Journal of the Chinese Statistical Association
Vol. 52, (2014) 24–43

AN INTRODUCTION TO MULTILINEAR PRINCIPAL COMPONENT ANALYSIS

Ting-Li Chen[a], Su-Yun Huang[a], Hung Hung[b], I-Ping Tu[a]

[a]Institute of Statistical Science, Academia Sinica

[b]Institute of Epidemiology & Preventive Medicine National, Taiwan University

ABSTRACT

Principal component analysis (PCA) is a simple and very popular method in statistical data analysis. It can be done by eigenvalue decomposition of covariance matrix. Traditional PCA deals with vector variables and each observation is represented in vector form. When observations are tensor objects, such as images, videos, EEG signals over a spatial domain or gene-gene interactions (as symmetric random matrices), traditional PCA first vectorizes these tensor objects and then proceeds with the eigenvalue decomposition of a large covariance matrix. This vectorized PCA for tensor data can be difficult and inefficient. The main reason is that the estimation process of PCA is unstable when the sample size is small compared to the dimension of the vectorized data. Multilinear principal component analysis (MPCA) is a modification of PCA. It preserves the natural tensor structure of observations in searching for principal components. The main advantage of preserving the tensor structure is the parsimonious usage of parameters in specifying the principal component subspaces, which mitigates the adverse influence of high-dimensionality, and hence, leads to efficiency gain in estimation and prediction. In this article, we provide a user-friendly introduction to the basic concept and technique for MPCA. One will see the rationale for the success of MPCA, from the statistical point of view and based on some real data applications.

Key words and phrases: Dimension reduction; high order singular value decomposition;

Kronecker product; multilinear principal component analysis; principal component analysis; singular value decomposition; tensor object.
JEL classification: C18.

1. Introduction: from SVD and PCA to HOSVD and MPCA

1.1. PCA and SVD

Principal component analysis (PCA) is a very basic and widely used statistical tool for data analysis (Jolliffe, 2002). Let $X = (x_1, \ldots, x_n)^\top$ be a data matrix of size $n \times p$, whose rows are iid copies of p-dimensional random vectors. For simplicity, we assume X has been centered so that it has zero column means, i.e., $\frac{1}{n} 1_n^\top X = (0, \ldots, 0) \in \mathbb{R}^{1\times p}$. PCA seeks an orthogonal transformation $\Gamma \in \mathbb{R}^{p\times p}$ to convert $X \in \mathbb{R}^{n\times p}$ possibly correlated column variables into $U = X\Gamma \in \mathbb{R}^{n\times p}$ linearly uncorrelated column variables. Conventionally, U are arranged column-wise in descending order by their within-column variance. Often, $\Gamma = [\gamma_1, \ldots, \gamma_p]$ is solved by eigenvalue decomposition of the sample covariance matrix,

$$\frac{1}{n} X^\top X = \Gamma \Lambda \Gamma^\top,$$

where Λ is a diagonal matrix with entries $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$ and $\Gamma^\top \Gamma = I_p$. The matrix of principal component loadings Γ can also be obtained by singular value decomposition (SVD) of the data matrix,

$$X = Z \Lambda^{1/2} \Gamma^\top, \tag{1}$$

where $Z \in \mathbb{R}^{n\times p}$ has uncorrelated and standardized columns, i.e., $\frac{1}{n} Z^\top Z = I_p$. PCA is probably the most popular and widely used (unsupervised) dimension reduction tool. Based on (1), X is approximated by leading eigen-components as

$$X \approx \widetilde{Z} \widetilde{\Lambda}^{1/2} \widetilde{\Gamma}^\top, \quad \text{where } \widetilde{Z} \in \mathbb{R}^{n\times \widetilde{p}}, \quad \widetilde{\Lambda} \in \mathbb{R}^{\widetilde{p}\times\widetilde{p}}, \quad \widetilde{\Gamma} \in \mathbb{R}^{p\times\widetilde{p}}, \quad \widetilde{p} \ll p.$$

In summary, when observations collected are p-vectors and the variable space of interest is naturally residing in $\mathbb{R}^p$, PCA seeks an orthonormal basis Γ for the variable

space $\mathbb{R}^p$. Observations represented with this new coordinate system $\{x_1^\top \Gamma, \ldots, x_n^\top \Gamma\}$ become uncorrelated. Moreover, a low-dimensional projection to the subspace by leading PCs can be used for dimension reduction and for obtaining a compressed data representation. What if data collected are naturally tensor- or array-structured, such as images (e.g., Figure 1) or videos?

