LOCAL DISCRIMINANT EMBEDDING WITH TENSOR REPRESENTATION

Jian Xia, Dit-Yan Yeung & Guang Dai

Department of Computer Science and Engineering, Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{piper, dyyeung, daiguang}@cse.ust.hk

ABSTRACT

We present a subspace learning method, called *Local Discriminant Embedding with Tensor representation* (LDET), that addresses simultaneously the generalization and data representation problems in subspace learning. LDET learns multiple interrelated subspaces for obtaining a lower-dimensional embedding by incorporating both class label information and neighborhood information. By encoding each object as a second- or higher-order tensor, LDET can capture higher-order structures in the data without requiring a large sample size. Extensive empirical studies have been performed to compare LDET with a second- or third-order tensor representation and the original LDE on their face recognition performance. Not only does LDET have a lower computational complexity than LDE, but LDET is also superior to LDE in terms of its recognition accuracy.

Index Terms— Learning systems, Pattern classification, Face recognition

1. INTRODUCTION

Subspace learning, which learns to map data from some input space to a lower-dimensional subspace, plays a very important role in many computer vision and pattern classification problems. Two representative linear subspace learning methods are principal component analysis (PCA) [1] and linear discriminant analysis (LDA) [2]. While PCA seeks to find a low-dimensional representation that minimizes the reconstruction error, LDA uses the label information to find a low-dimensional representation that best separates different classes.

There are two directions along which subspace learning methods can be extended. One direction is to extend the generalization ability of subspace learning methods, while another direction is to enhance their data representation ability. Along the first direction, some examples include independent component analysis (ICA) [3] proposed as a generalization of PCA to take into account higher-order statistical dependencies, kernel PCA (KPCA) [4] and kernel Fisher discriminant analysis (KFD) [5] proposed as nonlinear extensions to their linear counterparts based on the kernel approach, an enhanced LDA model [6] and direct LDA [7] as improvements to the original LDA, and locality preserving projection (LPP) [8] as an extension of PCA that can preserve the neighborhood structure of the data. More recently, a new subspace learning method called local discriminant embedding (LDE) [9] was proposed. LDE makes use of both the neighborhood relationships between data points and the class label information to obtain a lower-dimensional embedding. Unlike LDA and related methods, the discrimination ability of LDE does not strongly depend on the data distribution, such as the Gaussian assumption. Moreover, unlike many manifold learning methods such as Isomap [10] and locally linear embedding (LLE) [11], LDE uses label information to find the embedding and can naturally handle new test data in classification applications.

Along the second direction, some recent subspace learning methods represent each object as a two-dimensional (2-D) matrix rather than a one-dimensional (1-D) vector, e.g., [12, 13]. For applications in which the available data are scarce, a vector representation can make the *curse of dimensionality problem* (and hence the *small sample size problem*) a lot more serious. Moreover, for objects such as images, a 1-D representation ignores higher-order structures in the data. Working directly on 2-D image objects allows principal features in the rows or columns of the images to be found, resulting in substantial reduction in the dimensionality of some computational problems such as the eigenvalue problem. Recently, further generalization was proposed to represent each object as a general tensor of second or higher order, such as PCA with tensor representation [14] and LPP with tensor representation [15].

In this paper, we attempt to address both the generalization and data representation problems simultaneously. Specifically, we propose to reformulate LDE so that it can work directly on a tensor representation. We refer to this new method as *Local Discriminant Embedding with Tensor representation* (LDET). Based on face recognition experiments, we empirically compare the classification performance of our LDET algorithm and the original LDE algorithm.

2. BRIEF REVIEW OF LDE

Let $\mathcal{M}$ be a manifold embedded in $\mathbb{R}^d$. Given a set of n data points $\{\mathbf{x}_i \mid \mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n \subset \mathcal{M}$, which may also be written as a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n\times d}$, LDE seeks to find a projection matrix $\mathbf{V} \in \mathbb{R}^{d\times \ell}$ that maps each data point $\mathbf{x}_i$ in the d-dimensional space to a vector $\mathbf{z}_i$ in the ℓ-dimensional space. With an abuse of notation, the linear projection may also be expressed in the form of a mapping $\mathbf{V} : \mathbf{x}_i \in \mathbb{R}^d \to \mathbf{z}_i = \mathbf{V}^T\mathbf{x}_i \in \mathbb{R}^\ell$, where $\ell \ll d$.

