Expert Systems with Applications 39 (2012) 10500–10511

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Tensor distance based multilinear globality preserving embedding: A unified tensor based dimensionality reduction framework for image and video classification

Yang Liu[a,b,*], Yan Liu[b], Shenghua Zhong[b], Keith C.C. Chan[b]

[a] Department of Statistics, Yale University, New Haven, CT 06511, USA
[b] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, PR China

ARTICLE INFO

Keywords:
Multimedia mining
Pattern recognition
Image and video classification
Tensor based learning
Tensor distance metric
Multilinear globality preserving embedding strategy
Tensor distance based multilinear multidimensional scaling
Tensor distance based multilinear isometric embedding

ABSTRACT

Image and video classification tasks often suffer from the problem of high-dimensional feature space. How to discover the meaningful, low-dimensional representations of such high-order, high-dimensional observations remains a fundamental challenge. In this paper, we present a unified framework for tensor based dimensionality reduction including a new tensor distance (TD) metric and a novel multilinear globality preserving embedding (MGPE) strategy. Different with the traditional Euclidean distance, which is constrained by orthogonality assumption, TD measures the distance between data points by considering the relationships among different coordinates of high-order data. To preserve the natural tensor structure in low-dimensional space, MGPE directly works on the high-order form of input data and employs an iterative strategy to learn the transformation matrices. To provide faithful global representation for datasets, MGPE intends to preserve the distances between all pairs of data points. According to the proposed TD metric and MGPE strategy, we further derive two algorithms dubbed tensor distance based multilinear multidimensional scaling (TD-MMDS) and tensor distance based multilinear isometric embedding (TD-MIE). TD-MMDS finds the transformation matrices by keeping the TDs between all pairs of input data in the embedded space, while TD-MIE intends to preserve all pairwise distances calculated according to TDs along shortest paths in the neighborhood graph. By integrating tensor distance into tensor based embedding, TD-MMDS and TD-MIE perform tensor based dimensionality reduction through the whole learning procedure and achieve obvious performance improvement on various standard datasets.

1. Introduction

Visual pattern analysis tasks, such as image and video classification, often suffer from the problem of high-dimensional feature space, which will lead to low recognition accuracy and expensive computational cost (Duda, Hart, & Stork, 2001; Foley, 1972). Dimensionality reduction techniques provide a means to solve this problem by generating a low-dimensional equivalence of the original feature space for the given targets (Carreira-Perpinan, 1996; Tsai, 2012; van der Maaten, Postma, & van den Herik, 2009).

However, traditional linear and nonlinear dimensionality reduction methods generally unfold the input data to vectors before embedding, even though the data are naturally high-order tensors. For example, a middle size grayscale image (as shown in Fig. 1) with the resolution of 352 × 288, i.e., an intrinsically second-order tensor, is unfolded to a 101,376-dimensional vector as the input of embedding. Such kind of vectorization causes several problems. First, the dimension of vector form representation is generally very high, which makes it more difficult to unveil the low-dimensional structure. In addition, the huge dimensionality of the vector leads to very high computational cost (Lu, Plataniotis, & Venetsanopoulos, 2008; Tao, Li, Wu, & Maybank, 2007; Yan & Xu et al., 2007). More importantly, the intrinsic tensor structure of the data, i.e., the relationships among the orders, the relationships between the dimension and the order, the relationships of the dimensions within the same order, and the relationships of the dimensions in the different orders, will be seriously destroyed by the vectorization (Lu et al., 2008; Tao et al., 2007; Yan & Xu et al., 2007). For example, if there are two neighboring pixels in an image (a second-order tensor), such as the pixel in the coordinate location (i,j) and the pixel in the coordinate location $(i+1,j+1)$, they may be largely separated from each other after vectorization.

To address these problems, Vasilescu and Terzopoulos (2003) introduced multilinear algebra into dimensionality reduction for face image analysis. After that, many multilinear dimensionality reduction techniques, also referred as tensor based dimensionality reduction techniques, have been proposed and applied to different

[*] Corresponding author at: Department of Statistics, Yale University, New Haven, CT 06511, USA.

E-mail addresses: yang.liu@yale.edu, csygliu@comp.polyu.edu.hk (Y. Liu), csyliu@comp.polyu.edu.hk (Y. Liu), csshzhong@comp.polyu.edu.hk (S. Zhong), cskcchan@comp.polyu.edu.hk (K.C.C. Chan).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.02.139

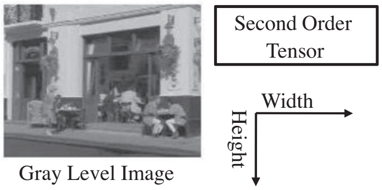


Fig. 1. A gray level image is a second-order tensor. The first order is the height order and the second order is the width order.

tasks successfully. Yang, Zhang, Frangi, and Yang (2004) extended the principal component analysis (PCA) (Hotelling, 1933) to its second-order version, which is called 2-D PCA. Ye, Janardan, and Li (2004) proposed the 2-D linear discriminant analysis (2-D LDA), the extension of classical LDA (Fisher, 1936), to seek an optimal tensor subspace that maximizes the discriminant information. Wang, Wang, and Feng (2006) showed that 2-D PCA and 2-D LDA are special cases of image block based feature extraction. Ding and Ye (2005) proposed 2-D singular value decomposition (2-D SVD) to compute the principal eigenvectors of row-row and column-column covariance matrices respectively. He, Cai, and Niyogi (2006) proposed a second-order tensor subspace analysis (TSA) algorithm according to the vector based algorithm locality preserving projections (LPP) (He & Niyogi, 2004). More tensor based dimensionality reduction techniques directly work on the input data with arbitrary order, such as multilinear PCA (MPCA) (Lu et al., 2008; Panagakis, Kotropoulos, & Arce, 2010; Xu et al., 2008), multilinear LDA (MLDA) (Tao et al., 2007; Yan & Xu et al., 2007), tensor based locality preserving projections (TLPP) (Dai & Yeung, 2006), tensor based neighborhood preserving embedding (TNPE) (Dai & Yeung, 2006), tensor based graph embedding framework (Yan et al., 2007), and so on (Ergin, Cakir, Nezih Gerek, & Gülmezoglu, 2011; Fu & Huang, 2008; Gao & Tian, 2009; Geng, Smith-Miles, Zhou, & Wang, 2011; Liu, Liu, & Chan, 2010; Luo, Ding, & Huang, 2011).

From the objective functions of above tensor based algorithms, it is obvious that most of them intend to preserve the distances between data points when mapping high-order data to low-dimensional space. Therefore, the performance of tensor based dimensionality reduction is closely related to the distance metric. Currently, most of the tensor based techniques simply use the Euclidean distance as their distance metric. In traditional Euclidean space, a high-order data point $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is represented by the coordinates $x_1, x_2, \ldots, x_{I_1 \times I_2 \times \ldots \times I_N}$ under the corresponding bases $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{I_1 \times I_2 \times \ldots \times I_N}$, where $\mathbf{e}_i^T \mathbf{e}_j = 1$ if $i = j$; $\mathbf{e}_i^T \mathbf{e}_j = 0$ otherwise. It means that any two different bases $\mathbf{e}_i$ and $\mathbf{e}_j$ in the Euclidean space are assumed to be mutually perpendicular, so that the coordinates x_i and x_j are independent of each other. Unfortunately, this orthogonality assumption ignores the relationships among different coordinates for high-order data, such as the spatial relationships of pixels in images, and thus limits the performance of further tensor based embedding.

To release the orthogonality assumption in the Euclidean distance metric, we propose a tensor distance (TD) to measure the relationships between high-order data points. Based on this new distance metric, we further present a unified dimensionality reduction framework called tensor distance based multilinear globality preserving embedding (TD-MGPE). The proposed MGPE strategy attempts to preserve both the natural tensor structure of high-order data and the faithful global representation of the whole dataset by sequentially learning a series of transformation matrices. According to the proposed TD metric and MGPE strategy, we further derive two new algorithms: tensor distance based multilinear multidimensional scaling (TD-MMDS) and tensor distance based multilinear isometric embedding (TD-MIE). Concretely, the proposed framework has several attractive characters:

- TD-MGPE directly works on more general tensor representations of the data with arbitrary order, which include the vectorized representation as a special case. By keeping the natural tensor structure, the intrinsic relationships among the dimensions and the orders for high-order data are better preserved. Besides, MGPE can reduce the computational complexity caused by vectorization.
- TD-MGPE utilizes the distance metric TD, which can reflect the relationships among different coordinates in high-order data, such as the spatial relationships of pixels in images and the spatial–temporal relationships of pixels in videos. Moreover, as a general distance metric, TD can be easily incorporated into other tensor based dimensionality reduction algorithms, such as MPCA and MLDA.
- By integrating tensor distance into tensor based embedding, TD-MGPE provides a unified framework for tensor based dimensionality reduction through the whole learning procedure, and preserves the global structure in the low-dimensional equivalence well. In addition, TD-MGPE can map the new data points straightforwardly with the explicit embedding function, and thus is suitable for dimensionality reduction in the classification problems.

The rest of this paper is organized as follows. Section 2 briefly reviews two representative global dimensionality reduction algorithms, i.e., multidimensional scaling (MDS) (Kruskal & Wish, 1977) and isometric feature mapping (Isomap) (Tenenbaum, de Silva, & Langford, 2000). Section 3 proposes the TD metric as well as the MGPE strategy, and derives two new algorithms TD-MMDS and TD-MIE under the TD-MGPE framework. Experimental results are reported in Section 4. The paper is concluded in Section 5.

2. Related work

2.1. Multidimensional scaling

Multidimensional scaling (MDS) (Kruskal & Wish, 1977) is one of the classical global embedding algorithms. It aims to map the high-dimensional data to the low-dimensional representations while retaining all pairwise Euclidean distances between the data points as much as possible. Concretely, given n data points: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, MDS maps $\mathbf{x}_i$ to its corresponding d-dimensional representation $\mathbf{y}_i$ $(d \ll D)$ by minimizing the following cost function:

$$\sum_{ij} (d_E(\mathbf{x}_i, \mathbf{x}_j) - d_E(\mathbf{y}_i, \mathbf{y}_j))^2 \tag{1}$$

where $d_E(\mathbf{x}_i,\mathbf{x}_j)$ is the Euclidean distance between original data points $\mathbf{x}_i$ and $\mathbf{x}_j$; $d_E(\mathbf{y}_i,\mathbf{y}_j)$ is the Euclidean distance between corresponding embedded data points $\mathbf{y}_i$ and $\mathbf{y}_j$.

