Research Article

Incremental Tensor Principal Component Analysis for Handwritten Digit Recognition

Chang Liu,^{1,2} Tao Yan,^{1,2} WeiDong Zhao,^{1,2} YongHong Liu,^{1,2} Dan Li,^{1,2} Feng Lin,³ and JiLiu Zhou³

¹ College of Information Science and Technology, Chengdu University, Chengdu 610106, China

² Key Laboratory of Pattern Recognition and Intelligent Information Processing, Institutions of Higher Education of Sichuan Province, Chengdu 610106, China

³ School of Computer Science, Sichuan University, Chengdu 610065, China

Correspondence should be addressed to YongHong Liu; 284424241@qq.com

Received 5 July 2013; Revised 21 September 2013; Accepted 22 September 2013; Published 30 January 2014

Academic Editor: Praveen Agarwal

Copyright © 2014 C. Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To overcome the shortcomings of traditional dimensionality reduction algorithms, incremental tensor principal component analysis (ITPCA) based on updated-SVD technique algorithm is proposed in this paper. This paper proves the relationship between PCA, 2DPCA, MPCA, and the graph embedding framework theoretically and derives the incremental learning procedure to add single sample and multiple samples in detail. The experiments on handwritten digit recognition have demonstrated that ITPCA has achieved better recognition performance than that of vector-based principal component analysis (PCA), incremental principal component analysis (IPCA), and multilinear principal component analysis (MPCA) algorithms. At the same time, ITPCA also has lower time and space complexity.

1. Introduction

Pattern recognition and computer vision require processing a large amount of multi-dimensional data, such as image and video data. Until now, a large number of dimensionality reduction algorithms have been investigated. These algorithms project the whole data into a low-dimensional space and construct new features by analyzing the statistical relationship hidden in the data set. The new features often give good information or hints about the data's intrinsic structure. As a classical dimensionality reduction algorithm, principal component analysis has been applied in various applications widely.

Traditional dimensionality reduction algorithms generally transform each multi-dimensional data into a vector by concatenating rows, which is called Vectorization. Such kind of the vectorization operation has largely increased the computational cost of data analysis and seriously destroys the intrinsic tensor structure of high-order data. Consequently, tensor dimensionality reduction algorithms are developed

based on tensor algebra [1–10]. Reference [10] has summarized existing multilinear subspace learning algorithms for tensor data. Reference [11] has generalized principal component analysis into tensor space and presented multilinear principal component analysis (MPCA). Reference [12] has proposed the graph embedding framework to unify all dimensionality reduction algorithms.

Furthermore, traditional dimensionality reduction algorithms generally employ off-line learning to deal with new added samples, which aggravates the computational cost. To address this problem, on-line learning algorithms are proposed [13, 14]. In particular, reference [15] has developed incremental principal component analysis (IPCA) based on updated-SVD technique. But most on-line learning algorithms focus on vector-based methods, only a limited number of works study incremental learning in tensor space [16–18].

To improve the incremental learning in tensor space, this paper presents incremental tensor principal component analysis (ITPCA) based on updated-SVD technique combining tensor representation with incremental learning.

This paper proves the relationship between PCA, 2DPCA, MPCA, and the graph embedding framework theoretically and derives the incremental learning procedure to add single sample and multiple samples in detail. The experiments on handwritten digit recognition have demonstrated that ITPCA has achieved better performance than vector-based incremental principal component analysis (IPCA) and multilinear principal component analysis (MPCA) algorithms. At the same time, ITPCA also has lower time and space complexity than MPCA.

2. Tensor Principal Component Analysis

In this section, we will employ tensor representation to express high-dimensional image data. Consequently, a high-dimensional image dataset can be expressed as a tensor dataset $X = \{X_1, \dots, X_M\}$, where $X_i \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is an N dimensional tensor and M is the number of samples in the dataset. Based on the representation, the following definitions are introduced.

Definition 1. For tensor dataset X , the mean tensor is defined as follows:

$$\bar{X} = \frac{1}{M} \sum_{i=1}^M X_i \in \mathbb{R}^{I_1 \times \dots \times I_N}. \quad (1)$$

Definition 2. The unfolding matrix of the mean tensor along the n th dimension is called the mode- n mean matrix and is defined as follows:

$$\bar{X}^{(n)} = \frac{1}{M} \sum_{i=1}^M X_i^{(n)} \in \mathbb{R}^{I_n \times \prod_{i \neq n} I_i}. \quad (2)$$

Definition 3. For tensor dataset X , the total scatter tensor is defined as follows:

$$\Psi_X = \sum_{m=1}^M \|X_m - \bar{X}\|^2, \quad (3)$$

where $\|A\|$ is the norm of the tensor.

Definition 4. For tensor dataset X , the mode- n total scatter matrix is defined as follows:

$$C^{(n)} = \sum_{i=1}^M \left(X_i^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}^{(n)} \right)^T, \quad (4)$$

where $\bar{X}^{(n)}$ is the mode- n mean matrix and $X_i^{(n)}$ is the mode- n unfolding matrix of tensor X_i .