Assume that each data point $\mathbf{x}_i \in \mathbf{X}$ belongs to one of h classes, with the corresponding class label $y_i \in \{1, \ldots, h\}$. Based on the assumption that any subset of data points belonging to the same class lies in a submanifold of $\mathcal{M}$, the LDE algorithm aims to find the optimal projection matrix $\mathbf{V}$ for the embedding by integrating the class label information of the data points and the neighborhood information between data points. The goal is to preserve the within-class neighborhood relationship while dissociating the submanifolds for different classes from each other. In so doing, classification based on the nearest neighbor criterion in the embedding subspace is expected to be able to predict the class labels of unlabeled test points more reliably.

The within-class neighborhood relationship is represented by a within-class neighborhood graph G while the between-class neighborhood relationship is represented by a between-class neighborhood graph G'. To construct G over all data points, we consider each pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$ that belong to the same class, i.e., $y_i = y_j$. An edge is added between $\mathbf{x}_i$ and $\mathbf{x}_j$ if $\mathbf{x}_j$ is one of the k nearest neighbors of $\mathbf{x}_i$, or vice versa. Alternatively, one may also consider the ε-ball implementation. For G', we instead consider each pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$ that belong to different classes, i.e., $y_i \neq y_j$. Edges are added in a way similar to that in G.

With these two neighborhood graphs, we then define the corresponding affinity matrices $\mathbf{W}$ and $\mathbf{W}'$. For the affinity matrix $\mathbf{W}$ of G, each element w_{ij} represents the weight of the edge between $\mathbf{x}_i$ and $\mathbf{x}_j$ and is given by

$$w_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}\right), \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm and σ is a positive constant. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are not connected, we set w_{ij} to 0. Apparently $\mathbf{W}$ is a sparse, symmetric matrix. The affinity matrix $\mathbf{W}'$ of G' can be computed in the same way.

Unlike PCA and LDA, LDE takes into account the local relationships between neighboring data points while incorporating the class information. Specifically, in the low-dimensional embedding subspace, we want to keep the neighboring points close to each other if they belong to the same class and prevent data points from other classes to enter the neighborhood of a data point. To incorporate these requirements, LDE solves the following optimization problem:

$$\begin{aligned} &\text{Maximize } J(\mathbf{V}) = \sum_{i,j} \|\mathbf{V}^T\mathbf{x}_i - \mathbf{V}^T\mathbf{x}_j\|^2 w'_{ij}, \\ &\text{subject to } \sum_{i,j} \|\mathbf{V}^T\mathbf{x}_i - \mathbf{V}^T\mathbf{x}_j\|^2 w_{ij} = 1. \end{aligned} \tag{2}$$

where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_\ell] \in \mathbb{R}^{d\times\ell}$. The solution to this optimization problem can be obtained by solving the following generalized eigenvalue problem

$$\mathbf{X}^T(\mathbf{D}' - \mathbf{W}')\mathbf{X}\mathbf{v} = \lambda\mathbf{X}^T(\mathbf{D} - \mathbf{W})\mathbf{X}\mathbf{v}, \tag{3}$$

for the eigenvectors $\mathbf{v}_i$ $(i = 1, \ldots, \ell)$ that correspond to the ℓ largest eigenvalues, where $\mathbf{D}$ and $\mathbf{D}'$ are diagonal matrices with diagonal elements $d_{ii} = \sum_j w_{ij}$ and $d'_{ii} = \sum_j w'_{ij}$. For notational simplicity, we denote $\mathbf{X}^T(\mathbf{D}' - \mathbf{W}')\mathbf{X}$ by $\mathbf{L}_b$, referred to as local between-class matrix, and $\mathbf{X}^T(\mathbf{D} - \mathbf{W})\mathbf{X}$ by $\mathbf{L}_w$, referred to as local within-class matrix. Then the generalized eigenvalue problem (3) can be rewritten as:

$$\mathbf{L}_b\mathbf{v} = \lambda\mathbf{L}_w\mathbf{v}. \tag{4}$$

Note that LDE has the same limitation as the classical LDA in that it suffers from the small sample size problem (or undersampling problem). Specifically, in many real-world applications such as face recognition where the dimensionality of the data is much larger than the sample size, the matrix $\mathbf{L}_w$ is singular and hence (4) cannot be solved directly. Moreover, its 1-D data representation also ignores the possibly useful higher-order structures in the data.