MDS first constructs an n-dimensional distance matrix $\mathbf{D}_E$ by using the Euclidean distances between original data points $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e., $\mathbf{D}_E = (d_E(\mathbf{x}_i,\mathbf{x}_j))$. Then it computes the inner product matrix $\tau(\mathbf{D}_E)$:

$$\tau(\mathbf{D}_E) = -\mathbf{HSH}/2 \tag{2}$$

where $\mathbf{S}_{ij} = d_E^2(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{H} = \mathbf{I} - (\mathbf{ee}^T/n)$, $\mathbf{I}$ is the identity matrix and $\mathbf{e}$ is the vector of all ones (Cox, 2005). Finally, let λ_p be the pth largest eigenvalue of matrix $\tau(\mathbf{D}_E)$, and $\mathbf{v}_p^i$ be the ith component of the pth eigenvector. Then the d-dimensional embedded coordinate vector $\mathbf{y} = \left(\sqrt{\lambda_1}\mathbf{v}_1^i, \sqrt{\lambda_2}\mathbf{v}_2^i, \ldots, \sqrt{\lambda_d}\mathbf{v}_d^i\right)^T$.

2.2. Isometric feature mapping

Isometric feature mapping (Isomap) (Tenenbaum et al., 2000), another representative global embedding technique, attempts to preserve all pairwise geodesic distances, which are estimated by computing the shortest path distances on the neighborhood graph. The objective of Isomap is to minimize the following cost function:

$$\sum_{ij}(d_G(\mathbf{x}_i,\mathbf{x}_j) - d_E(\mathbf{y}_i,\mathbf{y}_j))^2 \tag{3}$$

where $d_G(\mathbf{x}_i,\mathbf{x}_j)$ is the shortest path distance between original data points $\mathbf{x}_i$ and $\mathbf{x}_j$.

Isomap first constructs a neighbourhood graph G over all data points by connecting points $\mathbf{x}_i$ and $\mathbf{x}_j$ if $d_E(\mathbf{x}_i,\mathbf{x}_j)$ is smaller than a threshold ε (ε-Isomap), or if $\mathbf{x}_i$ (or $\mathbf{x}_j$) is one of the k-nearest neighbors of $\mathbf{x}_j$ (or $\mathbf{x}_i$) (k-Isomap). Then the edge lengths have been set equal to $d_E(\mathbf{x}_i,\mathbf{x}_j)$. The second step of Isomap initializes $d_0(\mathbf{x}_i,\mathbf{x}_j) = -d_E(\mathbf{x}_i,\mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ is connected by an edge; $d_0(\mathbf{x}_i,\mathbf{x}_j) = +\infty$ otherwise. Then Floyd's algorithm or Dijkstra's algorithm (Cormen, Leiserson, Rivest, & Stein, 2001) is employed to compute the final shortest paths $d_G(\mathbf{x}_i,\mathbf{x}_j)$ between all pairs of data points. The distance matrix of graph G is defined by $\mathbf{D}_G = (d_G(\mathbf{x}_i,\mathbf{x}_j))$. Similarly, we have the inner product matrix $\tau(\mathbf{D}_G)$:

$$\tau(\mathbf{D}_G) = -\mathbf{HSH}/2 \tag{4}$$

where $\mathbf{S}_{ij} = d_G^2(\mathbf{x}_i,\mathbf{x}_j)$, and $\mathbf{H}$ is defined similarly as in Eq. (2). Finally, similar as in MDS, let λ_p be the pth largest eigenvalue of matrix $\tau(\mathbf{D}_G)$, and $\mathbf{v}_p^i$ be the ith component of the pth eigenvector. Then the d-dimensional embedded coordinate vector $\mathbf{y} = (\sqrt{\lambda_1}\mathbf{v}_1^i, \sqrt{\lambda_2}\mathbf{v}_2^i, \ldots, \sqrt{\lambda_d}\mathbf{v}_d^i)^T$.

3. Tensor distance based multilinear globality preserving embedding

In this section, we present our framework: the tensor distance based multilinear globality preserving embedding (TD-MGPE). We first introduce a tensor distance metric. Then we propose a multilinear globality preserving embedding strategy. Under the proposed TD-MGPE framework, we further derive two new algorithms: tensor distance based multilinear multidimensional scaling (TD-MMDS) and tensor distance based multilinear isometric embedding (TD-MIE).

3.1. Tensor distance metric

For some kinds of high-order data, such as images and videos, traditional Euclidean distance may not reflect the real distance between two data points because of the orthogonality assumption discussed in the introduction. Wang, Zhang, and Feng (2005) proposed an image based Euclidean distance (IMED), which considers the spatial relationships of pixels and thus is robust to small perturbation of images. Inspired by this work, we propose a distance metric called tensor distance (TD) to model the correlation among different coordinates of data with arbitrary number of orders.

Given a data point $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ ($N > 1$), we use $\mathbf{x}$ to denote the vector form representation of $\mathcal{X}$. Therefore, the element $\mathcal{X}_{i_1 i_2 \ldots i_N}$ ($1 \leqslant i_j \leqslant I_j$, $1 \leqslant j \leqslant N$) in $\mathcal{X}$ is corresponding to x_l, i.e., the lth element in $\mathbf{x}$, where $l = i_1 + \sum_{j=2}^{N}(i_j - 1)\prod_{o=1}^{j-1} I_o$ ($2 \leqslant j \leqslant N$). Then the TD between two tensors $\mathcal{X}$ and $\mathcal{Y}$ can be represented as:

$$d_{TD}(\mathcal{X},\mathcal{Y}) = \sqrt{\sum_{l,m=1}^{I_1 \times I_2 \times \ldots \times I_N} g_{lm}(x_l - y_l)(x_m - y_m)} = \sqrt{(\mathbf{x}-\mathbf{y})^T \mathbf{G}(\mathbf{x}-\mathbf{y})} \tag{5}$$

where g_{lm} is the metric coefficient and $\mathbf{G}$ is the metric matrix. To reflect the intrinsic relationships between different coordinates for high-order data, a natural consideration is that the metric coefficients should be related to the element distances. Wang et al. (2005) have already demonstrated that, for image data, i.e., the second-order tensors, if the metric coefficients depend properly on distances of pixel locations, the obtained distance metric can effectively reflect the spatial relationships between pixels. Inspired by this study, we design a metric matrix $\mathbf{G}$:

$$g_{lm} = \frac{1}{2\pi\sigma^2} e^{-\frac{\|\mathbf{p}_l - \mathbf{p}_m\|_2^2}{2\sigma^2}} \tag{6}$$

where σ is a regularization parameter, $\|\mathbf{p}_l - \mathbf{p}_m\|$ is the location distance between elements $\mathcal{X}_{i_1 i_2 \ldots i_N}$ (corresponding to x_l) and $\mathcal{X}_{i'_1 i'_2 \ldots i'_N}$ (corresponding to x_m), which is defined as:

$$\|\mathbf{p}_l - \mathbf{p}_m\|_2 = \sqrt{(i_1 - i'_1)^2 + (i_2 - i'_2)^2 + \cdots + (i_N - i'_N)^2} \tag{7}$$

Then $d_{TD}(\mathcal{X},\mathcal{Y})$ can be rewritten as:

$$d_{TD}(\mathcal{X},\mathcal{Y}) = \sqrt{\frac{1}{2\pi\sigma^2} \sum_{l,m=1}^{I_1 \times I_2 \times \ldots \times I_N} e^{-\frac{\|\mathbf{p}_l - \mathbf{p}_m\|_2^2}{2\sigma^2}} (x_l - y_l)(x_m - y_m)} \tag{8}$$

Actually, Euclidean distance can be viewed as a special case of proposed TD. If we let the metric matrix to be identity matrix, i.e., $\mathbf{G} = \mathbf{I}$, which means that we only consider the distance between corresponding coordinates of two high-order data in tensor space, then TD is reduced to Euclidean distance. Since $\mathbf{G}$ is a real symmetric positive definite matrix, we can easily decompose it as follows:

$$\mathbf{G} = \mathbf{G}^{\frac{1}{2}}\mathbf{G}^{\frac{1}{2}} \tag{9}$$

where $\mathbf{G}^{\frac{1}{2}}$ is also a real symmetric matrix defined as:

$$\mathbf{G}^{\frac{1}{2}} = \mathbf{U}_G \Lambda_G^{\frac{1}{2}} \mathbf{U}_G^T \tag{10}$$

Here Λ_G is a diagonal matrix whose elements are eigenvalues of $\mathbf{G}$, and $\mathbf{U}_G$ is an orthogonal matrix whose column vectors are eigenvectors of $\mathbf{G}$. Applying the transformation $\mathbf{G}^{\frac{1}{2}}$ to the vector form representations $\mathbf{x}$ and $\mathbf{y}$, i.e., $\mathbf{x}' = \mathbf{G}^{\frac{1}{2}}\mathbf{x}$, $\mathbf{y}' = \mathbf{G}^{\frac{1}{2}}\mathbf{y}$, the proposed TD between $\mathbf{x}$ and $\mathbf{y}$ is then reduced to the traditional Euclidean distance between $\mathbf{x}'$ and $\mathbf{y}'$:

$$d_{TD}(\mathcal{X},\mathcal{Y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T \mathbf{G}^{\frac{1}{2}}\mathbf{G}^{\frac{1}{2}}(\mathbf{x}-\mathbf{y})} = \sqrt{(\mathbf{x}'-\mathbf{y}')^T(\mathbf{x}'-\mathbf{y}')} \tag{11}$$

So it is easy to embed TD to general learning procedures: we simply need to perform the transformation $\mathbf{G}^{\frac{1}{2}}$ on original data and then use transformed data in the following procedures.

3.2. Multilinear globality preserving embedding strategy

In this subsection, we propose the new multilinear globality preserving embedding (MGPE) strategy, which aims to preserve the structure of dataset at all scales by retaining all pairwise distances between the data points as much as possible. Unlike previous global methods such as MDS and Isomap, MGPE directly works on the high-order representations of input data and employs an iterative strategy to learn the transformation matrices sequentially. Before formally presenting the strategy, we introduce several necessary definitions in multilinear algebra. Further details of multilinear algebra are available in (Kolda, 2001; Lathauwer, 1997; Lathauwer, Moor, & Vandewalle, 2000; Tao, Li, Wu, Hu, & Maybank, 2007; Zhang & Chow, 2012).

Definition 1 (*tensor and its order, mode and mode size*). Assume $\mathcal{X}$ be a tensor of size $I_1 \times I_2 \times \cdots \times I_N$. The order of $\mathcal{X}$ is N and the jth mode of $\mathcal{X}$ is of size I_j ($1 \leqslant j \leqslant N$). In addition, an element of $\mathcal{X}$ is denoted by $\mathcal{X}_{i_1 i_2 \ldots i_N}$ ($1 \leqslant i_j \leqslant I_j$, $1 \leqslant j \leqslant N$).

Definition 2 (*tensor inner product, norm, and distance*). The inner product of two tensors $\mathcal{X},\mathcal{Y}\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ is defined as $\langle\mathcal{X},\mathcal{Y}\rangle=\sum_{i_1=1,i_2=1,\ldots,i_N=1}^{I_1,I_2,\ldots,I_N}\mathcal{X}_{i_1i_2\ldots i_N}\mathcal{Y}_{i_1i_2\ldots i_N}$. The Frobenius norm of a tensor $\mathcal{X}$ is given by $\|\mathcal{X}\|_F=\sqrt{\langle\mathcal{X},\mathcal{X}\rangle}$. The distance of two tensors, consequently, is defined as $d(\mathcal{X},\mathcal{Y})=\|\mathcal{X}-\mathcal{Y}\|_F$.

