Multilinear Principal Component Analysis of Tensor Objects

Tingyi Zhu

Department of Mathematics
Texas A&M University

July 12, 2012

Outline

Brief introduction

MPCA framework was proposed in 2008 by Lu, Plataniotis and Ventsanopoulos in the paper [1].

In this presentation, I will

- Review concepts in multilinear algebra
- Summarize the MPCA algorithm
- Present some experiment results of MPCA

Brief Introduction

Principal Component Analysis (PCA)

- Transform the original data set to a new set of variables
- Reduce the dimensionality
- Retain as much as possible the variation present in the original data set
- A well-known un-supervised linear technique for dimensionality reduction.

Brief Introduction

Principal Component Analysis (PCA)

- Transform the original data set to a new set of variables
- Reduce the dimensionality
- Retain as much as possible the variation present in the original data set
- A well-known un-supervised linear technique for dimensionality reduction.

Brief Introduction

Principal Component Analysis (PCA)

- Transform the original data set to a new set of variables
- Reduce the dimensionality
- Retain as much as possible the variation present in the original data set
- A well-known un-supervised linear technique for dimensionality reduction.

Brief Introduction

Principal Component Analysis (PCA)

- Transform the original data set to a new set of variables
- Reduce the dimensionality
- Retain as much as possible the variation present in the original data set
- A well-known un-supervised linear technique for dimensionality reduction.

Brief Introduction

Principal Component Analysis (PCA)

- Transform the original data set to a new set of variables
- Reduce the dimensionality
- Retain as much as possible the variation present in the original data set
- A well-known un-supervised linear technique for dimensionality reduction.

Brief Introduction

To apply PCA on tensor objects, it requires their reshaping into vectors, which results in high processing cost in terms of increased computational and memory demands.

For example, vectorizing a typical gait sequence of size $(120\times 80\times 20)$ results in a vector with dimensionality (192000×1).

Besides, reshaping breaks the natural structure and correlation in the original data.

Therefore, a dimensionality reduction algorithm operating directly on a tensor object rather than its vectorized version is desirable.

What's Tensor

- Tensors are a further extension of ideas we use to define vectors and matrix.
- The elements of a tensor are to be addressed by N indices, where N defines the order of the tensor object and each index defines one mode. Thus, vectors are first-order tensors (with $N = 1$) and matrices are second-order tensors (with $N = 2$). Tensors with $N > 2$ can be viewed as a generalization of vectors and matrices to higher order.

An Nth-order tensor is denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, which is addressed by N indices i_n, $n = 1, \ldots, N$, with each i_n addresses the n-mode of $\mathcal{A}$.

The Scalar product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$:

$$< \mathcal{A}, \mathcal{B} >= \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} \mathcal{A}(i_1, \ldots, i_N) \cdot \mathcal{B}(i_1, \ldots, i_N) \tag{1}$$

.

The Frobenius norm of $\mathcal{A}$:

$$||\mathcal{A}||_F = \sqrt{< \mathcal{A}, \mathcal{A} >}. \tag{2}$$

The "n-mode vector" of $\mathcal{A}$ are defined as the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying its index i_n while keeping all the other indices fixed. A rank-one tensor $\mathcal{A}$ equals to the outer product of N vectors:

$$\mathcal{A} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ \cdots \circ \mathbf{u}^{(N)}, \tag{3}$$

i.e.,

$$\mathcal{A}(i_1, \ldots, i_N) = \mathbf{u}^{(1)}(i_1) \cdot \mathbf{u}^{(2)}(i_2) \cdot \cdots \cdot \mathbf{u}^{(N)}(i_N) \tag{4}$$

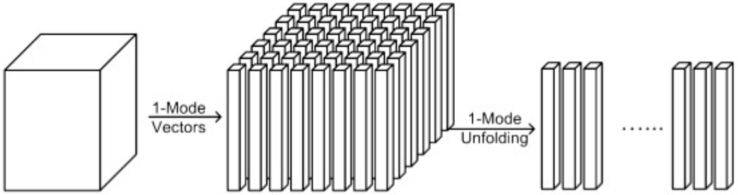
for all values of indices.

Unfolding $\mathcal{A}$ along the n-mode is denoted as:

$$\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N)}, \tag{5}$$

where the column vectors of $\mathbf{A}_(n)$ are the n-mode vectors of $\mathcal{A}$.