Tensor PCA is introduced in [11, 19]. The target is to compute N orthogonal projective matrices $\{U^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, \dots, N\}$ to maximize the total scatter tensor of the projected low-dimensional feature as follows:

$$\begin{aligned} f \{U^{(n)}, n = 1, \dots, N\} &= \arg \max_{U^{(n)}} \Psi_y \\ &= \arg \max_{U^{(n)}} \sum_{m=1}^M \|Y_m - \bar{Y}\|^2, \end{aligned} \quad (5)$$

where $Y_m = X_m \times_1 U^{(1)T} \times_2 U^{(2)T} \times \dots \times_{n-1} U^{(n-1)T} \times_{n+1} U^{(n+1)T} \times \dots \times_N U^{(N)T}$.

Since it is difficult to solve N orthogonal projective matrices simultaneously, an iterative procedure is employed to approximately compute these N orthogonal projective matrices. Generally, since it is assumed that the projective matrices $\{U^{(1)}, \dots, U^{(n-1)}, U^{(n+1)}, \dots, U^{(N)}\}$ are known, we can solve the following optimized problem to obtain $U^{(n)}$:

$$\arg \max \sum_{m=1}^M \left(C_m^{(n)} C_m^{(n)T} \right), \quad (6)$$

where $C_m = (X_m - \bar{X}) \times_1 U^{(1)T} \times_2 U^{(2)T} \times \dots \times_{n-1} U^{(n-1)T} \times_{n+1} U^{(n+1)T} \times \dots \times_N U^{(N)T}$ and $C_m^{(n)}$ is the mode- n unfolding matrix of tensor C_m .

According to the above analysis, it is easy to derive the following theorems.

Theorem 5 (see [11]). *For the order of tensor data $n = 1$, that is, for the first-order tensor, the objective function of MPCA is equal to that of PCA.*

Proof. For the first-order tensor, $X_m \in \mathbb{R}^{I \times 1}$ is a vector, then (6) is

$$\sum_{m=1}^M \left(C_m^{(n)} C_m^{(n)T} \right) = \sum_{m=1}^M \left(U^T (X_m - \bar{X}) (X_m - \bar{X})^T U \right). \quad (7)$$

So MPCA for first-order tensor is equal to vector-based PCA. $\square$

Theorem 6 (see [11]). *For the order of tensor data $n = 2$, that is, for the second-order tensor, the objective function of MPCA is equal to that of 2DsPCA.*

Proof. For the second-order tensor, $X_m \in \mathbb{R}^{I_1 \times I_2}$ is a matrix; it is needed to solve two projective matrices $U^{(1)}$ and $U^{(2)}$, then (5) becomes

$$\sum_{m=1}^M \|Y_m - \bar{Y}\|^2 = \sum_{m=1}^M \|U^{(1)T} (X_m - \bar{X}) U^{(2)}\|^2. \quad (8)$$

The above equation exactly is the objective function of B2DPCA (bidirectional 2DPCA) [20–22]. Letting $U^{(2)} = I$, the projective matrix $U^{(1)}$ is solved. In this case, the objective function is

$$\sum_{m=1}^M \|Y_m - \bar{Y}\|^2 = \sum_{m=1}^M \|U^{(1)T} (X_m - \bar{X}) I\|^2. \quad (9)$$

Then the above equation is simplified into the objective function of row 2DPCA [23, 24]. Similarly, letting $U^{(1)} = I$, the projective matrix $U^{(2)}$ is solved; the objective function is

$$\sum_{m=1}^M \|Y_m - \bar{Y}\|^2 = \sum_{m=1}^M \|I^T (X_m - \bar{X}) U^{(2)}\|^2. \quad (10)$$

Then the above equation is simplified into the objective function of column 2DPCA [23, 24]. $\square$

Although vector-based and 2DPCA can be respected as the special cases of MPCA, MPCA and 2DPCA employ different techniques to solve the projective matrices. 2DPCA carries out PCA to row data and column data, respectively, and MPCA employs an iterative solution to compute N projective matrices. If it is supposed that the projective matrices $\{U^{(1)}, \dots, U^{(n-1)}, U^{(n+1)}, \dots, U^{(N)}\}$ are known, then $U^{(n)}$ is solved. Equation (6) can be expressed as follows:

$$\begin{aligned} C^{(n)} &= \sum_{i=1}^M \left(\left(X_i^{(n)} - \bar{X}^{(n)} \right) \times_k U^{(k)T} \Big|_{\substack{k=1 \\ k \neq n}}^N \right) \\ &\quad \times \left(\left(X_i^{(n)} - \bar{X}^{(n)} \right) \times_k U^{(k)T} \Big|_{\substack{k=1 \\ k \neq n}}^N \right)^T \\ &= \sum_{i=1}^M \left(\left(X_i^{(n)} - \bar{X}^{(n)} \right) U^{(-n)} \right) \left(\left(X_i^{(n)} - \bar{X}^{(n)} \right) U^{(-n)} \right)^T \\ &= \sum_{i=1}^M \left(\left(X_i^{(n)} - \bar{X}^{(n)} \right) U^{(-n)} U^{(-n)T} \left(X_i^{(n)} - \bar{X}^{(n)} \right)^T \right), \end{aligned} \quad (11)$$

where $U^{(-n)} = U^{(N)} \dots \otimes U^{(n+1)} \otimes U^{(n-1)} \dots \otimes U^{(1)}$.