Shall we vectorize each of the array data into a long vector and proceed as vector data? Vectorization-based approaches lead to data analysis in an extremely high dimensional space, which can be troublesome. In the next subsection, we will introduce multilinear analogues of PCA and SVD, which have been shown successful to deal with tensor objects in many real data applications (De Lathauwer et al., 2000a, 2000b; Lu et al., 2008; Hung et al., 2012; Hung and Wang, 2013; Chen et al., 2013; Hung, 2013; Liu, 2013). A good review article on high-order tensor decompositions can be found in Kolda and Bader (2009).

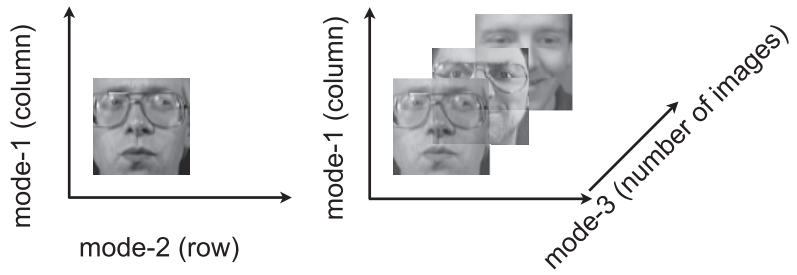

Figure 1: Olivetti faces dataset, http://www.cs.nyu.edu/~roweis/data.html.

1.2. High order SVD

When data collected are matrices, $\{X_i \in \mathbb{R}^{p\times q}\}_{i=1}^n$, such as images, traditional approaches vectorize each observation X_i column-wise into a long vector, denoted by $\text{vec}(X_i)$, and form a design matrix $\boldsymbol{X} = [\text{vec}(X_1), \ldots, \text{vec}(X_n)]^\top$, which is an $n \times pq$ matrix. This data matrix can be very thin and has very high dimensionality due to $n \ll pq$. This phenomenon of high dimensionality can get worse when data are higher order arrays. Methodology development for analyzing array data, known as tensor

methods in statistics, is one of important topics in contemporary statistics. Below we will introduce an extension of matrix SVD to an array high order SVD (HOSVD, De Lathauwer et al., 2000a). To assist the reader for a better understanding of HOSVD, we will use the *color tree* picture for visual illustration.

The color tree consists of three image slices in red, green and blue. Each of the RGB image slice is of size 3648×2736 pixels. That is, the color tree image is an order-3 tensor (or array) of size $3648 \times 2736 \times 3$ pixels. Refer to the RGB images shown in the left panel of Figure 2. The HOSVD of color tree consists of 3 sets orthonormal bases, one for the column mode of dimensionality 3648, one for the row mode of dimensionality 2736, and the other is for the color mode of dimensionality 3. Using color tree data tensor for illustration, HOSVD procedure goes as follows: (1) For each mode $k = 1, 2, 3$, the data tensor $\boldsymbol{\mathcal{X}}$ is flattened along this mode into an unfolding matrix denoted by $\boldsymbol{X}_{(k)}$. To avoid heavy mathematical notation and details, we use pictorial illustration for mode-k unfolding. See Figure 2 and Figure 3 for unfolding along the column mode, row mode and RGB mode. The mode-k unfolding re-arranges the tensor object $\boldsymbol{\mathcal{X}}$ into a matrix $\boldsymbol{X}_{(k)}$ of which each column is a mode-k vector of $\boldsymbol{\mathcal{X}}$. E.g., the column mode unfolding matrix $\boldsymbol{X}_{(1)}$ consists of all the columns of $\boldsymbol{\mathcal{X}}$; the row mode unfolding matrix $\boldsymbol{X}_{(2)}$ is a matrix whose columns are the rows of $\boldsymbol{\mathcal{X}}$; the RGB mode unfolding matrix $\boldsymbol{X}_{(3)}$ is a matrix whose columns consist of the RGB intensities of all pixels in this color image tensor $\boldsymbol{\mathcal{X}}$. Thus, the column space of the unfolding matrix $\boldsymbol{X}_{(k)}$ spans the same subspace as the mode-k subspace of $\boldsymbol{\mathcal{X}}$. (2) Next it performs a SVD on $\boldsymbol{X}_{(k)}$ to get the orthonormal basis (arranged in descending order by singular values) for each mode-k. This orthonormal basis is called the mode-k basis and is denoted by $\boldsymbol{A}_k$, $k = 1, 2, 3$.