The figure below illustrates the 1-mode (column mode) unfolding of a third-order tensor.

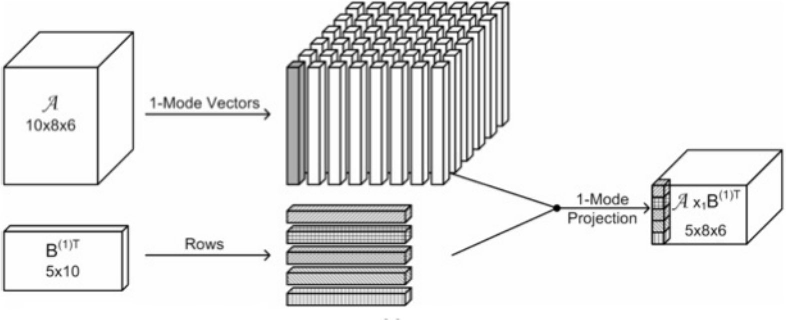


Multilinear projection

The n-mode product of a tensor $\mathcal{A}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$, denote as $\mathcal{A} \times_n \mathbf{U}$, is a tensor with entries:

$$(\mathcal{A} \times_n U)(i_1, \ldots, i_{n-1}, j_n, i_{n+1}, \ldots, i_N) = \sum_{i_n} \mathcal{A}(i_1, \ldots, i_N) \cdot \mathbf{U}(j_n, i_n). \quad (6)$$

Visual illustration:



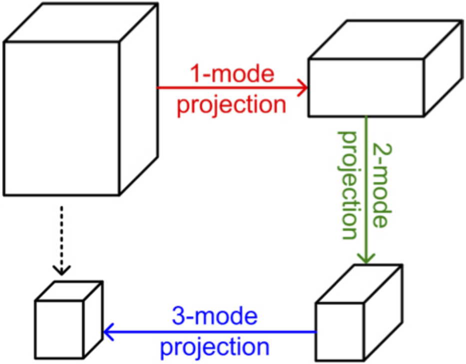
Multilinear projection

Based on the definitions above, a tensor can be projected to another tensor by N projection matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}$ as:

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \cdots \times_N \mathbf{U}^{(N)^T}. \tag{7}$$

Multilinear Projection

Visual illustration of tensor-to-tensor projection:



Following standard multilinear algebra, any tensor $\mathcal{A}$ can be expressed as the product

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \cdots \times_N \mathbf{U}^{(N)} \tag{8}$$

where $\mathcal{S} = \mathcal{A} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2^T)} \times \cdots \times_N \mathbf{U}^{(N)^T}$.

A matrix representation of this decomposition can be obtained by unfolding $\mathcal{A}$ and $\mathcal{S}$ as

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)} \cdot \mathbf{S}_{(n)} \cdot (\mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n+2)} \otimes \cdots \otimes \mathbf{U}^{(N)} \otimes \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \otimes \cdots \otimes \mathbf{U}^{(n+1)})^T \tag{9}$$

where $\otimes$ denotes the Kronecker product.

Multilinear Subspace Learning Problem

The problem of multilinear subspace learning based on the tensor-to-tensor projection can be mathematically defined as follows:

A set of M Nth-order tensor samples $\{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_M\}$ is available for training, where each sample $\mathcal{X}_m$ is an $I_1 \times I_2 \times \cdots \times I_N$ tensor in a tensor space $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.

Multilinear Subspace Learning Problem

The objective is to find a tensor-to-tensor projection $\{\tilde{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, \dots, N\}$ mapping from the original tensor space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \cdots \otimes \mathbb{R}^{I_N}$ into a tensor subspace $\mathbb{R}^{P_1} \otimes \mathbb{R}^{P_2} \cdots \otimes \mathbb{R}^{P_N}$ (with $P_n < I_n$, for $n = 1, \dots, N$):

$$\mathcal{Y}_m = \mathcal{X}_m \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} \cdots \times_N \tilde{\mathbf{U}}^{(N)^T}, \tag{10}$$

such that the projected samples (the extracted features) satisfy an optimality criterion, where the dimensionality of the projected space is much lower than the original tensor space.