Definition 3 (*k-mode product*). The k-mode product of a tensor $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ by a matrix $\mathbf{U}\in\mathbb{R}^{J_k\times I_k}$, denoted by $\mathcal{X}\times_k\mathbf{U}$, is an $(I_1\times\ldots\times I_{k-1}\times J_k\times I_{k+1}\times\cdots\times I_N)$-dimensional tensor of which the entries are given by $(\mathcal{X}\times_k\mathbf{U})_{i_1\ldots i_{k-1}j_ki_{k+1}\ldots i_N}=\sum_{i_k=1}^{I_k}\mathcal{X}_{i_1\ldots i_{k-1}i_ki_{k+1}\ldots i_N}\mathbf{U}_{j_ki_k},j_k=1,\ldots,J_k$.

Property 1. *Given a tensor* $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ *and the matrices* $\mathbf{U}\in\mathbb{R}^{J_k\times I_k},\mathbf{V}\in\mathbb{R}^{J_l\times I_l}$ *(k ≠ l), then* $(\mathcal{X}\times_k\mathbf{U})\times_l\mathbf{V}=(\mathcal{X}\times_l\mathbf{V})\times_k\mathbf{U}=\mathcal{X}\times_k\mathbf{U}\times_l\mathbf{V}$.

Definition 4 (*k-mode unfolding*). The k-mode unfolding of a tensor $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ into a matrix $\mathbf{X}^k\in\mathbb{R}^{I_k\times\prod_{j\neq k}I_j}$, i.e., $\mathbf{X}^k\Leftarrow_k\mathcal{X}$, is defined as $\mathbf{X}^k_{i_k,j}=\mathcal{X}_{i_1i_2\ldots i_N}$, $j=\sum_{m=2}^{N-1}(i_{p(m)}-1)\prod_{o=m+1}^{N}I_{p(o)}+i_{p(N)}$, where $p(m)$ is the mth element of sequence $\{k,k+1,\ldots,N-1,\ N,\ 1,\ 2,\ \ldots,\ k-1\}$.

Property 2. *If* $\mathbf{X}^k\Leftarrow_k\mathcal{X}$, *then* $\|\mathcal{X}\times_k\mathbf{U}_k^T\|_F=\|\mathbf{U}_k^T\mathbf{X}^k\|_F$

Based on above definitions and properties, we present the MGPE strategy as follows. Given n data points $\mathcal{X}_1,\mathcal{X}_2,\ldots,\mathcal{X}_n\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$, MGPE aims to find N transformation matrices $\mathbf{V}_k\in\mathbb{R}^{I_k\times I'_k}$ $(I'_k\ll I_k,\ k=1,\ldots,N)$ such that n low dimensional data points $\mathcal{Y}_1,\mathcal{Y}_2,\ldots,\mathcal{Y}_n\in\mathbb{R}^{I'_1\times I'_2\times\ldots\times I'_N}$ can be obtained by the multilinear transformation $\mathcal{Y}_j=\mathcal{X}_j\times_1\mathbf{V}_1^T\times_2\ldots\times_N\mathbf{V}_N^T$ $(j=1,\ldots,n)$.

In order to preserve the global structure of input data in the low-dimensional tensor subspace, the objective function of MGPE is formulated as follows based on the graph embedding framework (Yan et al., 2007):

$$\underset{\mathbf{V}_k^T\mathbf{V}_k=\mathbf{I}_{I'_k}}{\operatorname{argmin}}J(\mathbf{V}_1,\ldots,\mathbf{V}_N)=\sum_{i,j=1}^{N}\|\mathcal{Y}_i-\mathcal{Y}_j\|_F^2\tau(\mathbf{D})_{ij}\tag{12}$$

where $\mathbf{D}$ is a general distance matrix which could be obtained in terms of any distance metric. Eq. (12) can be rewritten as:

$$\underset{\mathbf{V}_k^T\mathbf{V}_k=\mathbf{I}_{I'_k}}{\operatorname{argmin}}J(\mathbf{V}_1,\ldots,\mathbf{V}_N)=\sum_{i,j=1}^{N}\|(\mathcal{X}_i-\mathcal{X}_j)\times_1\mathbf{V}_1^T\times_2\ldots\times_N\mathbf{V}_N^T\|_F^2\tau(\mathbf{D})_{ij}\tag{13}$$

Since Eq. (13) is a high-order non-convex optimization problem, there is no closed-form solution for it. Instead, we employ an iterative strategy (Kolda, 2001; Lathauwer et al., 2000; Lu et al., 2008; Yan & Xu et al., 2007) to find a local optimal solution based on the following theorem:

Theorem 1. *Assume that* $\mathbf{V}_1$, $\mathbf{V}_2$, …, $\mathbf{V}_{k-1}$, $\mathbf{V}_{k+1}$, …, $\mathbf{V}_N$ *are fixed, the optimal transformation matrix* $\mathbf{V}_k$ *minimizing the objective function* $J(\mathbf{V}_1,\ldots,\mathbf{V}_N)$ *is composed of the first* I'_k *eigenvectors of the following matrix:*

$$\mathbf{H}^k=-\sum_{i,j=1}^{n}\left(\mathbf{X}_i^k-\mathbf{X}_j^k\right)\left(\mathbf{X}_i^k-\mathbf{X}_j^k\right)^T\tau(\mathbf{D})_{ij}\tag{14}$$

corresponding to the first I'_k *largest eigenvalues, where* $\mathbf{X}_i^k$ *is the k-mode unfolding of the tensor* $\mathcal{X}_i^k$, *i.e.,* $\mathbf{X}_i^k\Leftarrow_k\mathcal{X}_i^k$, *and* $\mathcal{X}_i^k=\mathcal{X}_i\times_1\mathbf{V}_1^T\times_2\ldots\times_{k-1}\mathbf{V}_{k-1}^T\times_{k+1}\mathbf{V}_{k+1}^T\times_{k+2}\ldots\times_N\mathbf{V}_N^T$.

Proof. We denote the objective function $J(\mathbf{V}_1,\ldots,\mathbf{V}_N)$ as $J(\mathbf{V}_k)$ when the matrices $\mathbf{V}_1$, $\mathbf{V}_2$, …, $\mathbf{V}_{k-1}$, $\mathbf{V}_{k+1}$, …, $\mathbf{V}_N$ are fixed. Based on the properties of multilinear algebra and matrix trace, the objective function $J(\mathbf{V}_k)$ can be rewritten as follows:

$$\begin{aligned}
\min J(\mathbf{V}_k)&=\min\sum_{i,j=1}^{N}\left\|\mathcal{Y}_i-\mathcal{Y}_j\right\|_F^2\tau(\mathbf{D})_{ij}\\
&=\min\sum_{i,j=1}^{N}\left\|(\mathcal{X}_i-\mathcal{X}_j)\times_1\mathbf{V}_1^T\times_2\ldots\times_N\mathbf{V}_N^T\right\|_F^2\tau(\mathbf{D})_{ij}\\
&=\min\sum_{i,j=1}^{N}\left\|(\mathcal{X}_i-\mathcal{X}_j)\times_1\mathbf{V}_1^T\ldots\times_{k-1}\mathbf{V}_{k-1}^T\times_{k+1}\mathbf{V}_{k+1}^T\ldots\times_N\mathbf{V}_N^T\times_k\mathbf{V}_k^T\right\|_F^2\tau(\mathbf{D})_{ij}\\
&=\min\sum_{i,j=1}^{N}\left\|\left(\mathcal{X}_i^k-\mathcal{X}_j^k\right)\times_k\mathbf{V}_k^T\right\|_F^2\tau(\mathbf{D})_{ij}=\min\sum_{i,j=1}^{N}\left\|\mathbf{V}_k^T(\mathbf{X}_i^k-\mathbf{X}_j^k)\right\|_F^2\tau(\mathbf{D})_{ij}\\
&=\min\sum_{i,j=1}^{N}\operatorname{tr}\left(\mathbf{V}_k^T\left(\mathbf{X}_i^k-\mathbf{X}_j^k\right)\left(\mathbf{X}_i^k-\mathbf{X}_j^k\right)^T\mathbf{V}_k\right)\tau(\mathbf{D})_{ij}\\
&=\min tr\left(\sum_{i,j=1}^{N}\left(\mathbf{V}_k^T\left(\mathbf{X}_i^k-\mathbf{X}_j^k\right)\left(\mathbf{X}_i^k-\mathbf{X}_j^k\right)^T\mathbf{V}_k\right)\tau(\mathbf{D})_{ij}\right)\\
&=\min tr\left(\mathbf{V}_k^T\left(\sum_{i,j=1}^{N}\left(\mathbf{X}_i^k-\mathbf{X}_j^k\right)\left(\mathbf{X}_i^k-\mathbf{X}_j^k\right)^T\tau(\mathbf{D})_{ij}\right)\mathbf{V}_k\right)\\
&=\max tr\left(\mathbf{V}_k^T\left(-\sum_{i,j=1}^{N}\left(\mathbf{X}_i^k-\mathbf{X}_j^k\right)\left(\mathbf{X}_i^k-\mathbf{X}_j^k\right)^T\tau(\mathbf{D})_{ij}\right)\mathbf{V}_k\right)\\
&=\max tr\left(\mathbf{V}_k^T\mathbf{H}^k\mathbf{V}_k\right).
\end{aligned}\tag{15}$$

Therefore, all columns of the optimal transformation matrix $\mathbf{V}_k$ are given by the first I'_k eigenvectors of matrix $\mathbf{H}^k$ corresponding to the I'_k largest eigenvalues. □

According to above theorem, if $\mathbf{V}_1$, $\mathbf{V}_2$, …, $\mathbf{V}_{k-1}$, $\mathbf{V}_{k+1}$, …, $\mathbf{V}_N$ are fixed, the optimal $\mathbf{V}_k$ can be acquired by simple eigendecomposition. Therefore, the iterative strategy can be presented as follows. First we fix $\mathbf{V}_2$, …, $\mathbf{V}_N$, and obtain $\mathbf{V}_1$ by minimizing the objective function $J(\mathbf{V}_1)$, i.e., maximizing $\operatorname{tr}(\mathbf{V}_1^T\mathbf{H}^1\mathbf{V}_1)$. Then we fix $\mathbf{V}_1$, $\mathbf{V}_3$, …, $\mathbf{V}_N$, and obtain $\mathbf{V}_2$ by minimizing the objective function $J(\mathbf{V}_2)$. The rest could be deduced by analogy. At last we fix $\mathbf{V}_1$, $\mathbf{V}_2$, …, $\mathbf{V}_{N-1}$, and obtain the optimal $\mathbf{V}_N$ by minimizing the objective function $J(\mathbf{V}_N)$. Repeat above steps until the termination conditions are satisfied, and then we obtain a local optimal solution.