Because

$$\begin{aligned} U^{(-n)} U^{(-n)T} &= \left(U^{(N)} \dots \otimes U^{(n+1)} \otimes U^{(n-1)} \dots \otimes U^{(1)} \right) \\ &\quad \times \left(U^{(N)} \dots \otimes U^{(n+1)} \otimes U^{(n-1)} \dots \otimes U^{(1)} \right)^T. \end{aligned} \quad (12)$$

Based on the Kronecker product, we can get the following:

$$\begin{aligned} (A \otimes B)^T &= A^T \otimes B^T, \\ (A \otimes B)(C \otimes D) &= AC \otimes BD. \end{aligned} \quad (13)$$

So

$$\begin{aligned} U^{(-n)} U^{(-n)T} &= U^{(N)} U^{(N)T} \dots \otimes U^{(n+1)} U^{(n+1)T} \\ &\quad \otimes U^{(n-1)} U^{(n-1)T} \dots \otimes U^{(1)} U^{(1)T}. \end{aligned} \quad (14)$$

Since $U^{(i)} \in \mathbb{R}^{I_i \times I_i}$ is an orthogonal matrix, $U^{(i)} U^{(i)T} = I$, $i = 1, \dots, N$, $i \neq n$, and $U^{(-n)} U^{(-n)T} = I$.

If the dimensions of projective matrices do not change in iterative procedure, then

$$C^{(n)} = \sum_{i=1}^K \left(X_i^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}^{(n)} \right)^T. \quad (15)$$

The above equation is equal to B2DPCA. Because MPCA updates projective matrices during iterative procedure, it has achieved better performance than 2DPCA.

Theorem 7. MPCA can be unified into the graph embedding framework [12].

Proof. Based on the basic knowledge of tensor algebra, we can get the following:

$$\sum_{m=1}^M \|Y_m - \bar{Y}\|^2 = \sum_{m=1}^M \|\text{vec}(Y_m) - \text{vec}(\bar{Y})\|^2. \quad (16)$$

Letting $y_m = \text{vec}(Y_m)$, $\mu = \text{vec}(\bar{Y})$, we can get the following:

$$\begin{aligned} &\sum_{i=1}^K \|y_i - \mu\|^2 \\ &= \sum_{i=1}^K (y_i - \mu)(y_i - \mu)^T \\ &= \sum_{i=1}^K \left(y_i - \frac{1}{N} \sum_{j=1}^N y_j \right) \left(y_i - \frac{1}{N} \sum_{j=1}^N y_j \right)^T \\ &= \sum_{i=1}^K \left(y_i y_i^T - \frac{1}{N} y_i \left(\sum_{j=1}^K y_j \right)^T - \frac{1}{N} \left(\sum_{j=1}^K y_j \right) y_i^T \right. \\ &\quad \left. + \frac{1}{N^2} \left(\sum_{j=1}^K y_j \right) \left(\sum_{j=1}^K y_j \right)^T \right) \\ &= \sum_{i=1}^K y_i y_i^T - \frac{1}{N} \sum_{i=1}^K y_i \left(\sum_{j=1}^K y_j \right)^T \\ &\quad - \frac{1}{N} \sum_{i=1}^K \sum_{j=1}^K y_j y_i^T + \frac{1}{N} \left(\sum_{j=1}^K y_j \right) \left(\sum_{j=1}^K y_j \right)^T \\ &= \sum_{i=1}^K y_i y_i^T - \frac{1}{N} \sum_{i=1}^K \sum_{j=1}^K y_i y_j^T \\ &= \sum_{i=1}^K \left(\sum_{j=1}^K w_{ij} \right) y_i y_i^T - \sum_{i,j=1}^K w_{ij} y_i y_j^T \\ &= \frac{1}{2} \sum_{i,j=1}^K w_{ij} (y_i y_i^T + y_j y_j^T - y_i y_j^T - y_j y_i^T) \\ &= \frac{1}{2} \sum_{i,j=1}^K w_{ij} (y_i - y_j)(y_i - y_j)^T \\ &= \frac{1}{2} \sum_{i,j=1}^K w_{ij} \|y_i - y_j\|_F^2, \end{aligned} \quad (17)$$

where the similarity matrix $W \in \mathbb{R}^{M \times M}$; for any i, j , we have $W_{ij} = 1/K$. So (16) can be written as follows:

$$\begin{aligned} & \sum_{m=1}^M \|Y_m - \bar{Y}\|^2 \\ &= \frac{1}{2} \sum_{i,j=1}^M W_{ij} \|Y_i - Y_j\|^2 \\ &= \frac{1}{2} \sum_{i,j=1}^M W_{ij} \left\| X_i \times {}_n U^{(n)} \Big|_{n=1}^N - X_j \times {}_n U^{(n)} \Big|_{n=1}^N \right\|^2. \end{aligned} \quad (18)$$

So the theorem is proved. $\square$

3. Incremental Tensor Principal Component Analysis

3.1. Incremental Learning Based on Single Sample. Given initial training samples $X_{\text{old}} = \{X_1, \dots, X_K\}$, $X_i \in \mathbb{R}^{I_1 \times \dots \times I_N}$, when a new sample $X_{\text{new}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is added, the training dataset becomes $X = \{X_{\text{old}}, X_{\text{new}}\}$.