3. OUR LDET ALGORITHM

3.1. LDET Algorithm

Most previous approaches to subspace learning consider an object as a vector which is a 1-D representation. The corresponding learning algorithms are typically performed in very high-dimensional feature spaces. As a result, these methods usually suffer from the curse of dimensionality problem. Moreover, many objects found in image-based and video-based applications, such as face recognition, are more naturally represented as second- or higher-order tensors.

Formally, to perform subspace learning using a tensor representation, we are given a data set of n pth-order tensors $\{\mathbf{X}_i \mid \mathbf{X}_i \in \mathbb{R}^{d_1\times d_2\times\ldots\times d_p}\}_{i=1}^n$. The entire data set may also be represented as a $(p+1)$th-order sample tensor $\tilde{\mathbf{X}} \in \mathbb{R}^{d_1\times d_2\times\ldots\times d_p\times n}$. Embedding the data set to a lower-dimensional subspace corresponds to finding for each input tensor $\mathbf{X}_i$ a tensor $\mathbf{Z}_i \in \mathbb{R}^{\ell_1\times\ell_2\times\ldots\times\ell_p}$, with $\ell_j \ll d_j$ for $j = 1, \ldots, p$.

Let us first review some basic terminology on tensor operations [14]. The inner product of two tensors $\mathbf{A}$ and $\mathbf{B}$ with the same dimensions $d_1 \times d_2 \times \ldots \times d_p$ is $\langle\mathbf{A}, \mathbf{B}\rangle = \sum_{i_1,\ldots,i_p} \mathbf{A}_{i_1\ldots i_p}\mathbf{B}_{i_1\ldots i_p}$, the norm of a tensor $\mathbf{A}$ is $\|\mathbf{A}\| = \sqrt{\langle\mathbf{A}, \mathbf{A}\rangle}$, and the distance between two tensors $\mathbf{A}$ and $\mathbf{B}$ is $\|\mathbf{A} - \mathbf{B}\|$. In the case of second-order tensors, the tensor norm is just the matrix norm, called Frobenius norm, written as $\|\mathbf{A}\|_F$. The k-mode product of a tensor $\mathbf{A}$ with dimensions $d_1 \times d_2 \times \ldots \times d_p$ and a matrix $\mathbf{V} \in \mathbb{R}^{d_k\times\ell_k}$ is defined as $\mathbf{B} = \mathbf{A}\times_k\mathbf{V}$, where $\mathbf{B}_{i_1\ldots i_{k-1}ji_{k+1}\ldots i_p} = \sum_{i=1}^{d_k} V_{ij}\mathbf{A}_{i_1\ldots i_{k-1}ii_{k+1}\ldots i_p}$, for $j = 1, \ldots, \ell_k$. Besides the k-mode tensor-matrix product, there is another important tensor operation, called k-mode unfolding of a tensor into a matrix, which is defined as:

$$\begin{aligned} &\mathbf{A} \in \mathbb{R}^{d_1\times d_2\times\ldots\times d_p} \Rightarrow_k \mathbf{A}^k \in \mathbb{R}^{d_k\times\prod_{i\neq k} d_i}, \\ &\text{with } \mathbf{A}^k_{i_kj} = \mathbf{A}_{i_1\ldots i_p}, \\ &\qquad j = 1 + \textstyle\sum_{l=1,l\neq k}^n (i_l - 1)\prod_{o=l+1,o\neq k}^n d_o\,. \end{aligned} \tag{5}$$

With simple algebraic computation, we can obtain $\|\mathbf{A}\times_k\mathbf{V}\| = \|(\mathbf{A}^k)^T\mathbf{V}\|_F$.