Theorem 2. *The procedure of MGPE will converge to a local optimal solution.*

Proof. To prove the convergence of MGPE, we only need to prove that the objective function $J(\mathbf{V}_1,\ldots,\mathbf{V}_N)$ is nonincreasing and has a lower bound.

The derivation of Theorem 1 indicates that in each iteration of the MGPE procedure, the objective function $J(\mathbf{V}_1,\ldots,\mathbf{V}_N)$ is nonincreasing, i.e.,

$$\begin{aligned}
J\left(\mathbf{V}_1^t,\mathbf{V}_2^t,\ldots,\mathbf{V}_N^t\right)&\geqslant J\left(\mathbf{V}_1^{t+1},\mathbf{V}_2^t,\ldots,\mathbf{V}_N^t\right)\\
&\geqslant J\left(\mathbf{V}_1^{t+1},\mathbf{V}_2^{t+1},\ldots,\mathbf{V}_N^t\right)\geqslant\ldots\\
&\geqslant J(\mathbf{V}_1^{t+1},\mathbf{V}_2^{t+1},\ldots,\mathbf{V}_N^{t+1}),
\end{aligned}\tag{16}$$

This is because that each update of transformation matrix $\mathbf{V}_k$ minimizes current objective function $J(\mathbf{V}_k)$ while the other matrices $\mathbf{V}_i$ $(i=1,2,\ldots,k-1,k+1,\ldots,N)$ are fixed.

On the other hand, for any i, j, we have $\|\mathcal{Y}_i-\mathcal{Y}_j\|_F^2\geqslant 0$. Furthermore, because the row sum of matrix $\tau(\mathbf{D})$ is zero (Yan et al., 2007), the value of $\min[\tau(\mathbf{D})]$, i.e., the minimum element in matrix $\tau(\mathbf{D})$, is definitely nonpositive. Consequently, we have the following:

$$\begin{aligned}
J\left(\mathbf{V}_1^t,\mathbf{V}_2^t,\ldots,\mathbf{V}_N^t\right)&=\sum_{i,j=1}^{N}\|\mathcal{Y}_i-\mathcal{Y}_j\|_F^2\tau(\mathbf{D})_{ij}\\
&\geqslant\left(\sum_{i,j=1}^{N}\|\mathcal{Y}_i-\mathcal{Y}_j\|_F^2\right)\times\min[\tau(\mathbf{D})_{ij}]\\
&\geqslant\left(\sum_{i,j=1}^{N}\|\mathcal{X}_i-\mathcal{X}_j\|_F^2\right)\times\min[\tau(\mathbf{D})_{ij}]
\end{aligned}\tag{17}$$

where the second inequality results from the fact that the distance between embedded data points is not larger than that between the corresponding original data points. If the input data are given, both $\|\mathcal{X}_i - \mathcal{X}_j\|_F^2$ and $\min[\tau(\mathbf{D})_{ij}]$ are known, then the objective function is lower bounded by $\left(\sum_{i,j=1}^{N} \|\mathcal{X}_i - \mathcal{X}_j\|_F^2\right) \times \min[\tau(\mathbf{D})_{ij}]$.

From above we know that the objective function $J(\mathbf{V}_1, \ldots, \mathbf{V}_N)$ is nonincreasing and has a lower bound. Therefore, the procedure of MGPE will converge to a local optimal solution. □

With the explicit embedding matrices $\mathbf{V}_k$ $(k = 1, \ldots, N)$, the desired low-dimensional representation of both training data points and new data points can easily be acquired by the multilinear transformation $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{V}_1^T \times_2 \ldots \times_N \mathbf{V}_N^T$.

3.3. Tensor distance based multilinear multidimensional scaling

The first new algorithm is called tensor distance based multilinear multidimensional scaling (TD-MMDS), which is based on the classical MDS (Kruskal & Wish, 1977). An earlier version of TD-MMDS has been appeared in Liu and Liu (2009).

The objective of TD-MMDS is to preserve the TDs between all pairs of input data in the embedded space, which could be formulated as follows:

$$\underset{\mathbf{V}_k^T \mathbf{V}_k = \mathbf{I}_{I'_k}}{\operatorname{argmin}} J(\mathbf{V}_1, \ldots, \mathbf{V}_N) = \sum_{i,j=1}^{N} \|\mathcal{Y}_i - \mathcal{Y}_j\|_F^2 \tau(\mathbf{D}_{TD-MMDS})_{ij} \tag{18}$$

TD-MMDS first constructs the tensor distance matrix $\mathbf{D}_{TD-MMDS}$ using the pairwise TDs between data points, i.e., $\mathbf{D}_{TD-MMDS} = (d_{TD}(\mathcal{X}_i, \mathcal{X}_j))$. It then computes the inner product matrix $\tau(\mathbf{D}_{TD-MMDS})$:

$$\tau(\mathbf{D}_{TD-MMDS}) = -\frac{1}{2}\mathbf{HSH} \tag{19}$$

where $\mathbf{S}_{ij} = d_{TD}^2(\mathcal{X}_i, \mathcal{X}_j)$, and $\mathbf{H}$ is defined similarly as in Eq. (2). After generating the inner product matrix, TD-MMDS utilizes the MGPE strategy to obtain the transformation matrices. The detailed procedure of TD-MMDS is described in Algorithm 1.

Algorithm 1: Tensor distance based multilinear multidimensional scaling (TD-MMDS)

```
Input: Iteration number $T_{max}$, embedded low dimensions $I'_1, I'_2, \ldots, I'_N$, metric matrix $\mathbf{G}$, and training dataset $I$: $\{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}\}$
Output: Transformation matrices $\mathbf{V}_k = \mathbf{V}_k^t \in \mathbb{R}^{I_k \times I'_k} (k = 1, \ldots, N)$
for all pair $(\mathcal{X}_i, \mathcal{X}_j)$ do
    $d_{TD}(\mathcal{X}_i, \mathcal{X}_j) = \sqrt{\sum_{l,m=1}^{I_1 \times I_2 \times \ldots \times I_N} g_{lm} \left(x_l^i - y_l^j\right)\left(x_m^i - y_m^j\right)}$;
end
compute inner product matrix $\tau(\mathbf{D}_{TD-MMDS})$ according to Eq. (19);
initialize $\mathbf{V}_k^0 = \mathbf{I}_{I_k} (k = 1, \ldots, N)$;
for $t = 1, \ldots, T_{max}$ do
    for $k = 1, \ldots, N$ do
        $\mathcal{X}_i^k = \mathcal{X}_i \times_1 (\mathbf{V}_1^t)^T \ldots \times_{k-1} (\mathbf{V}_{k-1}^t)^T \times_{k+1} \left(\mathbf{V}_{k+1}^{t-1}\right)^T \ldots \times_N \left(\mathbf{V}_N^{t-1}\right)^T$;
        $\mathbf{X}_i^k \Leftarrow_k \mathcal{X}_i^k$;
        $\mathbf{H}^k = -\sum_{i,j=1}^{n} \left(\mathbf{X}_i^k - \mathbf{X}_j^k\right)\left(\mathbf{X}_i^k - \mathbf{X}_j^k\right)^T \tau(\mathbf{D}_{TD-MMDS})_{ij}$;
        $\mathbf{H}^k \mathbf{V}_k^t = \mathbf{V}_k^t \Lambda_k \left(\mathbf{V}_k^t \in \mathbb{R}^{I_k \times I'_k}\right)$;
    end
    if $\left\|\mathbf{V}_k^t - \mathbf{V}_k^{t-1}\right\|_F < \epsilon$ then
        break;
    end
end
```

Table 1
Training and test time cost of TD-MMDS and MDS.

	Training	Test
TD-MMDS	$O(n^2 I^{2N} + n^2 N I^{N+1} + N I^3)$	$O(n(I')^{2N} + \sum_{i=1}^{N} (I')^i I^{N+1-i})$
MDS	$O(n^2 I^N + n I^{2N} + I^{3N})$	$O((I')^N I^N)$

Table 1 lists the training and test time cost of TD-MMDS and classical MDS. For simplicity, we assume that the training data are of uniform size in each dimension, i.e., $I_1 = I_2 = \cdots = I_N = I$. The most demanding training steps of TD-MMDS are the computation of distance matrix $\mathbf{D}_{TD-MMDS}$, the generation of matrices $\mathbf{H}^k$, and the final eigendecomposition. The training time costs of these three steps are $O(n^2 I^{2N})$, $O(n^2 N I^{N+1})$, and $O(N I^3)$, respectively. So the total training time of TD-MMDS is $O(n^2 I^{2N} + n^2 N I^{N+1} + N I^3)$. For MDS, it generates two $I^N \times I^N$ matrices, i.e., $\mathbf{X}\tau(\mathbf{D}_E)\mathbf{X}^T$ (here $\mathbf{D}_E$ is the Euclidean distance matrix) and $\mathbf{X}\mathbf{X}^T$, and solves the generalized eigendecomposition problem, which have the complexities $O(n^2 I^N + n I^{2N})$ and $O(I^{3N})$, respectively. Therefore, the training time of MDS is $O(n^2 I^N + n I^{2N} + I^{3N})$. Although some iteration loops are required to achieve the convergence for TD-MMDS, the total training time cost of it is still lower than that of MDS for high-order data. Suppose that the embedded tensors $\mathcal{Y}_i$ are as well of uniform size in each dimension, i.e., $I'_1 = I'_2 = \cdots = I'_N = I'$. For a new test data point, the test time cost of TD-MMDS is $O(n(I')^{2N} + \sum_{i=1}^{N} (I')^i I^{N+1-i})$. However, in MDS, the test time cost is $O((I')^N I^N)$. Since we have $I' \ll I$ in general, the test time cost of TD-MMDS is lower than that of MDS as well.

3.4. Tensor distance based multilinear isometric embedding

The second new algorithm derived from the TD-MGPE framework is called tensor distance based multilinear isometric embedding (TD-MIE), which is based on the representative manifold learning algorithm Isomap (Tenenbaum et al., 2000).

TD-MIE intends to preserve all pairwise distances calculated according to TDs along shortest paths in the neighborhood graph. The objective function of TD-MIE can be formulated as follows:

$$\underset{\mathbf{V}_k^T \mathbf{V}_k = \mathbf{I}_{I'_k}}{\operatorname{argmin}} J(\mathbf{V}_1, \ldots, \mathbf{V}_N) = \sum_{i,j=1}^{N} \|\mathcal{Y}_i - \mathcal{Y}_j\|_F^2 \tau(\mathbf{D}_{TD-MIE})_{ij} \tag{20}$$

First, a neighborhood graph G over all data points is constructed by connecting points $\mathcal{X}_i$ and $\mathcal{X}_j$ if $d_{TD}(\mathcal{X}_i, \mathcal{X}_j)$ is smaller than a threshold ϵ, or if $\mathcal{X}_i$ (or $\mathcal{X}_j$) is one of the k-nearest neighbors of $\mathcal{X}_j$ (or $\mathcal{X}_i$) (Here the neighborhood relationship is measured by the proposed TD metric). Then the edge lengths have been set equal to $d_{TD}(\mathcal{X}_i, \mathcal{X}_j)$.