The mean tensor of initial samples is

$$\bar{X}_{\text{old}} = \frac{1}{K} \sum_{i=1}^K X_i. \quad (19)$$

The covariance tensor of initial samples is

$$C_{\text{old}} = \sum_{i=1}^K \|X_i - \bar{X}_{\text{old}}\|^2. \quad (20)$$

The mode- n covariance matrix of initial samples is

$$C_{\text{old}}^{(n)} = \sum_{i=1}^K (X_i^{(n)} - \bar{X}_{\text{old}}^{(n)}) (X_i^{(n)} - \bar{X}_{\text{old}}^{(n)})^T. \quad (21)$$

When the new sample is added, the mean tensor is

$$\begin{aligned} \bar{X} &= \frac{1}{K+1} \sum_{i=1}^{K+1} X_i \\ &= \frac{1}{K+1} \left(\sum_{i=1}^K X_i + X_{\text{new}} \right) = \frac{1}{K+1} (K\bar{X}_{\text{old}} + X_{\text{new}}). \end{aligned} \quad (22)$$

The mode- n covariance matrix is expressed as follows:

$$\begin{aligned} C^{(n)} &= \sum_{i=1}^{K+1} (X_i^{(n)} - \bar{X}^{(n)}) (X_i^{(n)} - \bar{X}^{(n)})^T \\ &= \sum_{i=1}^K (X_i^{(n)} - \bar{X}^{(n)}) (X_i^{(n)} - \bar{X}^{(n)})^T \\ &\quad + (X_{\text{new}}^{(n)} - \bar{X}^{(n)}) (X_{\text{new}}^{(n)} - \bar{X}^{(n)})^T, \end{aligned} \quad (23)$$

where the first item of (23) is

$$\begin{aligned} & \sum_{i=1}^K (X_i^{(n)} - \bar{X}^{(n)}) (X_i^{(n)} - \bar{X}^{(n)})^T \\ &= \sum_{i=1}^K (X_i^{(n)} - \bar{X}_{\text{old}}^{(n)} + \bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)}) \\ &\quad \times (X_i^{(n)} - \bar{X}_{\text{old}}^{(n)} + \bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)})^T \\ &= \sum_{i=1}^K \left[(X_i^{(n)} - \bar{X}_{\text{old}}^{(n)}) + (\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)}) \right] \\ &\quad \times \left[(X_i^{(n)} - \bar{X}_{\text{old}}^{(n)})^T + (\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)})^T \right] \\ &= \sum_{i=1}^K (X_i^{(n)} - \bar{X}_{\text{old}}^{(n)}) (X_i^{(n)} - \bar{X}_{\text{old}}^{(n)})^T \\ &\quad + K (\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)}) (\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)})^T \\ &\quad + \sum_{i=1}^K (X_i^{(n)} - \bar{X}_{\text{old}}^{(n)}) (\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)})^T \\ &\quad + (\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)}) (X_i^{(n)} - \bar{X}_{\text{old}}^{(n)})^T \\ &= \sum_{i=1}^K (X_i^{(n)} - \bar{X}_{\text{old}}^{(n)}) (X_i^{(n)} - \bar{X}_{\text{old}}^{(n)})^T \\ &\quad + K (\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)}) (\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)})^T \\ &= C_{\text{old}}^{(n)} + K \left(\bar{X}_{\text{old}}^{(n)} - \frac{K\bar{X}_{\text{old}}^{(n)} + X_{\text{new}}^{(n)}}{K+1} \right) \\ &\quad \times \left(\bar{X}_{\text{old}}^{(n)} - \frac{K\bar{X}_{\text{old}}^{(n)} + X_{\text{new}}^{(n)}}{K+1} \right)^T \\ &= C_{\text{old}}^{(n)} + \frac{K}{(K+1)^2} (\bar{X}_{\text{old}}^{(n)} - X_{\text{new}}^{(n)}) (\bar{X}_{\text{old}}^{(n)} - X_{\text{new}}^{(n)})^T. \end{aligned} \quad (24)$$

The second item of (23) is

$$\begin{aligned} & (X_{\text{new}}^{(n)} - \bar{X}^{(n)}) (X_{\text{new}}^{(n)} - \bar{X}^{(n)})^T \\ &= (X_{\text{new}}^{(n)} - \bar{X}^{(n)}) (X_{\text{new}}^{(n)} - \bar{X}^{(n)})^T \\ &= \left(X_{\text{new}}^{(n)} - \frac{K\bar{X}_{\text{old}}^{(n)} + X_{\text{new}}^{(n)}}{K+1} \right) \left(X_{\text{new}}^{(n)} - \frac{K\bar{X}_{\text{old}}^{(n)} + X_{\text{new}}^{(n)}}{K+1} \right)^T \\ &= \frac{K^2}{(K+1)^2} (\bar{X}_{\text{old}}^{(n)} - X_{\text{new}}^{(n)}) (\bar{X}_{\text{old}}^{(n)} - X_{\text{new}}^{(n)})^T. \end{aligned} \quad (25)$$

Consequently, the mode- n covariance matrix is updated as follows:

$$C^{(n)} = C_{\text{old}}^{(n)} + \frac{K}{K+1} \left(\bar{X}_{\text{old}}^{(n)} - X_{\text{new}}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - X_{\text{new}}^{(n)} \right)^T. \quad (26)$$

Therefore, when a new sample is added, the projective matrices are solved according to the eigen decomposition on (26).