Multilinear Subspace Learning Problem

The MPCA algorithm maximizes the following tensor-based scatter measure:

$$\Psi_{\mathcal{Y}} = \sum_{m=1}^{M} ||\mathcal{Y}_m - \bar{\mathcal{Y}}||_F^2, \tag{11}$$

named as the total tensor scatter, where

$$\bar{\mathcal{Y}} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{Y}_m \tag{12}$$

is the mean sample.

MPCA Algorithm

There is no known optimal solution which allows for the simultaneous optimization of the N projection matrices. Since the projection to an Nth-order tensor subspace consists of N projections to N vector subspaces, N optimization subproblems can be solved by finding the $\tilde{\mathbf{U}}^{(n)}$ that maximizes the scatter in the n-mode vector subspace. This is discussed in the following theorem.

MPCA Algorithm

Theorem

Let $\{\tilde{\mathbf{U}}(n), n = 1, ..., N\}$ be the solution of the optimization problem. Then, for given all the other projection matrices $\tilde{\mathbf{U}}(1), ..., \tilde{\mathbf{U}}(n-1), \tilde{\mathbf{U}}(n+1), ..., \tilde{\mathbf{U}}(N)$, the matrix $\tilde{\mathbf{U}}(n)$ consists of the P_n eigenvectors corresponding to the largest P_n eigenvalues of the matrix

$$\Phi^{(n)} = \sum_{m=1}^{M} (X_{m(n)} - \bar{X}_{(n)}) \cdot \tilde{\mathbf{U}}_{\Phi(n)} \cdot \tilde{\mathbf{U}}_{\Phi(n)}^T \cdot (X_{m(n)} - \bar{X}_{(n)})^T \quad (13)$$

where

$$\tilde{\mathbf{U}}_{\Phi(n)} = \tilde{\mathbf{U}}^{(n+1)} \otimes \tilde{U}^{(n+2)} \otimes \cdots \otimes \tilde{\mathbf{U}}^{(N)} \otimes \tilde{\mathbf{U}}^{(1)} \otimes \cdots \otimes \tilde{\mathbf{U}}^{(n-1)}. \quad (14)$$

MPCA Algorithm(Pseudocode)

Input: A set of tensor samples $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}, m = 1, ..., M\}$.

Output: Low-dimensional representations $\{\mathcal{Y}_m \in \mathbb{R}^{P_1 \times P_2 \times ... \times P_N}, m = 1, ..., M\}$ of the input tensor samples with maximum variation captured.

Algorithm:

Step 1 (Preprocessing): Center the input samples as $\{\tilde{\mathcal{X}}_m = \mathcal{X}_m - \bar{\mathcal{X}}, m = 1, ..., M\}$, where $\bar{\mathcal{X}} = \frac{1}{M}\sum_{m=1}^{M} \mathcal{X}_m$ is the sample mean.

Step 2 (Initialization): Calculate the eigen-decomposition of $\mathbf{\Phi}^{(n)*} = \sum_{m=1}^{M} \tilde{\mathbf{X}}_{m(n)} \cdot \tilde{\mathbf{X}}_{m(n)}^T$ and set $\tilde{\mathbf{U}}^{(n)}$ to consist of the eigenvectors corresponding to the most significant P_n eigenvalues, for $n = 1, ..., N$.

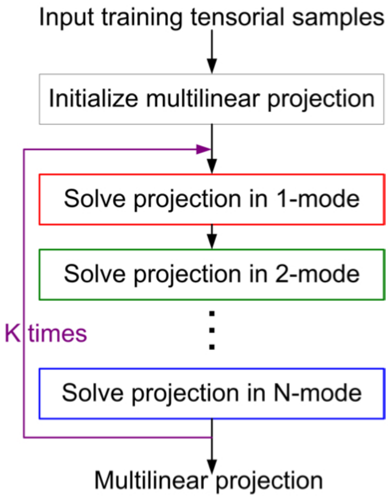
Step 3 (Local optimization):

- Calculate $\{\tilde{\mathcal{Y}}_m = \tilde{\mathcal{X}}_m \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} ... \times_N \tilde{\mathbf{U}}^{(N)^T}, m = 1, ..., M\}$.
- Calculate $\Psi_{\mathcal{Y}_0} = \sum_{m=1}^{M} \| \tilde{\mathcal{Y}}_m \|_F^2$ (the mean $\bar{\tilde{\mathcal{Y}}}$ is all zero since $\tilde{\mathcal{X}}_m$ is centered).
- For $k = 1 : K$
  - For $n = 1 : N$
    - Set the matrix $\tilde{\mathbf{U}}^{(n)}$ to consist of the P_n eigenvectors of the matrix $\mathbf{\Phi}^{(n)}$, as defined in (5), corresponding to the largest P_n eigenvalues.
  - Calculate $\{\tilde{\mathcal{Y}}_m, m = 1, ..., M\}$ and $\Psi_{\mathcal{Y}_k}$.
  - If $\Psi_{\mathcal{Y}_k} - \Psi_{\mathcal{Y}_{k-1}} < \eta$, break and go to Step 4.