Similar to Isomap, the second step of TD-MIE initializes $d_0(\mathcal{X}_i, \mathcal{X}_j) = d_{TD}(\mathcal{X}_i, \mathcal{X}_j)$ if $\mathcal{X}_i$ and $\mathcal{X}_j$ is connected by an edge; $d_0(\mathcal{X}_i, \mathcal{X}_j) = \infty$ otherwise. Then the shortest paths $d_{TD-G}(\mathcal{X}_i, \mathcal{X}_j)$ is obtained by Floyd's algorithm or Dijkstra's algorithm based on the TDs over the graph. The distance matrix of graph G is defined by $\mathbf{D}_{TD-MIE} = (d_{TD-G}(\mathcal{X}_i, \mathcal{X}_j))$, and thus we can get the inner product matrix $\tau(\mathbf{D}_{TD-MIE})$:

$$\tau(\mathbf{D}_{TD-MIE}) = -\frac{1}{2}\mathbf{HSH} \tag{21}$$

where $\mathbf{S}_{ij} = d_{TD-G}^2(\mathcal{X}_i, \mathcal{X}_j)$, and $\mathbf{H}$ is defined similarly as in Eq. (2).

Finally, TD-MIE uses the MGPE strategy to find the transformation matrices. The detailed procedure of TD-MIE is shown in Algorithm 2. Here $N(\mathcal{X}_i)$ denotes the set of nearest neighbors of data point $\mathcal{X}_i$. Note that we only show TD-MIE using the k nearest neighbor rule. Actually, the algorithm can also be implemented based on the ϵ nearest neighbor rule easily.

Algorithm 2: Tensor distance based multilinear isometric embedding (TD-MIE)

```
Input: Iteration number $T_{max}$, embedded low dimensions
  $I'_1, I'_2, \ldots, I'_N$, metric matrix $\mathbf{G}$, and training dataset $I$:
  $\{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}\}$
Output: Transformation matrices
  $\mathbf{V}_k = \mathbf{V}_k^t \in \mathbb{R}^{I_k \times I'_k}(k = 1, \ldots, N)$
for all pair $(\mathcal{X}_i, \mathcal{X}_j)$ do
    $d_{TD}(\mathcal{X}_i, \mathcal{X}_j) = \sqrt{\sum_{l,m=1}^{I_1 \times I_2 \times \ldots \times I_N} g_{lm}\left(x_l^i - y_l^j\right)\left(x_m^i - y_m^j\right)}$;
end
for all pair $(\mathcal{X}_i, \mathcal{X}_j)$ do
    if $\mathcal{X}_i \in N(\mathcal{X}_j)$ or $\mathcal{X}_j \in N(\mathcal{X}_i)$ then
        $d_0(\mathcal{X}_i, \mathcal{X}_j) = d_{TD}(\mathcal{X}_i, \mathcal{X}_j)$;
    else
        $d_0(\mathcal{X}_i, \mathcal{X}_j) = \infty$;
    end
end
for all pair $(\mathcal{X}_i, \mathcal{X}_j)$ do
    compute shortest paths $d_{TD-G}(\mathcal{X}_i, \mathcal{X}_j)$;
end
compute inner product matrix $\tau(\mathbf{D}_{TD-MIE})$ according to Eq. (21);
initialize $\mathbf{V}_k^0 = \mathbf{I}_{I_k}(k = 1, \ldots, N)$;
for $t = 1, \ldots, T_{max}$ do
    for $k = 1, \ldots, N$ do
        $\mathcal{X}_i^k = \mathcal{X}_i \times_1 (\mathbf{V}_1^t)^T \ldots \times_{k-1} (\mathbf{V}_{k-1}^t)^T \times_{k+1} \left(\mathbf{V}_{k+1}^{t-1}\right)^T \ldots \times_N \left(\mathbf{V}_N^{t-1}\right)^T$;
        $\mathbf{X}_i^k \Leftarrow_k \mathcal{X}_i^k$;
        $\mathbf{H}^k = -\sum_{i,j=1}^n (\mathbf{X}_i^k - \mathbf{X}_j^k)(\mathbf{X}_i^k - \mathbf{X}_j^k)^T \tau(\mathbf{D}_{TD-MIE})_{ij}$;
        $\mathbf{H}^k \mathbf{V}_k^t = \mathbf{V}_k^t \Lambda_k (\mathbf{V}_k^t \in \mathbb{R}^{I_k \times I'_k})$;
    end
    if $\|\mathbf{V}_k^t - \mathbf{V}_k^{t-1}\|_F < \epsilon$ then
        break;
    end
end
```

To make the embedding of new data points straightforward, Cai, He, and Han (2007) proposed a linear version of Isomap called isometric projection (IsoPro), which learns a transformation matrix $\mathbf{A}$ to map high-dimensional data to the low-dimensional space explicitly. Actually, IsoPro can be considered as a special case of TD-MIE. In the original tensor space, if we let the metric matrix to be the identity matrix, i.e., $\mathbf{G} = \mathbf{I}$, and assign the tensor order $N = 1$, then TD-MIE is reduced to IsoPro.

Table 2 lists the training and test time cost of TD-MIE and IsoPro. The most demanding training steps of TD-MIE are the computation of distance matrix $\mathbf{D}_{TD-MIE}$ using the TD metric, nearest neighbor searching, generation of the matrices $\mathbf{H}_k$, and final eigendecomposition. The training time costs of these four steps are $O(n^2I^{2N})$, $O(I^N n\log n)$ (Karger & Ruhl, 2002), $O(n^2NI^{N+1})$, and $O(NI^3)$, respectively. So the total training time of TD-MIE is $O(n^2I^{2N} + I^N n \log n + n^2NI^{N+1} + NI^3)$. For IsoPro, the training time of nearest neighbor searching is also $O(I^N n\log n)$. Furthermore, it generates two I^N - $\times$ I^N matrices, i.e., $\mathbf{X}\tau(\mathbf{D}_G)\mathbf{X}^T$ and $\mathbf{X}\mathbf{X}^T$, and solves the generalized eigendecomposition problem, which have the complexities $O(n^2I^N + nI^{2N})$ and $O(I^{3N})$, respectively. Therefore, the training time of IsoPro is $O(I^N n\log n + n^2I^N + nI^{2N} + I^{3N})$. Obviously, the total training time of TD-MIE is much less than that of IsoPro. Cai et al. (2007) also proposed a regression method to reduce the training time cost of IsoPro. However, the transformation vectors obtained by this method are least square approximations of desired transformation vectors, rather than the exact results. Therefore, some undesired errors will be brought into the final embedding results. For a new test datum, the test time cost of IsoPro is $O((I')^N I^N)$, and that of TD-MIE is $O(n(I')^{2N} + \sum_{i=1}^N (I')^i I^{N+1-i})$.

Table 2
Training and test time cost of TD-MIE and IsoPro.

	Training	Test
TD-MIE	$O(n^2I^{2N} + I^N n\log n + n^2NI^{N+1} + NI^3)$	$O(n(I')^{2N} + \sum_{i=1}^N (I')^i I^{N+1-i})$
IsoPro	$O(I^N n\log n + n^2I^N + nI^{2N} + I^{3N})$	$O((I')^N I^N)$

4. Experiments

In this section, we evaluate the proposed algorithms using image and video classification tasks on the following three databases: ORL face database (Samaria & Harter, 1994), USPS digit database (Hull, 1994), and Honda/UCSD video database (Lee, Ho, Yang, & Kriegman, 2005). The images in the first two databases are naturally second-order tensors, and the videos in the last database are third-order tensors.

The recognition process is composed of three steps. First, the subspace is calculated from the training dataset. Second, the test data are embedded into vector or tensor subspace according to different algorithms. Finally, the k nearest neighbor classifier is applied to low-dimensional subspace for classification. For our proposed algorithms, we simply set $\sigma = 1$ in all the experiments.

4.1. ORL face database

The ORL database contains 400 images of 40 different individuals (10 for each) (Samaria & Harter, 1994). Fig. 2 shows some examples from ORL database. All images are grayscale and normalized to a resolution of 64 × 64 pixels in our experiments.

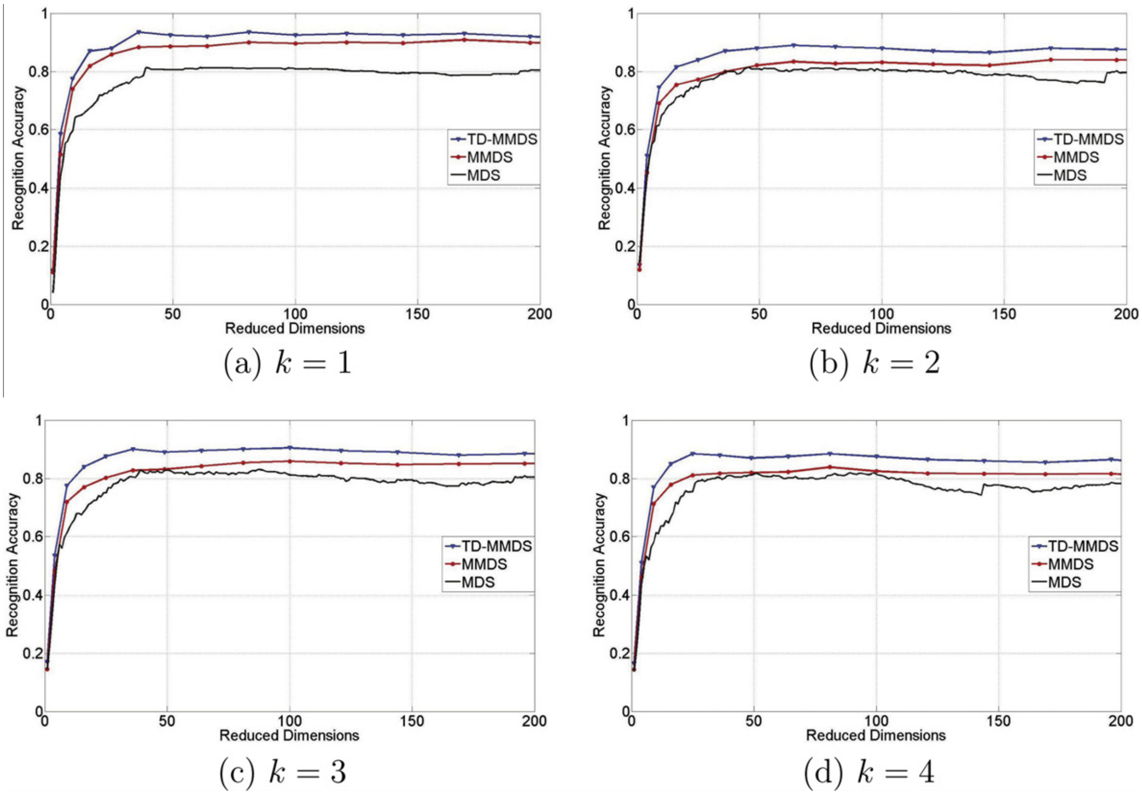
In this experiment, we compare TD-MMDS with MMDS and MDS (Kruskal & Wish, 1977), and compare TD-MIE with MIE (Liu, Liu, & Chan, 2009) and IsoPro (Cai et al., 2007). Here TD-MMDS and TD-MIE are two proposed tensor based dimensionality reduction algorithms with tensor distance; MMDS and MIE are two tensor based dimensionality reduction algorithms with Euclidean distance; MDS and IsoPro are two vector based dimensionality reduction algorithms with Euclidean distance. We compare the classification accuracy of these algorithms under different neighborhood sizes k. For each individual, we use 5 images for training and 5 images for test. For each value of k, we repeat the experiment 10 times on randomly selected training and test sets, and compute the average recognition accuracy. For the tensor based algorithms TD-MMDS, MMDS, TD-MIE, and MIE, we only show their performance in $(d \times d)$-dimensional tensor subspaces, i.e., 1, 4, 9, etc.