3.2. Incremental Learning Based on Multiple Samples. Given an initial training dataset $X_{\text{old}} = \{X_1, \dots, X_K\}$, $X_i \in \mathbb{R}^{I_1 \times \dots \times I_N}$, when new samples are added into training dataset, $X_{\text{new}} = \{X_{K+1}, \dots, X_{K+T}\}$, then training dataset becomes into $X = \{X_1, \dots, X_K, X_{K+1}, \dots, X_{K+T}\}$. In this case, the mean tensor is updated into the following:

$$\bar{X} = \frac{1}{K+T} \sum_{i=1}^{K+T} X_i = \frac{1}{K+T} \left(\sum_{i=1}^K X_i + \sum_{i=K+1}^{K+T} X_i \right) \quad (27)$$

$$= \frac{1}{K+T} (K\bar{X}_{\text{old}} + T\bar{X}_{\text{new}}).$$

Its mode- n covariance matrix is

$$\begin{aligned} C^{(n)} &= \sum_{i=1}^{K+T} \left(X_i^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}^{(n)} \right)^T \\ &= \sum_{i=1}^K \left(X_i^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}^{(n)} \right)^T \\ &\quad + \sum_{i=K+1}^{K+T} \left(X_i^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}^{(n)} \right)^T. \end{aligned} \quad (28)$$

The first item in (28) is written as follows:

$$\begin{aligned} &\sum_{i=1}^K \left(X_i^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}^{(n)} \right)^T \\ &= \sum_{i=1}^K \left(X_i^{(n)} - \bar{X}_{\text{old}}^{(n)} \right) \left(X_i^{(n)} - \bar{X}_{\text{old}}^{(n)} \right)^T \\ &\quad + K \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)} \right)^T \\ &\quad + \sum_{i=1}^K \left[\left(X_i^{(n)} - \bar{X}_{\text{old}}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)} \right)^T \right. \\ &\quad \left. + \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}_{\text{old}}^{(n)} \right)^T \right], \end{aligned} \quad (29)$$

where

$$\begin{aligned} &\sum_{i=1}^K \left[\left(X_i^{(n)} - \bar{X}_{\text{old}}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)} \right)^T \right. \\ &\quad \left. + \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}_{\text{old}}^{(n)} \right)^T \right] \\ &= K \bar{X}_{\text{old}}^{(n)} \bar{X}_{\text{old}}^{(n)T} - K \bar{X}_{\text{old}}^{(n)} \bar{X}^{(n)T} - K \bar{X}_{\text{old}}^{(n)} \bar{X}_{\text{old}}^{(n)T} \\ &\quad + K \bar{X}_{\text{old}}^{(n)} \bar{X}^{(n)T} + K \bar{X}_{\text{old}}^{(n)} \bar{X}_{\text{old}}^{(n)T} - K \bar{X}_{\text{old}}^{(n)} \bar{X}_{\text{old}}^{(n)T} \\ &\quad - K \bar{X}^{(n)} \bar{X}_{\text{old}}^{(n)T} + K \bar{X}^{(n)} \bar{X}_{\text{old}}^{(n)T} \\ &= 0, \\ &K \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}^{(n)} \right)^T \\ &= \frac{KT^2}{(K+T)^2} \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right)^T. \end{aligned} \quad (30)$$

Putting (30) into (29), then (29) becomes as follows:

$$\begin{aligned} &\sum_{i=1}^{K+T} \left(X_i^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}^{(n)} \right)^T \\ &= C_{\text{old}}^{(n)} + \frac{KT^2}{(K+T)^2} \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right)^T. \end{aligned} \quad (31)$$

The second item in (28) is written as follows:

$$\begin{aligned} &\sum_{i=K+1}^{K+T} \left(X_i^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}^{(n)} \right)^T \\ &= C_{\text{new}}^{(n)} + T \left(\bar{X}_{\text{new}}^{(n)} - \bar{X}^{(n)} \right) \left(\bar{X}_{\text{new}}^{(n)} - \bar{X}^{(n)} \right)^T, \end{aligned} \quad (32)$$

where

$$\begin{aligned} &T \left(\bar{X}_{\text{new}}^{(n)} - \bar{X}^{(n)} \right) \left(\bar{X}_{\text{new}}^{(n)} - \bar{X}^{(n)} \right)^T \\ &= \frac{K^2T}{(K+T)^2} \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right)^T \end{aligned} \quad (33)$$

Then (32) becomes as follows:

$$\begin{aligned} &\sum_{i=K+1}^{K+T} \left(X_i^{(n)} - \bar{X}^{(n)} \right) \left(X_i^{(n)} - \bar{X}^{(n)} \right)^T \\ &= C_{\text{new}}^{(n)} + \frac{K^2T}{(K+T)^2} \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right)^T \end{aligned} \quad (34)$$

