tr>
</tbody>
</table>

The ‘Acc.’ is the accuracy over 20 random splits. For MFA, LFDA and 2DLDA, the ‘dim’ is the corresponding dimensionality of the best result. For 1D LTDA and 2D LTDA, the ‘dim’ is the average value of $m$ and $1 + r$ over 20 random splits, respectively.
1458 tracks, which are classified into six genres, including classical, electronic, jazz, blues, metal, punk, rock, pop and world. The original tracks are used instead of the segmented small pieces. The stereo audio signals are reduced to mono and down-sampled from 44 to 11 kHz. We use the feature extraction and similarity measurement methods by using the music analysis toolbox [26]. The parameters for feature extraction algorithms are the same as the default setups.

We extracted three kinds of features in this experiment, i.e., fluctuation pattern (FP), spectrum histograms (SH) and periodicity histograms (PH). FP is a feature to describe periodicities of music tracks. Each FP feature can be represented by a $20 \times 60$ matrix. SH is a two-dimensional feature which characterizes the timber of a music track. We get a $20 \times 25$ matrix since there are 20 rows for critical bands and 25 columns for loudness resolution. PH is a feature which describes periodically reoccurring beats of a music track. We finally get a $30 \times 41$ matrix. Fig. 4 shows some samples with FP, SH and PH features, respectively.

On this data set, we randomly select 20 samples per class for training and the remaining samples for testing. Our algorithms also demonstrate the best performances.

6.2.4. Discussions

The scatters defined in LDA and MMC are based on non-local structure in data; thus the optimal dimensionality extracted by them is not more than $c - 1$, where $c$ is the class number. In contrast, MFA, LFDA, 2DLDA and our method can extract more features than LDA and MMC. However, in MFA, LFDA and 2DLDA, the optimal dimensionality cannot be determined by themselves, and thus in the experiments, the results on test error under different reduced dimensions are recorded and the best result is reported between these results. This way is actually infeasible in practice since the test error rate is usually unavailable, while in our algorithm, the optimal dimensionality is automatically calculated, which makes our algorithm more favorable in practice.

For the supervised dimensionality reduction algorithms, there are three main factors that might influence the performance of classification for real-world problems, including the definition of criterion, the definition of the scatters and the representation of the input data. Theoretically, local scatters-based methods are better than non-local scatters-based methods for multimodal data, and tensor-based methods are better than vector-based methods for high-order tensor data. However, these cases do not always occur in real-world applications, which can also be seen from the above experiments. The methods used in the experiments share different kinds of criteria, scatters and representations (see Table 3). For the UMIST data set, the tensor representation, the local scatters and the criterion all have significant effects on the performance. For the COIL-20 data set, the tensor representation and the criterion have significant effects on the performance, while the local scatters demonstrate trivial improvement. For the music data set, the local matrices and the criterion have significant effects on the performance, while the tensor representation does not give an improved performance.

6.3. Performance analysis on the related parameters

In the proposed algorithm, two parameters $k_b$ and $k_w$ are introduced to define the scatters. In this subsection, we present experiments to look into the influence of different $k_b$ and $k_w$ on classification performance. Moreover, in the algorithm, the parameter $\gamma$ in Eq. (15) is automatically calculated according to Eq. (17), and the reduced dimensionality $m$ is theoretically determined to maximize the criterion in Eq. (15). In order to validate the feasibility, we also present experiments to explore the influence of different $\gamma$ and $m$.

In these experiments, two UCI data sets are used, including ionosphere and waveform-21. We randomly select 50 samples per class for training and the remaining samples are used for testing. The average results are reported over 20 random splits. The classification is performed by 1-nearest neighbor classifier.

6.3.1. Performance analysis on $k_b$ and $k_w$

In this experiment, the parameter $\gamma$ and the reduced dimensionality $m$ is automatically determined by the algorithm. We fix $k_w$ to
Fig. 4. Some samples of the three kinds of features for music genre classification. (a) FP feature, (b) SH feature and (c) PH feature.

Table 3
Summary for the different kind of criteria, scatters and representations shared by the methods that are used in the experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>Criterion</th>
<th>Scatters</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ratio</td>
<td>Difference</td>
<td>Non-local</td>
</tr>
<tr>
<td></td>
<td>form</td>
<td>form</td>
<td>scatters</td>
</tr>
<tr>
<td>LDA</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>MMC</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>MFA</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>LFDA</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>2DLDA</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>1D LTDA</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>2D LTDA</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

25, and change $k_w$, or fix $k_b$ to 20, and change $k_w$. The results are shown in Figs. 5 and 6, respectively. The results indicate that $k_b$ and $k_w$ could be insensitive in a large range, which makes them convenient to be tuned. Relative small values of $k_b$ and $k_w$ imply that local scatters are used in the algorithm. The results also reveal that local scatters are more suitable to deal with multimodal data. In general, local scatters are suitable to deal with the data whether they are multimodal or not. Therefore, in practice, we could set $k_b$ and $k_w$ to the relative small values to take advantage of local scatters.