Step 4 (Projection): The feature tensor after projection is obtained as $\{\mathcal{Y}_m = \mathcal{X}_m \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} ... \times_N \tilde{\mathbf{U}}^{(N)^T}, m = 1, ..., M\}$.

The computations of the projection matrices are interdependent, which implies there is no closed-form solution to the optimization problem. Therefore, an iterative procedure is utilized. The input tensors are centered first. With initializations through full projection truncation (FPT), the projection matrices are computed one by one with all the others fixed (local optimization). The local optimization procedure can be repeated until the result converges or a maximum number K of iterations is reached.

MPCA Algorithm(Algorithm flow)

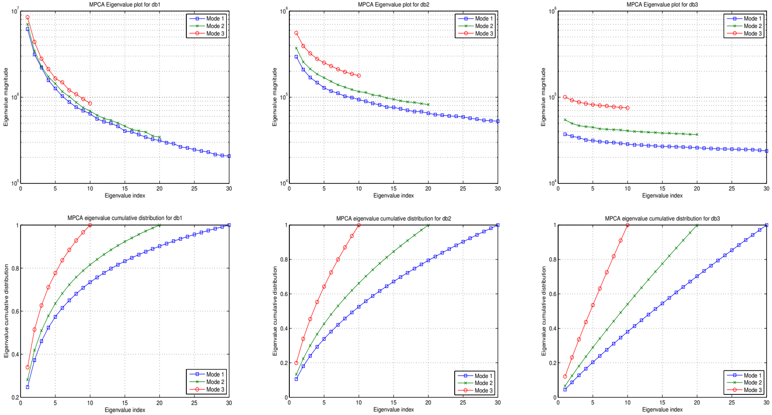


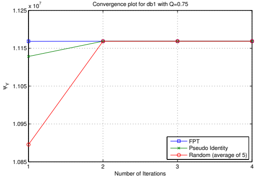
Synthetic Data Generation

100 third-order tensors $\mathcal{A}_m \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ are generated per set according to $\mathcal{A}_m = \mathcal{B}_m \times_1 \mathbf{C}^{(1)} \times_2 \mathbf{C}^{(2)} \times_3 \mathbf{C}^{(3)} + \mathcal{D}_m$ where

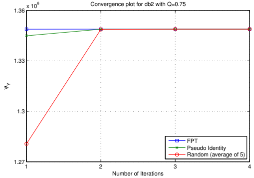
- All entries in $\mathcal{B}_m$ are drawn from a zero-mean unit-variance Gaussian distribution and are multiplied by $[(I_1 \cdot I_2 \cdot I_3)/(i_1 \cdot i_2 \cdot i_3)]^f$.
- The matrices $\mathbf{C}^{(n)}$ are orthogonal matrices obtained by applying SVD on random matrices with entries drawn from zero-mean, unit-variance Gaussian distribution.
- All entries of $\mathcal{D}_m$ are drawn from a zero-mean Gaussian distribution with variance 0.01.

Three synthetic data sets db1, db2, and db3 with $f = 1/2, 1/4, 1/16$ respectively, are created.

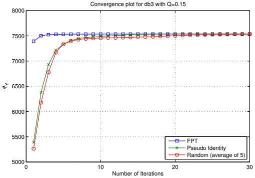

MPCA Eigenvalue plot for db1
MPCA Eigenvalue plot for db2
MPCA Eigenvalue plot for db3
Eigenvalue magnitude
Eigenvalue index
MPCA eigenvalue cumulative distribution for db1
MPCA eigenvalue cumulative distribution for db2
MPCA eigenvalue cumulative distribution for db3
Eigenvalue cumulative distribution
Mode 1
Mode 2
Mode 3



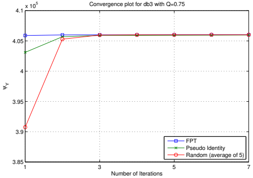
(a)



(b)



(c)



(d)

Figure: Convergence plot for different initializations. (a) Convergence plot for db1 with Q=0.75. (b) Convergence plot for db2 with Q=0.75. (c) Convergence plot for db3 with Q=0.15. (d) Convergence plot for db3 with Q=0.75.