(3) Assemble these mode-k bases by Kronecker product,[1] $\boldsymbol{A}_3 \otimes \boldsymbol{A}_2 \otimes \boldsymbol{A}_1$. Here $\boldsymbol{A}_k$ is a complete orthonormal basis for mode-k. However, if the aim is for dimension reduction, one can use leading columns in $\boldsymbol{A}_k$ to form partial basis. Thus, from now on, $\boldsymbol{A}_k$ can be a partial basis of much lower dimensionality than the original mode-k dimensionality.

The k-mode product of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$ by a matrix $\boldsymbol{M} \in \mathbb{R}^{J_k \times I_k}$, denoted by $\boldsymbol{\mathcal{X}} \times_k \boldsymbol{M}$, is an $(I_1 \times \cdots \times I_{k-1} \times J_k \times I_{k+1} \times \cdots \times I_K)$-tensor, of which the entries are given by

$$(\boldsymbol{\mathcal{X}} \times_k \boldsymbol{M})[i_1, \ldots, i_{k-1}, j_k, i_{k+1}, \ldots, i_K] = \sum_{i_k=1}^{I_k} \boldsymbol{\mathcal{X}}[i_1, \ldots, i_k, \ldots, i_K] M[j_k, i_k],$$

where $\boldsymbol{\mathcal{X}}[i_1, \ldots, i_k, \ldots, i_K]$ and $M[j_k, i_k]$ are entries of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{M}$, respectively. For instance, let $\boldsymbol{\mathcal{X}}$ be an order-3 tensor in $\mathbb{R}^{p \times q \times r}$, and let $\boldsymbol{A}$ be a matrix in $\mathbb{R}^{p \times \widetilde{p}}$. Then, $\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}^\top$ is a tensor in $\mathbb{R}^{\widetilde{p} \times q \times r}$, whose entries are given by

$$\left(\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}^\top\right)[i, j, k] = \sum_{\ell=1}^{p} \boldsymbol{\mathcal{X}}[\ell, j, k](\boldsymbol{A}^\top)[i, \ell] = \sum_{\ell=1}^{p} \boldsymbol{\mathcal{X}}[\ell, j, k] \boldsymbol{A}[\ell, i],$$

where $i = 1, \ldots, \widetilde{p},\ j = 1, \ldots, q$ and $k = 1, \ldots, r$.

With the extracted mode-wise leading singular vectors $(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3)$, the original data tensor $\boldsymbol{\mathcal{X}}$ can be projected as

$$\boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}_1^\top \times_2 \boldsymbol{A}_2^\top \times_3 \boldsymbol{A}_3^\top.$$

This low-dimensional tensor $\boldsymbol{\mathcal{U}}$ is called the core tensor, which is the multilinear projection of the original data tensor $\boldsymbol{\mathcal{X}}$ onto the subspace spanned by $\boldsymbol{A}_3 \otimes \boldsymbol{A}_2 \otimes \boldsymbol{A}_1$. Note that changing the order of k-mode products is equivalent to changing the order of projections, which will not affect the resultant core tensor $\boldsymbol{\mathcal{U}}$. This can be easily verified mathematically.