Fig. 3 shows the comparison results of TD-MMDS, MMDS, and MDS. Fig. 4 shows the results of TD-MIE, MIE, and IsoPro. Obviously, tensor based algorithms generally outperform vector based algorithms under different reduced dimensions. Moreover, by utilizing the proposed TD metric in the embedding process, TD-MMDS and TD-MIE achieve better results than MMDS and MIE, respectively. In addition, the proposed algorithms are robust to the variation of values of k.

4.2. USPS digit database

The United State Postal Service (USPS) database (Hull, 1994) contains 11000 images of hand written digital characters from 0 to 9. Each digit has 1100 normalized grayscale images of size 16 × 16. Fig. 5 shows some examples from USPS database.

Fig. 2. Examples from ORL face database.



(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

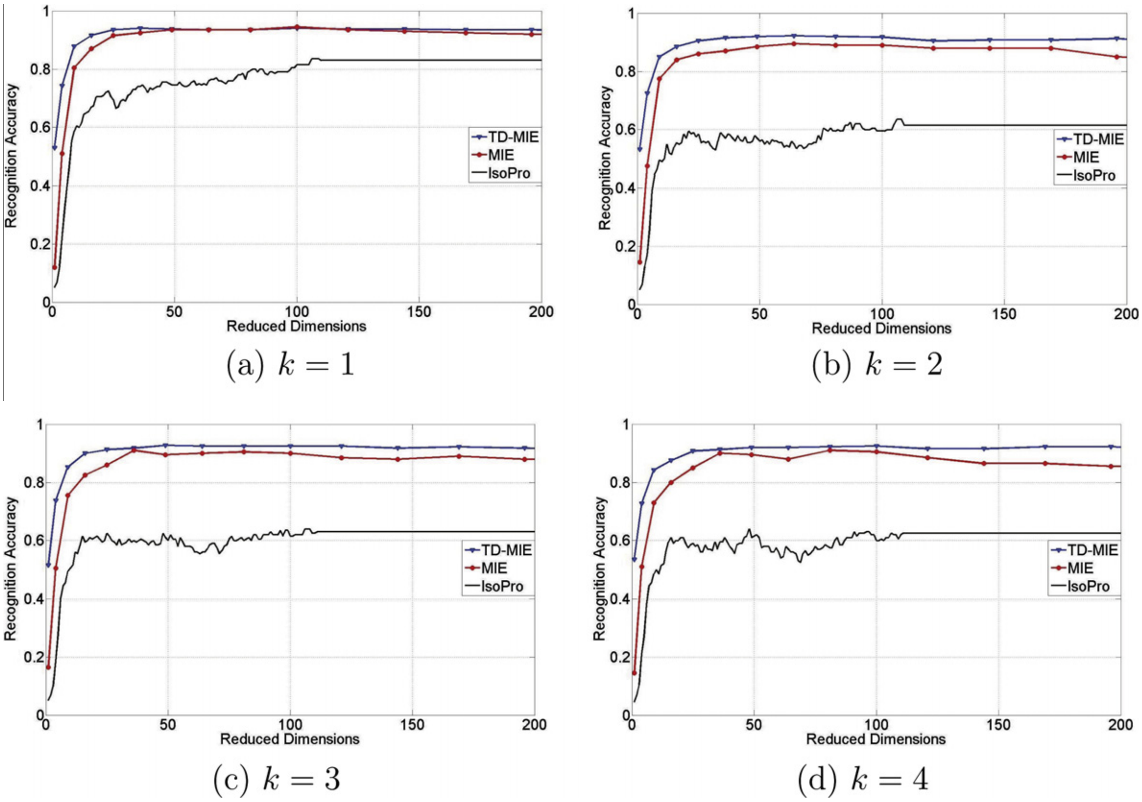
(d) $k = 4$

Fig. 3. Recognition accuracy of TD-MMDS, MMDS, and MDS on ORL face database with different neighborhood sizes k.

We conduct two experiments on USPS database. First, we compare TD-MMDS with MMDS and MDS, and compare TD-MIE with MIE and IsoPro under different numbers of training data. For each digit, p images are randomly selected for training and the remaining $(1100 - p)$ images are used for test. We repeat the experiment 10 times and calculate the average recognition accuracy for each algorithm. In this experiment, the neighborhood size k is fixed to 4. Similar as in previous experiment, for TD-MMDS, MMDS, TD-MIE, and MIE, we only show their performance in the $(d \times d)$-dimensional tensor subspace, i.e., 1, 4, 9, etc.

As shown in Figs. 6 and 7, for various numbers of training data, MMDS and MIE perform better than MDS and IsoPro respectively since they use the natural second-order representation as the input of embedding. Furthermore, compared with MMDS and MIE, the proposed TD-MMDS and TD-MIE achieve further improvement since the TD metric gives more faithful measure of the distances between high-order data points.

In the second experiment, we compare TD-MMDS and TD-MIE with other twelve typical vector based and tensor based dimensionality reduction algorithms: MDS, MMDS, IsoPro, MIE, PCA (Hotelling, 1933), MPCA (Lu et al., 2008), LDA (Fisher, 1936), MLDA (Yan & Xu et al., 2007), LPP (He & Niyogi, 2004), TLPP (Dai & Yeung, 2006), NPE (He, Cai, Yan, & Zhang, 2005), and TNPE (Dai & Yeung, 2006). Here LPP and TLPP are the linear and multilinear versions of a representative nonlinear dimensionality reduction algorithm Laplacian eigenmaps (LE) (Belkin & Niyogi, 2002), respectively. NPE and TNPE are the linear and multilinear versions of another nonlinear dimensionality



(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

(d) $k = 4$

Fig. 4. Recognition accuracy of TD-MIE, MIE, and IsoPro on ORL face database with different neighborhood sizes *k*.

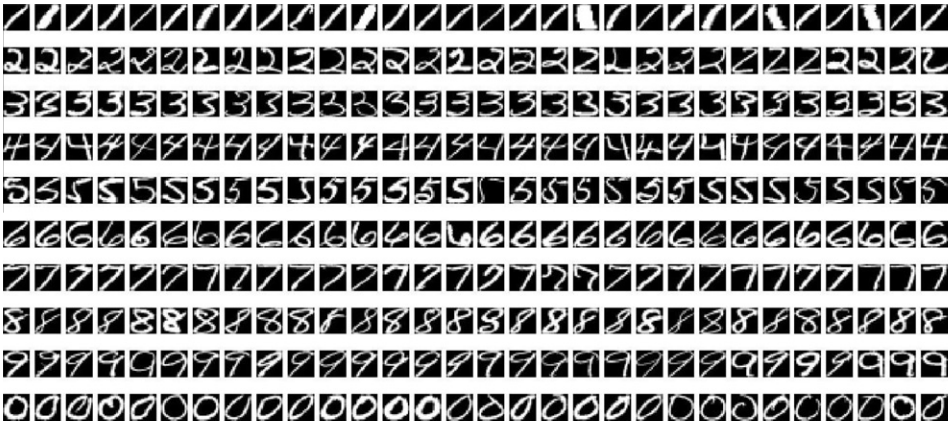
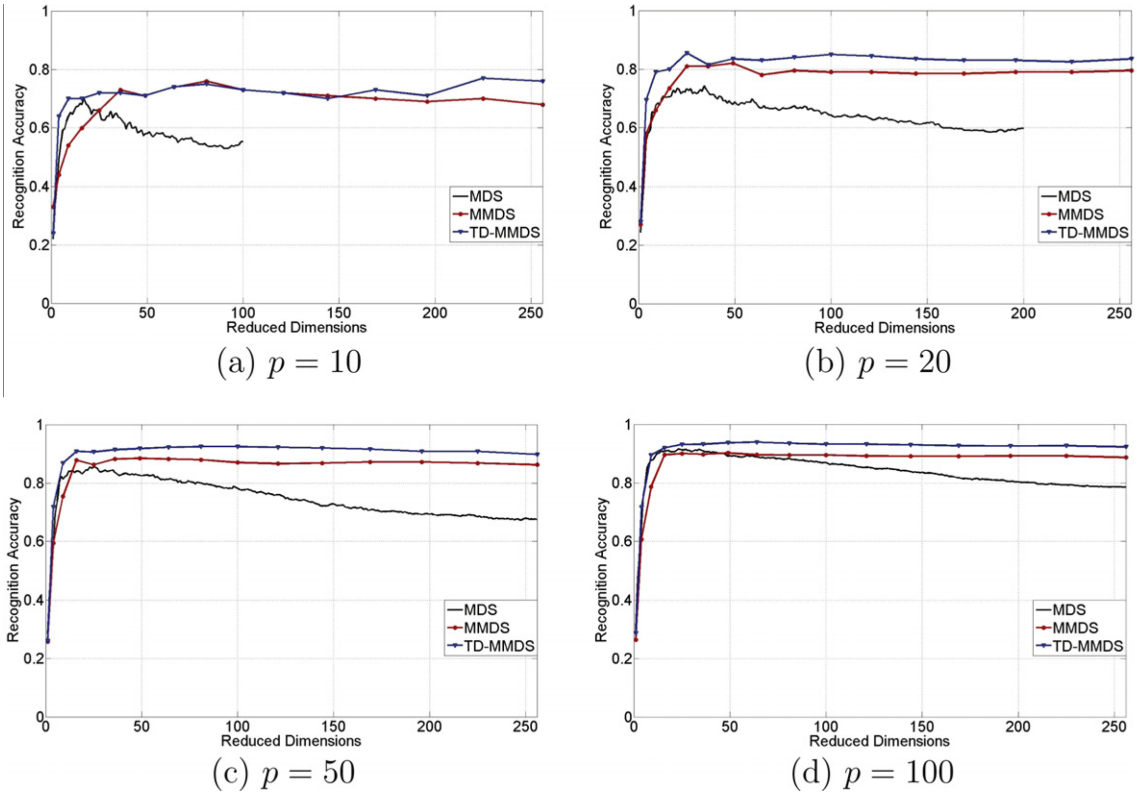


Fig. 5. Examples from USPS digit database.

reduction algorithm locally linear embedding (LLE) (Roweis et al., 2000), respectively. Furthermore, we apply TD on the vector based algorithms MDS and IsoPro, which are denoted by TD-MDS and TD-IsoPro in Table 3, respectively. In this experiment, we fix the training data number $p = 100$ and the neighborhood size $k = 4$. We repeat the experiment 10 times and calculate the average recognition accuracy. Similar as the proposed algorithms which include a regularization parameter, other four algorithms LPP, TLPP, NPE, and TNPE also contain such a regularization parameter σ. To make the comparison fair, we set $\sigma = 1$ for these algorithms. Table 3 lists the best recognition results and corresponding optimal reduced dimensions of all algorithms. The proposed TD-MMDS and TD-MIE achieve higher recognition accuracy than all other algorithms.