Putting (31) and (34) into (28), then we can get the following:

$$C^{(n)} = C_{\text{old}}^{(n)} + C_{\text{new}}^{(n)} + \frac{KT}{K+T} \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right)^T. \quad (35)$$

It is worthy to note that when new samples are available, it has no need to recompute the mode- n covariance matrix of all training samples. We just have to solve the mode- n covariance matrix of new added samples and the difference between original training samples and new added samples. However, like traditional incremental PCA, eigen decomposition on $C^{(n)}$ has been repeated once new samples are added. It is certain that the repeated eigen decomposition on $C^{(n)}$ will cause heavy computational cost, which is called “the eigen decomposition updating problem.” For traditional vector-based incremental learning algorithm, the updated-SVD technique is proposed in [25] to fit the eigen decomposition. This paper will introduce the updated-SVD technique into tensor-based incremental learning algorithm.

For original samples, the mode- n covariance matrix is

$$C_{\text{old}}^{(n)} = \sum_{i=1}^K \left(X_i^{(n)} - \bar{X}_{\text{old}}^{(n)} \right) \left(X_i^{(n)} - \bar{X}_{\text{old}}^{(n)} \right)^T = S_{\text{old}}^{(n)} S_{\text{old}}^{(n)T}, \quad (36)$$

where $S_{\text{old}}^{(n)} = [X_1^{(n)} - \bar{X}_{\text{old}}^{(n)}, \dots, X_K^{(n)} - \bar{X}_{\text{old}}^{(n)}]$. According to the eigen decomposition $S_{\text{old}}^{(n)} = \text{svd}(U\Sigma V^T)$, we can get the following:

$$\begin{aligned} S_{\text{old}}^{(n)} S_{\text{old}}^{(n)T} &= (U\Sigma V^T) (U\Sigma V^T)^T \\ &= U\Sigma V^T V \Sigma U^T = U\Sigma^2 U^T = \text{eig}(C_{\text{old}}^{(n)}). \end{aligned} \quad (37)$$

So it is easy to derive that the eigen-vector of $C_{\text{old}}^{(n)}$ is the left singular vector of $S_{\text{old}}^{(n)}$ and the eigen-values correspond to the extraction of left singular values of $S_{\text{old}}^{(n)}$.

For new samples, the mode- n covariance matrix is

$$C_{\text{new}}^{(n)} = \sum_{i=K+1}^{K+T} \left(X_i^{(n)} - \bar{X}_{\text{new}}^{(n)} \right) \left(X_i^{(n)} - \bar{X}_{\text{new}}^{(n)} \right)^T = S_{\text{new}}^{(n)} S_{\text{new}}^{(n)T}, \quad (38)$$

where $S_{\text{new}}^{(n)} = [X_1^{(n)} - \bar{X}_{\text{new}}^{(n)}, \dots, X_K^{(n)} - \bar{X}_{\text{new}}^{(n)}]$. According to (35), the updated mode- n covariance matrix is defined as follows:

$$\begin{aligned} C^{(n)} &= C_{\text{old}}^{(n)} + C_{\text{new}}^{(n)} + \frac{KT}{K+T} \\ &\quad \times \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right) \left(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)} \right)^T = S^{(n)} S^{(n)T}, \end{aligned} \quad (39)$$

where $S^{(n)} = [S_{\text{old}}^{(n)}, S_{\text{new}}^{(n)}, \sqrt{KT/(K+T)}(\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)})]$. Therefore, the updated projective matrix $U^{(n)}$ is the eigen-vectors corresponding to the largest P_n eigen-values of $S^{(n)}$. The main steps of incremental tensor principal component analysis are listed as follows:

input: original samples and new added samples,

output: N projective matrices.

Step 1. Computing and saving

$$\text{eig}(C_{\text{old}}^{(n)}) \approx [U_r^{(n)}, \Sigma_r^{(n)}]. \quad (40)$$

Step 2. For $i = 1 : N$

$$B = \left[S_{\text{new}}^{(n)}, \sqrt{\frac{KT}{K+T}} (\bar{X}_{\text{old}}^{(n)} - \bar{X}_{\text{new}}^{(n)}) \right]. \quad (41)$$

Processing QR decomposition for the following equation:

$$QR = \left(I - U_r^{(n)} U_r^{(n)T} \right) B. \quad (42)$$

Processing SVD decomposition for the following equation:

$$\text{svd} \left[\begin{array}{c} \sqrt{\Sigma_r^{(n)}} U_r^{(n)T} B \\ 0 \quad R \end{array} \right] = \hat{U} \hat{\Sigma} \hat{V}^T. \quad (43)$$

Computing the following equation:

$$[S_{\text{old}}^{(n)}, B] \approx ([U_r^{(n)}, Q] \hat{U}) \hat{\Sigma} \left(\begin{bmatrix} V_r^{(n)} & 0 \\ 0 & I \end{bmatrix} \hat{V} \right)^T. \quad (44)$$

Then the updated projective matrix is computed as follows:

$$U^{(n)} = [U_r^{(n)}, Q] \hat{U}, \quad (45)$$

end.

Step 3. Repeating the above steps until the incremental learning is finished.