[1] The Kronecker product of two matrices is defined as

$$A \otimes B = \begin{bmatrix} a_{1,1}B & \cdots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \cdots & a_{m,n}B \end{bmatrix}, \text{where } A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix}.$$

We can also reconstruct an approximate tensor of $\boldsymbol{\mathcal{X}}$ from the core tensor $\boldsymbol{\mathcal{U}}$,

$$\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{U}} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \times_3 \boldsymbol{A}_3.$$

Figure 3 gives a complete illustration for HOSVD and its usage for dimension reduction. The right-most tree picture uses $\boldsymbol{A}_1$ of size 3648×18, $\boldsymbol{A}_2$ of size 2736×18, and $\boldsymbol{A}_3$ of size 3×2 for image reconstruction. In this color tree example, the core tensor $\boldsymbol{\mathcal{U}}$ has size $18 \times 18 \times 2$.

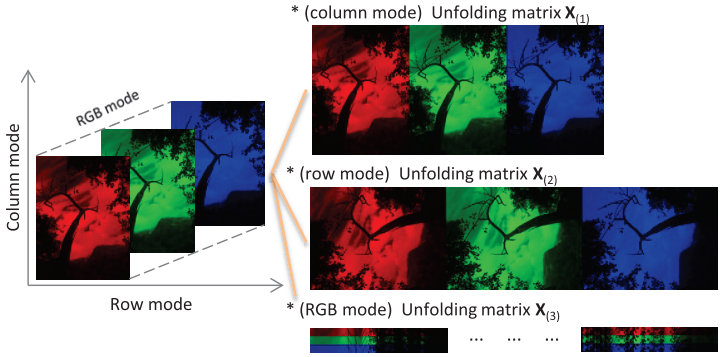

Figure 2: Unfolding the data tensor into matrices along each mode.

1.3. Multilinear PCA

HOSVD learns the bases for each mode separately. However, in many applications (e.g., images) there are spatial structures, which lead to some dependence between modes. Multilinear PCA introduced below will model the dependence between modes.

From now on we will use matrix (order-2 array) data for illustration. The extension to higher order is almost straightforward. Assume that we have a collection of matrix data $\{X_i \in \mathbb{R}^{p \times q}\}_{i=1}^n$. For a pre-specified dimensionality $(\widetilde{p}, \widetilde{q})$, also called rank-$(\widetilde{p}, \widetilde{q})$ with $\widetilde{p} \leq p$ and $\widetilde{q} \leq q$, MPCA finds the best approximation of $\{(X_i - \bar{X})\}_{i=1}^n$ by minimizing the following error distance, (see De Lathauwer et al., 2000b; Lu et al., 2008)

$$\underset{\boldsymbol{A}_1 \in \mathcal{O}_{p,\widetilde{p}}, \boldsymbol{A}_2 \in \mathcal{O}_{q,\widetilde{q}}}{\operatorname{argmin}} \sum_{i=1}^n \|(X_i - \bar{X}) - \boldsymbol{A}_1 \boldsymbol{A}_1^\top (X_i - \bar{X}) \boldsymbol{A}_2 \boldsymbol{A}_2^\top\|_F^2,$$

where $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, $\|\cdot\|_F$ is the matrix Frobenius norm[2] and $\mathcal{O}_{p,\tilde{p}} = \{\boldsymbol{M} \in \mathbb{R}^{p\times\tilde{p}} : \boldsymbol{M}^\top\boldsymbol{M} = I_{\tilde{p}}\}$. It is not difficult to show that the above minimization problem is equivalent to the following maximization problem,

$$\underset{\boldsymbol{A}_1\in\mathcal{O}_{p,\tilde{p}},\boldsymbol{A}_2\in\mathcal{O}_{q,\tilde{q}}}{\operatorname{argmax}} \sum_{i=1}^{n} \|\boldsymbol{A}_1^\top(X_i-\bar{X})\boldsymbol{A}_2\|_F^2.$$

At the solution, $\boldsymbol{A}_1^\top(X_i-\bar{X})\boldsymbol{A}_2$ is the extracted low-dimensional core tensor of $(X_i-\bar{X})$, and $(X_i-\bar{X})$ is approximated by $\boldsymbol{A}_1\boldsymbol{A}_1^\top(X_i-\bar{X})\boldsymbol{A}_2\boldsymbol{A}_2^\top$.