One interesting observation is the performance comparison of MDS with TD-MDS, MMDS, and TD-MMDS. TD-MDS intends to improve the performance of vector based algorithm MDS by utilizing the tensor distance metric, but compared with MDS, the accuracy increasing of TD-MDS is not distinct. Moreover, the recognition accuracy of MMDS even decreases when multilinear embedding strategy is applied to MDS. However, by integrating the tensor distance and tensor embedding, TD-MDS improves the recognition accuracy obviously. The comparison of IsoPro with TD-IsoPro, MIE, and TD-MIE holds the similar situation. These experiment results provide the empirical support to the argument that the unified tensor based dimensionality reduction procedure is indeed the key of the good performance.



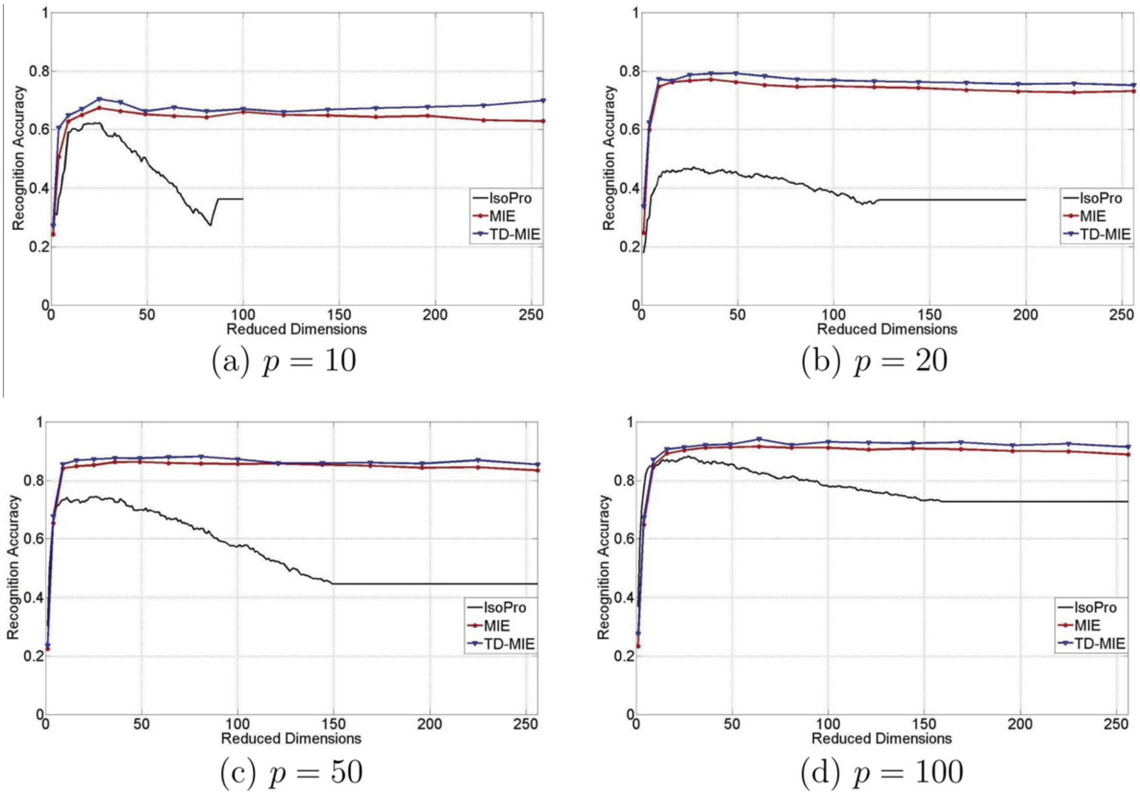
(a) $p = 10$

(b) $p = 20$

(c) $p = 50$

(d) $p = 100$

Fig. 6. Recognition accuracy of TD-MMDS, MMDS, and MDS on USPS digit database with different training data numbers *p*.



(a) $p = 10$

(b) $p = 20$

(c) $p = 50$

(d) $p = 100$

Fig. 7. Recognition accuracy of TD-MIE, MIE, and IsoPro on USPS digit database with different training data numbers *p*.

Table 3
Recognition accuracy (%) as well as the corresponding optimal reduced dimensions of sixteen vector based and tensor based dimensionality reduction algorithms on USPS digit database.

Methods	TD-MMDS	TD-MIE	MMDS	MIE
Recognition accuracy	**93.9**	**94.1**	90.7	91.5
Reduced dimensions	8^2	8^2	7^2	8^2
Methods	TD-MDS	TD-IsoPro	MDS	IsoPro
Recognition accuracy	91.8	91.3	91.5	88.3
Reduced dimensions	27	26	23	27
Methods	MLDA	LDA	MPCA	PCA
Recognition accuracy	91.8	89.1	87.4	82.9
Reduced dimensions	6^2	20	12^2	29
Methods	TLPP	LPP	TNPE	NPE
Recognition accuracy	91.0	85.2	91.2	87.6
Reduced dimensions	13^2	38	6^2	22

4.3. Honda/UCSD video database

In this subsection, we use the first data set from a standard video database: the Honda/UCSD video database (Lee et al., 2005) to test the performance of proposed algorithm. This data set contains 75 videos from 20 human subjects. Each video sequence is recorded in an indoor environment at 15 frames per second, and each lasted for at least 15 seconds. The resolution of each video sequence is 640 × 480. Videos in this data set include variations in 2-D (in-plane) and 3-D (out-of-plane) head movement as well as in facial expression. Fig. 8 shows some examples of the same person in different videos from Honda/UCSD database. We can see that huge variations exist even in videos of the same person, which bring great challenge to the recognition task.

In our experiments, the original videos are downsampled into 64 × 48 pixels, and only the luminance channel is used to represent the original data. The range of luminance values is from 0 to 255 which reflect the gray level of each pixel (0 means pure black and 255 means pure white). In order to collect more training and test data, we further cut each video to several shorter ones of uniform length: 3 seconds, i.e., 45 frames. Therefore, the input data are third-order tensors of size 64 × 48 × 45. Note that even the data are downsampled, the size of **G** is still very large. To avoid store this big matrix in memory, we can calculate the TD of two high-order data points using the following formula:

$$d_{TD}(\mathcal{X},\mathcal{Y}) = \sqrt{\sum_{l,m=1}^{I_1\times I_2\times\ldots\times I_N} g_{lm}(x_l - y_l)(x_m - y_m)} \qquad (22)$$

in which the variables are defined similarly as those in Eq. (5). By summating g_{lm} one by one, we can obtain $d_{TD}(\mathcal{X},\mathcal{Y})$ without storing the whole matrix **G**.

Table 4
Recognition accuracy (%) of TD-MMDS on Honda/UCSD video database, with different reduced dimensions.

$d_1\backslash d_2$	1	2	3	4	5	6	7	8
$d_3 = 1$								
1	39.0	73.8	71.9	82.3	84.2	84.2	84.2	84.2
2	59.7	80.4	80.4	83.2	84.1	86.0	86.0	87.0
3	73.8	86.0	84.2	88.9	87.0	85.1	85.1	88.9
4	75.7	93.6	88.9	90.7	87.0	87.9	87.9	88.9
5	81.3	93.6	90.7	90.7	89.8	89.8	89.8	89.8
6	82.3	**94.5**	90.7	90.7	89.8	89.8	90.7	89.8
7	84.2	94.5	89.8	91.7	90.7	90.7	91.7	91.7
8	82.3	93.6	90.7	90.7	91.7	90.7	89.8	88.9
$d_3 = 2$								
1	34.3	70.1	70.1	84.2	83.2	84.2	81.3	82.3
2	56.9	80.4	77.6	84.2	85.1	86.0	86.0	87.0
3	71.9	85.1	83.2	87.0	86.0	85.1	85.1	88.9
4	74.8	88.9	86.0	89.8	87.9	87.9	87.9	88.9
5	79.5	91.7	89.8	90.7	89.8	89.8	89.8	87.9
6	82.3	90.7	89.8	90.7	89.8	89.8	88.9	87.9
7	82.3	93.6	88.9	91.7	90.7	90.7	90.7	88.9
8	82.3	92.6	88.9	90.7	90.7	88.9	87.9	88.9

Table 5
Recognition accuracy (%) of TD-MIE on Honda/UCSD video database, with different reduced dimensions.

$d_1\backslash d_2$	1	2	3	4	5	6	7	8
$d_3 = 1$								
1	24.6	42.8	52.9	60.0	72.1	77.1	80.2	79.2
2	53.9	65.0	78.1	80.2	81.2	83.2	91.3	91.3
3	63.0	78.1	81.2	85.2	87.2	91.3	**94.3**	93.3
4	79.2	79.2	86.2	86.2	92.3	91.3	89.9	87.2
5	82.1	79.2	85.2	89.3	89.3	91.3	91.3	90.3
6	84.6	77.2	86.2	89.3	90.3	90.3	91.3	90.3
7	85.2	83.6	88.3	90.3	89.3	89.3	89.3	89.3
8	84.2	86.4	91.3	90.3	89.3	89.3	89.3	89.3
$d_3 = 2$								
1	26.6	35.7	45.8	59.9	68.0	75.1	79.1	77.2
2	51.9	66.0	76.1	78.1	81.2	81.2	85.2	86.2
3	61.9	77.1	77.1	85.2	87.2	89.3	90.3	89.3
4	78.1	77.1	83.1	85.2	89.3	89.3	89.3	89.3
5	82.2	77.1	83.2	85.2	89.2	89.2	89.3	89.3
6	82.2	77.1	85.1	89.3	91.3	89.3	89.3	89.3
7	81.2	83.2	85.0	89.3	89.3	89.3	89.3	89.3
8	83.2	85.4	85.9	89.3	89.3	89.3	89.3	89.3

We also conduct two experiments on this dataset. In the first experiment, we report the recognition accuracy of TD-MMDS and TD-MIE with different reduced dimensions. For each individual, we randomly select 10 videos, 5 for training and 5 for test. Totally we have 100 training videos and 100 test videos. We fix the

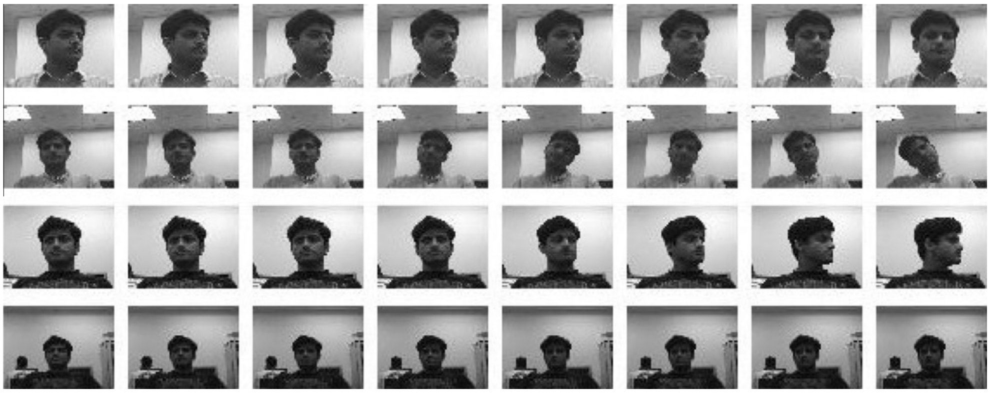


Fig. 8. Examples of the same person in different videos from Honda/UCSD video database.