3.3. The Complexity Analysis. For tensor dataset $X = \{X_1, \dots, X_M\}$, $X_i \in \mathbb{R}^{I_1 \times \dots \times I_N}$, without loss of generality, it is assumed that all dimensions are equal, that is, $I_1 = \dots = I_N = I$.

Vector-based PCA converts all data into vector and constructs a data matrix $X \in \mathbb{R}^{M \times D}$, $D = I^N$. For vector-based PCA, the main computational cost contains three parts: the computation of the covariance matrix, the eigen decomposition of the covariance matrix, and the computation of low-dimensional features. The time complexity to compute covariance matrix is $O(MI^{2N})$, the time complexity of the eigen decomposition is $O(I^{3N})$, and the time complexity to compute low-dimensional features is $O(MI^{2N} + I^{3N})$.

Letting the iterative number be 1, the time complexity to computing the mode- n covariance matrix for MPCA is $O(MNI^{N+1})$, the time complexity of eigen decomposition is $O(NI^3)$, and the time complexity to compute low-dimensional features is $O(MNI^{N+1})$, so the total time complexity is $O(MNI^{N+1} + NI^3)$. Considering the time complexity, MPCA is superior to PCA.

For ITPCA, it is assumed that T incremental datasets are added; MPCA has to recompute mode- n covariance matrix and conducts eigen decomposition for initial dataset and incremental dataset. The more the training samples are, the higher time complexity they have. If updated-SVD is used, we only need to compute QR decomposition and SVD decomposition. The time complexity of QR decomposition is $O(NI^{N+1})$. The time complexity of the rank- k decomposition of the matrix with the size of $(r+I) \times (r+I^{N-1})$ is

FIGURE 1: The samples in USPS dataset.

$O(N(r + I)k^2)$. It can be seen that the time complexity of updated-SVD has nothing to do with the number of new added samples.

Taking the space complexity into account, if training samples are reduced into low-dimensional space and the dimension is $D = \prod_{n=1}^N d_n$, then PCA needs $D \prod_{n=1}^N I_n$ bytes to save projective matrices and MPCA needs $\sum_{n=1}^N I_n d_n$ bytes. So MPCA has lower space complexity than PCA. For incremental learning, both PCA and MPCA need $M \prod_{n=1}^N I_n$ bytes to save initial training samples; ITPCA only need $\sum_{n=1}^N I_n^2$ bytes to keep mode- n covariance matrix.

4. Experiments

In this section, the handwritten digit recognition experiments on the USPS image dataset are conducted to evaluate the performance of incremental tensor principal component analysis. The USPS handwritten digit dataset has 9298 images from zero to nine shown in Figure 1. For each image, the size is 16×16 . In this paper, we choose 1000 images and divide them into initial training samples, new added samples, and test samples. Furthermore, the nearest neighbor classifier is employed to classify the low-dimensional features. The recognition results are compared with PCA [26], IPCA [15], and MPCA [11].

At first, we choose 70 samples belonging to four classes from initial training samples. For each time of incremental learning, 70 samples which belong to the other two classes are added. So after three times, the class labels of the training samples are ten and there are 70 samples in each class. The resting samples of original training samples are considered as testing dataset. All algorithms are implemented in MATLAB 2010 on an Intel (R) Core (TM) i5-3210 M CPU @ 2.5 GHz with 4 G RAM.

Firstly, 36 PCs are preserved and fed into the nearest neighbor classifier to obtain the recognition results. The results are plotted in Figure 2. It can be seen that MPCA and ITPCA are better than PCA and IPCA for initial learning; the probable reason is that MPCA and ITPCA employ tensor representation to preserve the structure information.

The recognition results under different learning stages are shown in Figures 3, 4, and 5. It can be seen that the recognition results of these four methods always fluctuate violently

FIGURE 2: The recognition results for 36 PCs of the initial learning.

FIGURE 3: The recognition results of different methods of the first incremental learning.

when the numbers of low-dimensional features are small. However, with the increment of the feature number, the recognition performance keeps stable. Generally MPCA and ITPCA are superior to PCA and IPCA. Although ITPCA have comparative performance at first two learning, ITPCA begin to surmount MPCA after the third learning. Figure 6 has given the best recognition percents of different methods. We can get the same conclusion as shown in Figures 3, 4, and 5.

The time and space complexity of different methods are shown in Figures 7 and 8, respectively. Taking the time complexity into account, it can be found that at the stage of initial learning, PCA has the lowest time complexity. With

FIGURE 4: The recognition results of different methods of the second incremental learning.

FIGURE 5: The recognition results of different methods of the third incremental learning.

the increment of new samples, the time complexity of PCA and MPCA grows greatly and the time complexity of IPCA and ITPCA becomes stable. ITPCA has slower increment than MPCA. The reason is that ITPCA introduces incremental learning based on the updated-SVD technique and avoids decomposing the mode- n covariance matrix of original samples again. Considering the space complexity, it is easy to find that ITPCA has the lowest space complexity among four compared methods.

FIGURE 6: The comparison of recognition performance of different methods.

FIGURE 7: The comparison of time complexity of different methods.