To generalize LDE to work for tensors rather than vectors, we first change the matrix-vector multiplication $\mathbf{z}_i = \mathbf{V}^T\mathbf{x}_i$ to $\mathbf{Z}_i = \mathbf{X}_i \times_1 \mathbf{V}_1 \times_2 \ldots \times_p \mathbf{V}_p$, where $\mathbf{V}_j \in \mathbb{R}^{d_j\times\ell_j}$ for $j = 1, \ldots, p$. We thus rewrite (2) in the following form:

$$\begin{aligned} &\text{Maximize } Q(\mathbf{V}_1, \ldots, \mathbf{V}_p) = \\ &\qquad \sum_{i,j} \|\mathbf{X}_i \times_1 \ldots \times_p \mathbf{V}_p - \mathbf{X}_j \times_1 \ldots \times_p \mathbf{V}_p\|^2 w'_{ij}, \\ &\text{subject to} \\ &\qquad \sum_{i,j} \|\mathbf{X}_i \times_1 \ldots \times_p \mathbf{V}_p - \mathbf{X}_j \times_1 \ldots \times_p \mathbf{V}_p\|^2 w_{ij} = 1. \end{aligned} \tag{6}$$

This means that we find multiple interrelated projection matrices $\mathbf{V}_1, \ldots, \mathbf{V}_p$ that keep the neighboring points close to each other if they belong to the same class and prevent data points from other classes to enter the neighborhood of a data point.

Unfortunately this optimization problem is a high-order nonlinear programming problem with a nonlinear constraint, which has no closed-form solution. An alternative approach is to use an iterative method [13] to find the subspaces. The basic idea is as follows. To solve for $\mathbf{V}_k$, we assume that $\mathbf{V}_1, \ldots, \mathbf{V}_{k-1}, \mathbf{V}_{k+1}, \ldots, \mathbf{V}_p$ are known so that (6) can be rewritten as:

$$\begin{aligned} &\text{Maximize } Q(\mathbf{V}_k) = \\ &\qquad \sum_{i,j} \|\mathbf{X}_i \times_1 \ldots \times_p \mathbf{V}_p - \mathbf{X}_j \times_1 \ldots \times_p \mathbf{V}_p\|^2 w'_{ij}, \\ &\text{subject to} \\ &\qquad \sum_{i,j} \|\mathbf{X}_i \times_1 \ldots \times_p \mathbf{V}_p - \mathbf{X}_j \times_1 \ldots \times_p \mathbf{V}_p\|^2 w_{ij} = 1. \end{aligned} \tag{7}$$

For known $\mathbf{X}_i$ and $\mathbf{V}_1, \ldots, \mathbf{V}_{k-1}, \mathbf{V}_{k+1}, \ldots, \mathbf{V}_p$, we denote $\mathbf{X}_i \times_1 \mathbf{V}_1 \ldots \times_{k-1} \mathbf{V}_{k-1} \times_{k+1} \mathbf{V}_{k+1} \ldots \times_p \mathbf{V}_p$ by $\mathbf{Y}_i$. Then the optimization problem (7) can be rewritten as:

$$\begin{aligned} &\text{Maximize } Q(\mathbf{V}_k) = \sum_{i,j} \|\mathbf{Y}_i \times_k \mathbf{V}_k - \mathbf{Y}_j \times_k \mathbf{V}_k\|^2 w'_{ij}, \\ &\text{subject to } \sum_{i,j} \|\mathbf{Y}_i \times_k \mathbf{V}_k - \mathbf{Y}_j \times_k \mathbf{V}_k\|^2 w_{ij} = 1. \end{aligned} \tag{8}$$

Algorithm 1 LDET

Given the training data $\tilde{\mathbf{X}} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_p \times n}$, their class labels $\{y_i \mid y_i \in \{1, \ldots, h\}\}_{i=1}^{n}$, and the final lower dimensions $\ell_1 \times \ell_2 \times \ldots \times \ell_p$.