Table 6
Recognition accuracy (%) as well as the corresponding optimal reduced dimensions of eight tensor based dimensionality reduction algorithms on Honda/UCSD video database.

Methods	TD-MMDS	TD-MIE	MMDS	MIE
Recognition accuracy	**94.5**	**94.3**	92.0	92.3
Reduced dimensions	$6 \times 2 \times 1$	$3 \times 7 \times 1$	$2 \times 5 \times 1$	$2 \times 4 \times 1$
Methods	MLDA	MPCA	TLPP	TNPE
Recognition accuracy	92.6	89.2	91.5	90.8
Reduced dimensions	$3 \times 3 \times 2$	$6 \times 6 \times 2$	$10 \times 5 \times 1$	$3 \times 6 \times 1$

neighborhood sizes $k = 4$. Like the previous experiments, we repeat the experiment 10 times and calculate the average recognition accuracy.

Tables 4 and 5 show the recognition accuracy of TD-MMDS and TD-MIE on different reduced dimensions, respectively. In this experiment, we vary the values of d_1, d_2, and d_3 from 1 to 8, respectively, and compute the recognition accuracy on all 8^3 combinations with different d_1, d_2, and d_3. In the experiment, an interesting observation is that the results converge very fast along the third dimension d_3, which indicates that the temporal dimension is highly redundant, so that all frames can be collapsed to a single one or two. Since the results converge fast along this dimension, we only show the recognition accuracy of TD-MMDS and TD-MIE with $d_3 = 1,2$. From the table, we can observe that when $d_1 = 6$, $d_2 = 2$, and $d_3 = 1$, TD-MMDS achieves a very good recognition accuracy at 94.5%; and when $d_1 = 3$, $d_2 = 7$, and $d_3 = 1$, TD-MIE also achieves a similar recognition accuracy at 94.3% in such a low-dimensional space.

In the second experiment, we compare TD-MMDS and TD-MIE with other six representative tensor based dimensionality reduction techniques: MMDS, MIE, MPCA, MLDA, TLPP, and TNPE. The experimental setting is same as that in the first experiment. The recognition accuracy and the corresponding optimal reduced dimensions ($d_1 \times d_2 \times d_3$) of these six tensor based algorithms are reported in Table 6. The proposed algorithms achieve very good results.

5. Conclusion

This paper proposes a unified framework TD-MGPE for tensor based dimensionality reduction with a distance metric TD and an embedding strategy MGPE. Unlike traditional Euclidean distance, TD measures the distances between high-order data by taking into account the correlations of different dimensions and orders. As a multilinear global strategy, MGPE directly works on more general tensor representation and aims to preserve distances between all data points. Based on the proposed framework, we further derive two new algorithms: TD-MMDS and TD-MIE. TD-MMDS learns the transformation matrices by keeping the TDs between all pairs of input data points in the embedded space, while TD-MIE intends to preserve all pairwise distances calculated according to TDs along shortest paths in the neighborhood graph. By integrating tensor distance and tensor based dimensionality reduction in a whole learning procedure, TD-MGPE shows attractive characteristics of preserving intrinsic tensor structure of data point, reflecting relationships between different data points, and keeping the global structure of datasets. Both theoretical analysis and empirical evaluation on image and video classification tasks validate the effectiveness of the proposed framework and algorithms.

References

Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems, 14*.

Cai, D., He, X., & Han, J. (2007). Isometric projection. In *Proceedings of The twenty-second AAAI conference on artificial intelligence* (pp. 528–533).

Carreira-Perpinan, M. A. (1996). A review of dimension reduction techniques. Tech. Rep. CS-96-09, Dept. of Computer Science, University of Sheffield.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). Introduction to algorithms. *2nd ed.*. MIT Press.

Cox, T. (2005). An introduction to multivariate data analysis. London: Hodder Arnold.

Dai, G., & Yeung, D.-Y. (2006). Tensor embedding methods. In *Proceedings of The twenty-first AAAI conference on artificial intelligence* (pp. 330–335).

Ding, C., & Ye, J. (2005). Two-dimensional singular value decomposition for 2d maps and images. In *Proceedings of SIAM international conference on data mining* (pp. 32–43).

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York: Wiley.

Ergin, S., Cakir, S., Nezih Gerek, Ö., & Gülmezoglu, M. B. (2011). A new implementation of common matrix approach using third-order tensors for face recognition. *Expert Systems with Applications, 38*(4), 3246–3251.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7*, 179–188.

Foley, D. H. (1972). Considerations of sample and feature size. *IEEE Transactions on Information Theory, IT-18*(5), 618–626.

Fu, Y., & Huang, T. S. (2008). Image classification using correlation tensor analysis. *IEEE Transactions on Image Processing, 17*(2), 226–234.

Gao, X., & Tian, C. (2009). Multi-view face recognition based on tensor subspace analysis and view manifold modeling. *Neurocomputing, 72*(16–18), 3742–3750.

Geng, X., Smith-Miles, K., Zhou, Z.-H., & Wang, L. (2011). Face image modeling by multilinear subspace analysis with missing values. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 41*(3), 881–892.

He, X., & Niyogi, P. (2004). Locality preserving projections. In *Advances in Neural Information Processing Systems, 16*.

He, X., Cai, D., & Niyogi, P. (2006). Tensor subspace analysis. *Advances in Neural Information Processing Systems, 506*, 499.

He, X., Cai, D., Yan, S., & Zhang, H.-J. (2005). Neighborhood preserving embedding. In *ICCV* (pp. 1208–1213).

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology, 24*, 498–520. 417–441.

Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16*(5), 550–554.

Karger, D., & Ruhl, M. (2002). Finding nearest neighbors in growth-restricted metrics. In *Proceedings of the 44th annual ACM symposium on theory of computing* (pp. 741–750).

Kolda, T. G. (2001). Orthogonal tensor decompositions. *SIAM Journal of Matrix Analysis and Applications, 23*(1), 243–255.

Kruskal, J. B., & Wish, M. (1977). *Multidimensional scaling*. Beverly Hills, CA: Sage Publications.

Lathauwer, L. D. (1997). Signal processing based on multilinear algebra. Ph.D. thesis, Katholike Universiteit Leuven.

Lathauwer, L. D., Moor, B. D., & Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM Journal of Matrix Analysis and Applications, 21*(4), 1253–1278.

Lee, K.-C., Ho, J., Yang, M.-H., & Kriegman, D. (2005). Visual tracking and recognition using probabilistic appearance manifolds. *Computer Vision and Image Understanding, 99*(3), 303–331.

Liu, Y., & Liu, Y. (2009). Tensor distance based multilinear multidimensional scaling for image and video analysis. In *Proceedings of the seventeenth ACM international conference on multimedia* (pp. 577–580).

Liu, Y., Liu, Y., & Chan, K. C. C. (2009). Multilinear isometric embedding for visual pattern analysis. In *Proceedings of the subspace2009 workshop* (pp. 212–218).

Liu, Y., Liu, Y., & Chan, K. C. C. (2010). Multilinear maximum distance embedding via l_1-norm optimization. In *Proceedings of the twenty-fourth AAAI conference on artificial intelligence* (pp. 525–530).

Luo, D., Ding, C. H. Q., & Huang, H. (2011). Multi-level cluster indicator decompositions of matrices and tensors. In *Proceedings of the twenty-fifth AAAI conference on artificial intelligence* (pp. 423–428).

Lu, H., Plataniotis, K. N., & Venetsanopoulos, A. N. (2008). Mpca: multilinear principal component analysis of tensor objects. *IEEE Transactions Neural Networks, 19*(1), 18–39.

Panagakis, Y., Kotropoulos, C., & Arce, G. (2010). Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification. *IEEE Transactions on Audio, Speech, and Language Processing, 18*(3), 576–588.

Roweis, S., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*(5500), 2323–2326.

Samaria, F. S., & Harter, A. C. (1994). Parameterisation of a stochastic model for human face identification. In *Proceedings of the second IEEE workshop on applications of computer vision* (pp. 138–142).

Tao, D., Li, X., Wu, X., Hu, W., & Maybank, S. J. (2007). Supervised tensor learning. *Knowledge and Information Systems, 13*(1), 1–42.

Tao, D., Li, X., Wu, X., & Maybank, S. J. (2007). General tensor discriminant analysis and gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(10), 1700–1715.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290*(5500), 2319–2323.

Tsai, F. S. (2012). A visualization metric for dimensionality reduction. *Expert Systems with Applications, 39*(2), 1747–1752.

van der Maaten, L., Postma, E., & van den Herik, H. (2009). Dimensionality reduction: a comparative review. Tech. Rep. TiCC-TR 2009-005, Tilburg University.

Vasilescu, M. A. O., & Terzopoulos, D. (2003). Multilinear subspace analysis of image ensembles. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (pp. 93–99).

Wang, L., Wang, X., & Feng, J. (2006). On image matrix based feature extraction algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 36*(1), 194–197.

Wang, L., Zhang, Y., & Feng, J. (2005). On the euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(8), 1334–1339.

Xu, D., Yan, S., Zhang, L., Lin, S., Zhang, H.-J., & Huang, T. S. (2008). Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis. *IEEE Transactions on Circuits Systems for Video Technology, 18*(1), 36–47.

Yang, J., Zhang, D., Frangi, A. F., & Yang, J. (2004). Two-dimensional pca: A new approach to appearance-based face represen-tation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(1), 131–137.

Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X., & Zhang, H. (2007). Multilinear discriminant analysis for face recognition. *IEEE Transactions on Image Processing, 16*(1), 212–220.

Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., & Lin, S. (2007). Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(1), 40–51.

Ye, J., Janardan, R., & Li, Q. (2004). Two-dimensional linear discriminant analysis. *Advances in Neural Information Processing Systems, 17*, 1569–1576.

Zhang, Z., & Chow, T. W. (2012). Maximum margin multisurface support tensor machines with application to image classification and segmentation. *Expert Systems with Applications, 39*(1), 849–860.