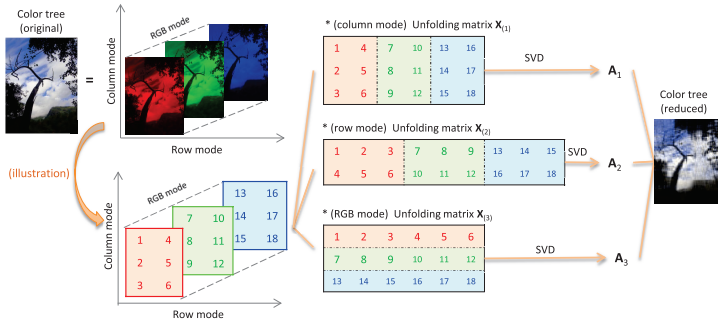

Figure 3: A pictorial illustration for HOSVD for the color tree tensor. The right-most image is reconstructed using 18 singular vectors from column mode, 18 singular vectors from row mode, and 2 singular vectors from RGB mode.

2. Algorithm of MPCA

2.1. Maximization by iterative alternating eigenvalue problems

Solving the maximization problem

$$\underset{\boldsymbol{A}_1\in\mathcal{O}_{p,\tilde{p}},\boldsymbol{A}_2\in\mathcal{O}_{q,\tilde{q}}}{\operatorname{argmax}} \sum_{i=1}^{n} \|\boldsymbol{A}_1^\top(X_i-\bar{X})\boldsymbol{A}_2\|_F^2$$

directly for $(\boldsymbol{A}_1, \boldsymbol{A}_2)$ is difficult. The term to be maximized can be rewritten as

$$\sum_{i=1}^{n} \|\boldsymbol{A}_1^\top(X_i-\bar{X})\boldsymbol{A}_2\|_F^2 = \operatorname{tr}\left[\boldsymbol{A}_1^\top\sum_{i=1}^{n}\left\{(X_i-\bar{X})\boldsymbol{A}_2\boldsymbol{A}_2^\top(X_i-\bar{X})^\top\right\}\boldsymbol{A}_1\right].$$

[2]The Frobenius norm of a matrix A is defined as

$$\|A\|_F = (\sum_i\sum_j |a[i,j]|^2)^{1/2} = \{\operatorname{tr}(AA^\top)\}^{1/2}.$$

Algorithm 1 Algorithm for MPCA (GLRAM, Ye, 2005)

INPUT: Data matrices $\{X_i \in \mathbb{R}^{p\times q}\}_{i=1}^n$; Dimensionality $(\widetilde{p}, \widetilde{q})$; Stopping parameter δ_0.

OUTPUT: $\widehat{\boldsymbol{A}}_1 \in \mathcal{O}_{p,\widetilde{p}},\ \widehat{\boldsymbol{A}}_2 \in \mathcal{O}_{q,\widetilde{q}}$.

An initial matrix $\boldsymbol{A}_1^{(0)} \in \mathcal{O}_{p,\widetilde{p}}$ is selected at random or obtained by HOSVD.