1. Construct neighborhood graphs G and G';
2. Compute affinity matrices $\mathbf{W}$ and $\mathbf{W}'$ for G and G';
3. Compute the embedding as follows:

Initialize $\mathbf{V}_1^0 = \mathbf{I}_{d_1}, \mathbf{V}_2^0 = \mathbf{I}_{d_2}, \ldots, \mathbf{V}_p^0 = \mathbf{I}_{d_p}$;

for $t = 1, 2, \ldots, T_{\text{max}}$ **do**

  for $k = 1, 2, \ldots, p$ **do**

    $\mathbf{Y}_i = \mathbf{X}_i \times_1 \mathbf{V}_1^t \ldots \times_{k-1} \mathbf{V}_{k-1}^t \times_{k+1} \mathbf{V}_{k+1}^t \ldots \times_p \mathbf{V}_p^t$;

    $\mathbf{Y}_i \Rightarrow_k \mathbf{Y}_i^k$;

    $\mathbf{L}_w = \sum_{i,j} w_{ij} (\mathbf{Y}_i^k - \mathbf{Y}_j^k)(\mathbf{Y}_i^k - \mathbf{Y}_j^k)^T$;

    $\mathbf{L}_b = \sum_{i,j} w'_{ij} (\mathbf{Y}_i^k - \mathbf{Y}_j^k)(\mathbf{Y}_i^k - \mathbf{Y}_j^k)^T$;

    $\mathbf{L}_b \mathbf{V}_k^t = \mathbf{L}_w \mathbf{V}_k^t \Lambda_k, \mathbf{V}_k^t \in \mathbb{R}^{d_k \times \ell_k}$;

    if $t > 2$ and $\|\mathbf{V}_k^t - \mathbf{V}_k^{t-1}\| < d_k \ell_k \varepsilon$ for each k **then**

      break;

    end if

  end for

end for

Output projection matrices $\mathbf{V}_k = \mathbf{V}_k^t \in \mathbb{R}^{d_k \times \ell_k}$, $k = 1, \ldots, p$.

Since $\|\mathbf{A} \times_k \mathbf{V}\| = \|(\mathbf{A}^k)^T \mathbf{V}\|_F$, we can express $Q(\mathbf{V}_k)$ in terms of the trace:

$$\begin{aligned}
Q(\mathbf{V}_k) &= \sum_{i,j} \|\mathbf{Y}_i \times_k \mathbf{V}_k - \mathbf{Y}_j \times_k \mathbf{V}_k\|^2 w'_{ij} \\
&= \sum_{i,j} \|(\mathbf{Y}_i^k)^T \mathbf{V}_k - (\mathbf{Y}_j^k)^T \mathbf{V}_k\|_F^2 w'_{ij} \\
&= \sum_{i,j} \text{Tr}\{(\mathbf{Y}_i^{kT} \mathbf{V}_k - \mathbf{Y}_j^{kT} \mathbf{V}_k)^T (\mathbf{Y}_i^{kT} \mathbf{V}_k - \mathbf{Y}_j^{kT} \mathbf{V}_k)\} w'_{ij} \\
&= \sum_{i,j} \text{Tr}\{\mathbf{V}_k^T (\mathbf{Y}_i^k - \mathbf{Y}_j^k)(\mathbf{Y}_i^k - \mathbf{Y}_j^k)^T \mathbf{V}_k\} w'_{ij} \\
&= \text{Tr}\{\mathbf{V}_k^T \sum_{i,j} w'_{ij} (\mathbf{Y}_i^k - \mathbf{Y}_j^k)(\mathbf{Y}_i^k - \mathbf{Y}_j^k)^T \mathbf{V}_k\}.
\end{aligned}$$

Hence the optimization problem (8) can further be rewritten as:

$$\begin{aligned}
&\text{Maximize } Q(\mathbf{V}_k) = \\
&\qquad \text{Tr}\{\mathbf{V}_k^T \sum_{i,j} w'_{ij} (\mathbf{Y}_i^k - \mathbf{Y}_j^k)(\mathbf{Y}_i^k - \mathbf{Y}_j^k)^T \mathbf{V}_k\}, \\
&\text{subject to} \qquad\qquad\qquad\qquad\qquad\qquad\qquad (9) \\
&\qquad \text{Tr}\{\mathbf{V}_k^T \sum_{i,j} w_{ij} (\mathbf{Y}_i^k - \mathbf{Y}_j^k)(\mathbf{Y}_i^k - \mathbf{Y}_j^k)^T \mathbf{V}_k\} = 1.
\end{aligned}$$

Similar to LDE, we define the local between-class matrix as $\mathbf{L}_b = \sum_{i,j} w'_{ij} (\mathbf{Y}_i^k - \mathbf{Y}_j^k)(\mathbf{Y}_i^k - \mathbf{Y}_j^k)^T$ and the local within-class matrix as $\mathbf{L}_w = \sum_{i,j} w_{ij} (\mathbf{Y}_i^k - \mathbf{Y}_j^k)(\mathbf{Y}_i^k - \mathbf{Y}_j^k)^T$. The columns of projection matrix $\mathbf{V}_k$ can be found by solving the generalized eigenvalue problem

$$\mathbf{L}_b \mathbf{v} = \lambda \mathbf{L}_w \mathbf{v}, \tag{10}$$

for the eigenvectors corresponding to the ℓ_k largest eigenvalues.