Facial Image Dataset

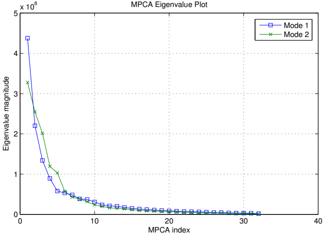
- 320 samples of face images
- Size of each image: 32×32
- Set Q=0.97
- Then, $P_1 = 23$ and $P_2 = 20$
- 91% of original variation captured

Facial Image Dataset

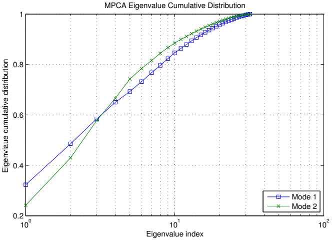
(a) (b)

Figure: Face image data of face recognition technology (FERET) database. (a) The first eight samples out of the total 320 samples. (b) The mean of face image samples.

Facial Image Dataset



(a)



(b)

Figure: Eigenvalue magnitudes and their cumulative distributions for the facial datase.

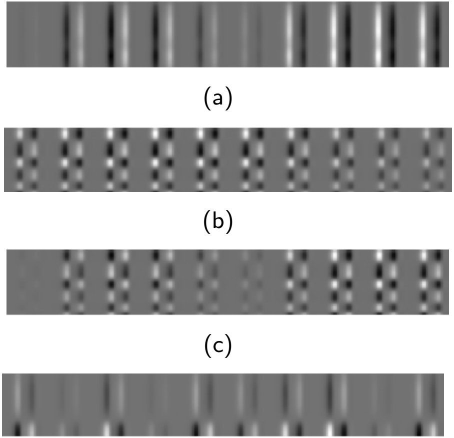
MPCA-Based Gait Recognition

- Gait gallery data: 731 samples
- Size of each sample: $I_1 \times I_2 \times I_3 = 32 \times 22 \times 10$
- I_1 and I_2 are frame size and I_3 is spatial size
- Dimensionality of the projected tensors: $P_1 \times P_2 \times P_3 = 26 \times 15 \times 10$.

(a)

(b)

Figure: (a) 1-mode unfolding of a gait silhouette sample. (b) 1-mode unfolding of the mean of the gait silhouette samples.



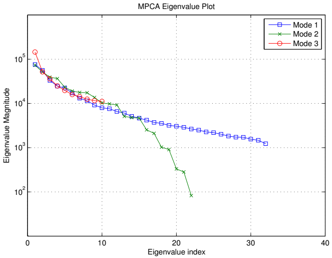
(a)

(b)

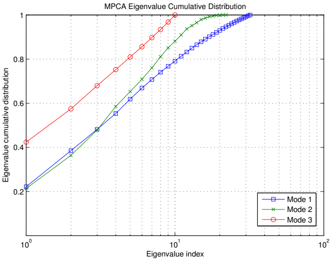
(c)

(d)

Figure: Representative ETGs: The first, second, third, 4th most discriminative ETGs.



(a)



(b)

Figure: Eigenvalue magnitudes and their cumulative distributions for the gallery set.

Summary

References

Haiping Lu and K. N. Plataniotis and A. N. Venetsanopoulos
A Survey of Multilinear Subspace Learning for Tensor Data.
Pattern Recognition, 44(7):15401551, 201.

H. Lu and K. N. Plataniotis and A. N. Venetsanopoulosi.
Multilinear Principal Component Analysis of Tensor Objects.
IEEE Trans. on Neural Networks, 19(1):18–39 2008.

L. D. Lathauwer, B. D. Moor, and J. Vandewalle
A multilinear singular value decomposition.
SIAM J. Matrix Anal. Appl, vol. 21, no. 4, pp. 1253-1278, 2000.

Data Sources:
http://www.dsp.utoronto.ca/haiping/index.php?page=code.

Tensor Toolbox:
http://csmr.ca.sandia.gov/tgkolda/TensorToolbox/.