(HOSVD-initial is recommended.) $t = 0$. $\delta = 2\delta_0$.

while $\delta > \delta_0$ **do**

- $\boldsymbol{A}_2^{(t+1)} = (\boldsymbol{a}_{21}, \boldsymbol{a}_{22}, \ldots, \boldsymbol{a}_{2\widetilde{q}})$, where $\boldsymbol{a}_{2i}$'s are the leading $\widetilde{q}$ eigenvectors of $\sum_{i=1}^n \left\{(X_i - \bar{X})^\top \boldsymbol{A}_1^{(t)} \boldsymbol{A}_1^{(t)^\top} (X_i - \bar{X})\right\}$.
- $\boldsymbol{A}_1^{(t+1)} = (\boldsymbol{a}_{11}, \boldsymbol{a}_{12}, \ldots, \boldsymbol{a}_{1\widetilde{p}})$, where $\boldsymbol{a}_{1i}$'s are the leading $\widetilde{p}$ eigenvectors of $\sum_{i=1}^n \left\{(X_i - \bar{X}) \boldsymbol{A}_2^{(t+1)} \boldsymbol{A}_2^{(t+1)^\top} (X_i - \bar{X})^\top\right\}$.
- $\delta = \frac{1}{n}\left(\sum_{i=1}^n \|\boldsymbol{A}_1^{(t+1)^\top}(X_i - \bar{X})\boldsymbol{A}_2^{(t+1)}\|_F^2 - \sum_{i=1}^n \|\boldsymbol{A}_1^{(t)^\top}(X_i - \bar{X})\boldsymbol{A}_2^{(t)}\|_F^2\right)$.
- $t = t + 1$.

end while

return $\widehat{\boldsymbol{A}}_1 = \boldsymbol{A}_1^{(t)}$ and $\widehat{\boldsymbol{A}}_2 = \boldsymbol{A}_2^{(t)}$.

Suppose that $(\widehat{\boldsymbol{A}}_1, \widehat{\boldsymbol{A}}_2)$ is the maximizer. Then

$$\widehat{\boldsymbol{A}}_1 = \underset{\boldsymbol{A}_1 \in \mathcal{O}_{p,\widetilde{p}}}{\operatorname{argmax}}\ \operatorname{tr}\left[\boldsymbol{A}_1^\top \sum_{i=1}^n \left\{(X_i - \bar{X})\widehat{\boldsymbol{A}}_2 \widehat{\boldsymbol{A}}_2^\top (X_i - \bar{X})^\top\right\} \boldsymbol{A}_1\right].$$

Let $\widehat{\boldsymbol{A}}_1 = (\hat{\boldsymbol{a}}_{11}, \hat{\boldsymbol{a}}_{12}, \ldots, \hat{\boldsymbol{a}}_{1\widetilde{p}})$. The above term is maximized when $\hat{\boldsymbol{a}}_{1i}$'s are the leading $\widetilde{p}$ eigenvectors of $\sum_{i=1}^n \left\{(X_i - \bar{X})\widehat{\boldsymbol{A}}_2 \widehat{\boldsymbol{A}}_2^\top (X_i - \bar{X})^\top\right\}$. Similarly, $\widehat{\boldsymbol{A}}_2 = (\hat{\boldsymbol{a}}_{21}, \hat{\boldsymbol{a}}_{22}, \ldots, \hat{\boldsymbol{a}}_{2\widetilde{q}})$, where $\hat{\boldsymbol{a}}_{2i}$'s are the leading $\widetilde{q}$ eigenvectors of $\sum_{i=1}^n \left\{(X_i - \bar{X})^\top \widehat{\boldsymbol{A}}_1 \widehat{\boldsymbol{A}}_1^\top (X_i - \bar{X})\right\}$.

Though we can not exactly solve for $(\widehat{\boldsymbol{A}}_1, \widehat{\boldsymbol{A}}_2)$, we can iteratively refine the approximations of $(\widehat{\boldsymbol{A}}_1, \widehat{\boldsymbol{A}}_2)$ based on their properties discussed above. The algorithm is presented in **Algorithm 1**. This algorithm is called GLRAM and is due to Ye (2005).