Algorithm 1 summarizes the complete LDET algorithm. We can notice that LDET degenerates to LDE if a first-order tensor representation (equivalent to a vector representation) is used, and hence LDE is a special case of LDET. We will show through experiments in the next section that LDET with higher-order tensors outperforms LDE.

3.2. Algorithmic Analysis

We now shift our focus to evaluate the merits of the proposed approach in terms of its learnability and computational complexity.

Singularity and Curse of Dimensionality: In LDE, the size of $\mathbf{L}_b$ and $\mathbf{L}_w$ is $\prod_{i=1}^{p} d_i \times \prod_{i=1}^{p} d_i$. For many applications, $\prod_{i=1}^{p} d_i$ is very large and hence the singularity problem is often encountered. In LDET, however, the stepwise matrices $\mathbf{L}_b$ and $\mathbf{L}_w$ are of size $d_k \times d_k$, which is much smaller than that of LDE. Moreover, the objects analyzed in LDET are the column vectors of the unfolded matrices and hence the sample size is essentially enlarged to $n \prod_{i \neq k}^{p} d_i$. Since d_k is much smaller than $n \prod_{i \neq k}^{p} d_i$ especially when higher-order tensors are used, it is less likely to suffer from the singularity problem.

Computational Cost: For simplicity of analysis, let us assume that the tensors have the same dimensionality for all tensor dimensions, i.e., $d_i = d$ for all $i = 1, \ldots, p$. Therefore, the time complexity of LDE is $O(d^{3p})$, while in LDET, the complexity of computing the matrices is $O(pd^{p+1})$ and the complexity of solving the generalized eigenvalue problem is $O(pd^3)$ for each iteration. Apparently the computational complexity of LDET is much lower. Although LDET has no closed-form solution and many iterations are required to solve the optimization problem, in practice it is still much faster than LDE due mainly to the simplicity of the computation required in each iteration.

4. EXPERIMENTAL RESULTS

In this section, we empirically evaluate the performance of LDET on face recognition. Since LDE has been shown to outperform most other subspace learning methods for face recognition [9], we only compare LDET with LDE here due to space limitation. LDET/2 and LDET/3 refer to LDET using second- and third-order tensors, respectively.

Dataset: Our experiments are performed on FERET [16] which is the most representative benchmark face dataset. In our experiments, 47 persons are randomly selected from the FERET dataset with 10 different gray-scale images per person. Five images for each person are randomly chosen for training and the remaining five for testing. The images are downsampled to a resolution of 56×46. Following [17], we extract 40 Gabor features with five different scales and eight different directions in the downsampled positions. Thus each image is encoded as a third-order tensor of size $56 \times 46 \times 40$ for LDET/3. To overcome the singularity problem in LDE, PCA is first applied to reduce the data dimensionality to $n - 1$.

Effect of the number of iterations T_{max}: We first study the effect of the number of iterations (T_{max} in Algorithm 1) on LDET/2. We also vary the numbers of nearest neighbors (k and k') for the neighborhood graph construction. For simplicity, we set $k = k'$ to be equal to 10 or 100. Figure 1 shows the classification rates of a nearest neighbor classifier under different settings, with $\ell_1 = \ell_2 = 6$. It is clear that both curves do not change much with the number of iterations. The implication is that we can simply set $T_{\text{max}} = 1$ in Algorithm 1 without having to repeat the outer loop.