5. Conclusion

This paper presents incremental tensor principal component analysis based on updated-SVD technique to take full advantage of redundancy of the space structure information and online learning. Furthermore, this paper proves that PCA and 2DPCA are the special cases of MPCA and all of them can be unified into the graph embedding framework. This

FIGURE 8: The comparison of space complexity of different methods.

paper also analyzes incremental learning based on single sample and multiple samples in detail. The experiments on handwritten digit recognition have demonstrated that principal component analysis based on tensor representation is superior to tensor principal component analysis based on vector representation. Although at the stage of initial learning, MPCA has better recognition performance than ITPCA, the learning capability of ITPCA becomes well gradually and exceeds MPCA. Moreover, even if new samples are added, the time and space complexity of ITPCA still keep slower increment.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This present work has been funded with support from the National Natural Science Foundation of China (61272448), the Doctoral Fund of Ministry of Education of China (20110181130007), the Young Scientist Project of Chengdu University (no. 2013XJZ21).

References

- [1] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 103–123, 2009.
- [2] C. Liu, K. He, J.-L. Zhou, and C.-B. Gao, "Discriminant orthogonal rank-one tensor projections for face recognition," in *Intelligent Information and Database Systems*, N. T. Nguyen, C.-G. Kim, and A. Janiak, Eds., vol. 6592 of *Lecture Notes in Computer Science*, pp. 203–211, 2011.
- [3] G.-F. Lu, Z. Lin, and Z. Jin, "Face recognition using discriminant locality preserving projections based on maximum margin criterion," *Pattern Recognition*, vol. 43, no. 10, pp. 3572–3579, 2010.
- [4] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1700–1715, 2007.
- [5] F. Nie, S. Xiang, Y. Song, and C. Zhang, "Extracting the optimal dimensionality for local tensor discriminant analysis," *Pattern Recognition*, vol. 42, no. 1, pp. 105–114, 2009.
- [6] Z.-Z. Yu, C.-C. Jia, W. Pang, C.-Y. Zhang, and L.-H. Zhong, "Tensor discriminant analysis with multiscale features for action modeling and categorization," *IEEE Signal Processing Letters*, vol. 19, no. 2, pp. 95–98, 2012.
- [7] S. J. Wang, J. Yang, M. F. Sun, X. J. Peng, M. M. Sun, and C. G. Zhou, "Sparse tensor discriminant color space for face verification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 6, pp. 876–888, 2012.
- [8] J. L. Minoi, C. E. Thomaz, and D. F. Gillies, "Tensor-based multivariate statistical discriminant methods for face applications," in *Proceedings of the International Conference on Statistics in Science, Business, and Engineering (ICSSBE '12)*, pp. 1–6, September 2012.
- [9] N. Tang, X. Gao, and X. Li, "Tensor subclass discriminant analysis for radar target classification," *Electronics Letters*, vol. 48, no. 8, pp. 455–456, 2012.
- [10] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognition*, vol. 44, no. 7, pp. 1540–1551, 2011.
- [11] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: multilinear principal component analysis of tensor objects," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.
- [12] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.
- [13] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000.
- [14] C. M. Johnson, "A survey of current research on online communities of practice," *Internet and Higher Education*, vol. 4, no. 1, pp. 45–60, 2001.
- [15] P. Hall, D. Marshall, and R. Martin, "Merging and splitting eigenspace models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 1042–1049, 2000.
- [16] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Incremental tensor analysis: theory and applications," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 3, article 11, 2008.
- [17] J. Wen, X. Gao, Y. Yuan, D. Tao, and J. Li, "Incremental tensor biased discriminant analysis: a new color-based visual tracking method," *Neurocomputing*, vol. 73, no. 4–6, pp. 827–839, 2010.
- [18] J.-G. Wang, E. Sung, and W.-Y. Yau, "Incremental two-dimensional linear discriminant analysis with applications to face recognition," *Journal of Network and Computer Applications*, vol. 33, no. 3, pp. 314–322, 2010.

- [19] X. Qiao, R. Xu, Y.-W. Chen, T. Igarashi, K. Nakao, and A. Kashimoto, "Generalized N-Dimensional Principal Component Analysis (GND-PCA) based statistical appearance modeling of facial images with multiple modes," *IPSJ Transactions on Computer Vision and Applications*, vol. 1, pp. 231–241, 2009.
- [20] H. Kong, X. Li, L. Wang, E. K. Teoh, J.-G. Wang, and R. Venkateswarlu, "Generalized 2D principal component analysis," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '05)*, vol. 1, pp. 108–113, August 2005.
- [21] D. Zhang and Z.-H. Zhou, "(2D)2 PCA: two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1–3, pp. 224–231, 2005.
- [22] J. Ye, "Generalized low rank approximations of matrices," *Machine Learning*, vol. 61, no. 1–3, pp. 167–191, 2005.
- [23] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.
- [24] J. Yang and J.-Y. Yang, "From image vector to matrix: a straightforward image projection technique-IMPCA vs. PCA," *Pattern Recognition*, vol. 35, no. 9, pp. 1997–1999, 2002.
- [25] J. Kwok and H. Zhao, "Incremental eigen decomposition," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN '03)*, pp. 270–273, Istanbul, Turkey, June 2003.
- [26] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.