6.3.2. Performance analysis on $\gamma$
In this experiment, we simply set $k_b$ to 20, and set $k_w$ to 25. The reduced dimensionality $m$ is automatically determined by the algorithm. The parameter $\gamma$ is changed from 0 to 0.3 with interval 0.01. The results are shown in Fig. 7. In the 20 runs, the minimum, mean and maximum value of $\gamma$ calculated by Eq. (17) are also shown in the figure. From the results we can see that the value of $\gamma$ has a significant influence on classification performance. Therefore, how to select the value of $\gamma$ is a very important issue. The results show that the $\gamma$ calculated by Eq. (17) lies in the range with the best performance, which indicates that it could be a feasible selection.

6.3.3. Performance analysis on reduced dimensionality $m$
In this experiment, we simply set $k_b$ to 20, and set $k_w$ to 25. The value of $\gamma$ is calculated by Eq. (17). The reduced dimensionality $m$ is changed from 1 to the original dimensionality of data. The results are shown in Fig. 8. In the 20 runs, the minimum, mean and maximum optimal dimensionality $m$ calculated by our algorithm are also shown in the figure. The results indicate that the reduced
Fig. 5. Accuracy rate vs. different $k_b$: (a) ionosphere and (b) waveform-21.

Fig. 6. Accuracy rate vs. different $k_w$: (a) ionosphere and (b) waveform-21.

Fig. 7. Accuracy rate vs. different $\gamma$. The red dashed lines denote the minimum, mean and maximum value of $\gamma$ calculated by (17) in the 20 runs, respectively: (a) ionosphere and (b) waveform-21.

Fig. 8. Accuracy rate vs. reduced dimensionality. The red dashed lines denote the minimum, mean and maximum optimal dimensionality $m$ calculated by our algorithm in the 20 runs, respectively: (a) ionosphere and (b) waveform-21.
dimensionality \( m \) may have a significant influence on classification performance. Like the well-known Fisher’s criterion used in LDA, the theoretical optimal dimensionality calculated in our algorithm is not directly related to the classification rate, and thus may not be always guaranteed to achieve the best classification performance in actual applications. However, as can be seen in the experiment, the optimal dimensionality calculated in our algorithm can be expected to obtain a near-optimal classification performance since it maximizes a well-defined criterion.

7. Conclusions

How to extract the optimal dimensionality is an important problem for dimensionality reduction, especially for the tensor-based dimensionality reduction algorithms. In this paper, we propose a novel supervised dimensionality reduction algorithm for tensor data, where the optimal dimensionality can be extracted automatically. Under the extracted dimensionality, the optimal value of a well-defined criterion is maximized. Our algorithm also combines the benefits of local scatters and tensor representation and effectively overcomes the limitation of Gaussian assumption and the SSS problem. Experiments on the toy example and three different applications demonstrate the effectiveness of our algorithm.

Acknowledgements

This work is supported by NSFC (Grant No. 60721003 and 60675009).

Appendix A. Proof of Theorem 5.1

To prove the theorem, we need two lemmas. The notations are the same as those in Section 5.

Lemma A.1. Suppose \( A \) is symmetrical, \( C \) is a column full rank matrix, then \( p(CAC^T) = p(A) \).

Proof. Let the columns of \( C^\perp \) be a basis for the orthogonal complement of the column space of \( C \), then

\[
CAC^T = \begin{bmatrix} C & C^\perp \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} C & C^\perp \end{bmatrix}^T
\]

so \( CAC^T \) is congruent to \( \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \). According to Sylvester’s law of inertia, \( CAC^T \) and \( \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \) have the same inertia index, therefore, \( p(CAC^T) = p(A) \). \( \square \)

Lemma A.2. Suppose \( A \) is positive, \( B \) is symmetrical, \( C \) is negative. Let

\[
H = \begin{bmatrix} A & D & 0 \\ D^T & B & E \\ 0 & E^T & C \end{bmatrix}
\]

then \( r(A) \leq p(H) \leq r(A) + r(B) \).