Effect of the reduced dimensionality ℓ: We next study the effect of the reduced dimensionality ℓ on both the LDE and LDET algorithms, where the value of ℓ is $\prod_{k=1}^{p} \ell_k$ for LDET. We have performed extensive experiments using different values of ℓ. For simplicity, we choose the same dimensionality for all subspaces, i.e., $\ell_i = \sqrt[p]{\ell}$ for $i = 1, \ldots, p$. Figure 2 summarizes the results. The recognition rates of LDET/2 and LDET/3 become stabilized and reach the maximum

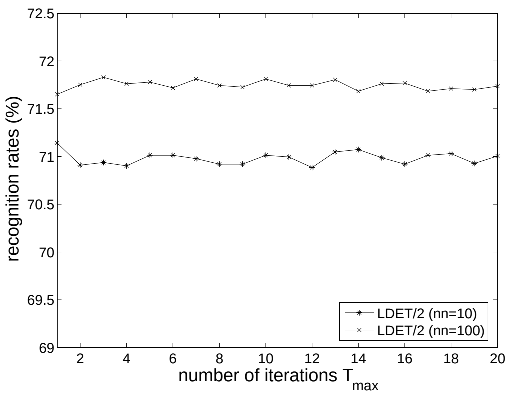

Fig. 1. Effect of the number of iterations on LDET/2 under two different numbers of nearest neighbors.

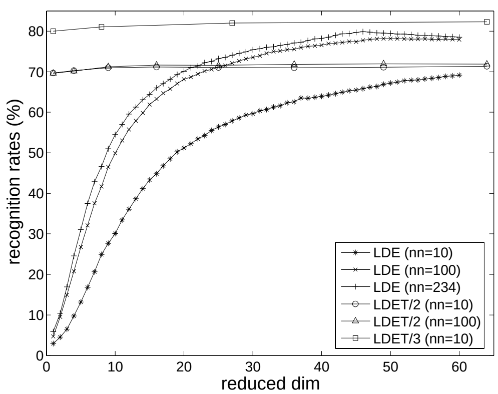

Fig. 2. Performance comparison of LDE and LDET under different settings.

values when the dimensionality reaches 8 to 27, while LDE reaches the maximum at around 50 to 60.

From all the experiments, we can say that LDET is a more effective subspace learning algorithm than LDE, especially for LDET/3 which achieves the highest recognition accuracy.

5. CONCLUSION

In this paper, we have proposed a tensor extension to the recently proposed LDE algorithm for subspace learning. By addressing the generalization and data representation problems simultaneously, the proposed LDET algorithm outperforms LDE in all our face recognition experiments. A natural extension of the empirical study is to consider even higher-order tensors, such as applying LDET/4 to video-based face recognition.

6. ACKNOWLEDGMENTS

This research has been supported by Competitive Earmarked Research Grant HKUST621305 from the Research Grants Council of the Hong Kong Special Administrative Region, China.

7. REFERENCES

[1] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[2] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.

[3] P.C. Yuen and J.H. Lai, "Face representation using independent component analysis," *Pattern Recognition*, vol. 35, no. 6, pp. 1247–1257, 2002.

[4] B. Schölkopf, A. Smola, and K.R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1999.

[5] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, pp. 2385–2404, 2000.

[6] C.J. Liu and H. Wechsler, "Robust coding schemes for indexing and retrieval from large face databases," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 132–137, January 2000.

[7] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data — with application to face recognition," *Pattern Recognition*, vol. 34, no. 10, pp. 2067–2070, 2001.

[8] X.F. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[9] H.T. Chen, H.W. Chang, and T.L. Liu, "Local discriminant embedding and its variants," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 846–853.

[10] J.B. Tenenbaum, V.D. Silva, and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[11] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[12] J. Yang, D. Zhang, A.F. Frangi, and J.Y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, 2004.

[13] J.P. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," in *Advances in Neural Information Processing Systems 17*, pp. 1569–1576. MIT Press, Cambridge, MA, 2005.

[14] D. Xu, S.C. Yan, L. Zhang, H.J. Zhang, Z.K. Liu, and H.Y. Shum, "Concurrent subspaces analysis," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 203–208.

[15] X.F. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 2006.

[16] P.J. Phillips, H. Wechsler, J. Huang, and P.J. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," in *Image and Vision Computing*, 1998, vol. 16, pp. 295–306.

[17] S.C. Yan, D. Xu, Q. Yang, L. Zhang, X.O. Tang, and H.J. Zhang, "Discriminant analysis with tensor representation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 526–532.