Proof. Denote \( F = B - D^T A^{-1} D - E^{-1} E^T \), note that

\[
H = \begin{bmatrix} 1 & 0 & 0 & A & 0 & 0 & A & 0 & 0 \\ D^T A^{-1} & I & E^{-1} & 0 & F & 0 & 0 & 0 & C \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T
\]

Denote

\[
K = \begin{bmatrix} A & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & C \end{bmatrix}
\]

so \( H \) is congruent to \( K \). According to Sylvester’s law of inertia, we have \( p(H) = p(K) = p(A) + p(F) + p(C) \). Therefore, \( r(A) \leq p(H) \leq r(A) + r(B) \). \( \square \)

Proof of Theorem 5.1. Since both matrices \( A \) and \( B \) are semi-positive, there exist two column full rank matrices \( U \) and \( V \) such that \( A = UU^T \) and \( B = VV^T \). Thus, \( G \) can be rewritten as

\[
G = UU^T - \gamma VV^T
\]

Suppose the orthogonal basis of \( R(U) \cap R(V) \) are \( \omega_1, \omega_2, ..., \omega_s \), the orthogonal basis of \( R(U) \) are \( \alpha_1, \alpha_2, ..., \alpha_s, \alpha_{s+1}, ..., \alpha_{s+p} \), and the orthogonal basis of \( R(V) \) are \( \omega_1, \omega_2, ..., \omega_s, \beta_1, \beta_2, ..., \beta_q \). Let \( C = [\alpha_1, \alpha_2, ..., \alpha_p, \alpha_{s+1}, ..., \alpha_{s+p}, \beta_1, \beta_2, ..., \beta_q] \), obviously \( C \) is a column full rank matrix. \( U \) and \( V \) can be written as

\[
U = C \begin{bmatrix} U_1 \\ U_2 \\ 0 \end{bmatrix}, \quad V = C \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}
\]

Therefore,

\[
G = C \begin{bmatrix} U_1 & U_2 \\ 0 & 0 \end{bmatrix} C^T - \gamma C \begin{bmatrix} V_1 & V_2 \\ 0 & 0 \end{bmatrix} C^T = C \begin{bmatrix} U_1 U_1^T & U_1 U_2 \\ 0 & 0 \end{bmatrix} C^T - \gamma C \begin{bmatrix} V_1 V_1^T & V_1 V_2 \\ 0 & 0 \end{bmatrix} C^T
\]

Denote

\[
H = \begin{bmatrix} U_1 U_1^T & 0 \\ U_2 U_1^T & -\gamma V_1 V_1^T \\ 0 & -\gamma V_2 V_1^T \end{bmatrix}
\]

then \( G = CHC^T \).

According to Lemma A.1, we know \( p(G) = p(H) \). Note that \( U_1 U_1^T \) is positive, and \(-\gamma V_2 V_2^T \) is negative, so according to Lemma A.2, we have

\[
p(H) \leq r(U_1 U_1^T) + r(U_2 U_1^T) - \gamma r(V_1 V_1^T) \leq r(U) - r(A)
\]

and

\[
p(H) \geq r(U_1 U_1^T) = r(U) - \dim(R(U) \cap R(V)) = r(A) - \dim(R(A) \cap R(B))
\]

Therefore, \( r(A) - \dim(R(A) \cap R(B)) \leq p(G) \leq r(A) \). \( \square \)

References


About the author—FEIPING NIE received his B.S. degree in Computer Science from North China University of Water Conservancy and Electric Power, Zhengzhou, China, in 2000, and received his M.S. degree in Computer Science from Lanzhou University, Lanzhou, China, in 2003. He is currently a Ph.D. candidate in the Department of Automation, Tsinghua University, Beijing, China. His research interests include machine learning, pattern recognition, data mining and image processing.

About the author—SHIMING XIANG received his B.S. degree from the Department of Mathematics of Chongqing Normal University, China, in 1993, and his M.S. degree from the Department of Mechanics and Mathematics of Chongqing University, China, in 1996, and his Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 2004. He was a post doctor in Department of Automation, Tsinghua University until 2006. He is currently an associate researcher in the Institute of Automation Chinese Academy of Science. His interests include computer vision, pattern recognition, machine learning, etc.

About the author—YANGQIU SONG received his B.S. degree from the Department of Automation, Tsinghua University, China, in 2006. He is currently a Ph.D candidate in the Department of Automation, Tsinghua University. His research interests focus on machine learning and its applications.

About the author—CHANGSHUI ZHANG received his B.S. degree in Mathematics from Peking University, China, in 1986, and his Ph.D. degree from the Department of Automation, Tsinghua University, in 1992. He is currently a professor at the Department of Automation, Tsinghua University. He is an Associate Editor of the journal Pattern Recognition. His interests include artificial intelligence, image processing, pattern recognition, machine learning, evolutionary computation and complex system analysis, etc.