2.2. Convergence of Algorithm 1

Since

$$\boldsymbol{A}_2^{(t+1)} = \underset{\boldsymbol{A}_2 \in \mathcal{O}_{q,\widetilde{q}}}{\operatorname{argmax}} \sum_{i=1}^{n} \| {\boldsymbol{A}_1^{(t)}}^\top (X_i - \bar{X}) \boldsymbol{A}_2 \|_F^2$$

and

$$\boldsymbol{A}_1^{(t+1)} = \underset{\boldsymbol{A}_1 \in \mathcal{O}_{p,\widetilde{p}}}{\operatorname{argmax}} \sum_{i=1}^{n} \| {\boldsymbol{A}_1}^\top (X_i - \bar{X}) \boldsymbol{A}_2^{(t+1)} \|_F^2,$$

$\sum_{i=1}^{n} \| {\boldsymbol{A}_1^{(t)}}^\top (X_i - \bar{X}) \boldsymbol{A}_2^{(t)} \|_F^2$ is an increasing function with respect to t. Since $\boldsymbol{A}_1 \in \mathcal{O}_{p,\widetilde{p}}$ and $\boldsymbol{A}_2 \in \mathcal{O}_{q,\widetilde{q}}$, $\sum_{i=1}^{n} \| {\boldsymbol{A}_1^{(t)}}^\top (X_i - \bar{X}) \boldsymbol{A}_2^{(t)} \|_F^2$ is bounded above and hence $(\boldsymbol{A}_1^{(t)}, \boldsymbol{A}_2^{(t)})$ converges. The convergence of $(\boldsymbol{A}_1^{(t)}, \boldsymbol{A}_2^{(t)})$ only ensures that $\lim_{t\to\infty}(\boldsymbol{A}_1^{(t)}, \boldsymbol{A}_2^{(t)})$ is a local maximum point, which might not be the global maximum point. For example, let $p = q = 2$, $\widetilde{p} = \widetilde{q} = 1$, $n = 2$, and

$$X_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } X_2 = \begin{pmatrix} -2 & 0 \\ 0 & -1 \end{pmatrix}.$$

It is easy to check that $(\boldsymbol{A}_1 = (1,0)^\top, \boldsymbol{A}_2 = (1,0)^\top)$ is the global maximum and $(\boldsymbol{A}_1 = (0,1)^\top, \boldsymbol{A}_2 = (0,1)^\top)$ is a local maximum. If we start with $\boldsymbol{A}_1^{(0)} = (0,1)^\top$, we will have $\boldsymbol{A}_2^{(1)} = (0,1)^\top$ and $\boldsymbol{A}_1^{(1)} = (0,1)^\top$. The proposed algorithm will stop at this local maximum. However, if we start with an initial value that is close to the global maximum, a convergence to the maximum can be expected. The solution of HOSVD is the Kronecker product of the k-mode bases. It is marginally optimal in each mode, and hence it is likely to close to the global maximum. Along with the advantage of its straightforward computation, the solution of HOSVD is recommended over a random initial to be the starting assignment.

3. Statistical model of MPCA

We start with the conventional PCA model for the vectorized observation $\operatorname{vec}(X_i)$, and extend it to the MPCA model for matrices $\{X_i\}_{i=1}^n$. We will see a connection between PCA and MPCA models, which sheds some light on the success of MPCA, especially in the case of small sample size.

3.1. Model specification

Let $\mu = E[X_i]$ be the mean matrix and $m = pq$. The conventional probabilistic PCA model assumes that the random vector $\text{vec}(X_i)$ has the structure

$$\text{vec}(X_i) = \text{vec}(\mu) + \boldsymbol{\Gamma}\boldsymbol{\nu}_i + \text{vec}(\boldsymbol{\varepsilon}_i), \tag{2}$$

where $\boldsymbol{\Gamma} \in \mathcal{O}_{m,r}$ with $r \leq m$ is a (partial) basis for $\mathbb{R}^m$, $\boldsymbol{\nu}_i \in \mathbb{R}^r$ is the random coordinate vector (called PCA scores) of X_i, and $\boldsymbol{\varepsilon}_i \in \mathbb{R}^{p\times q}$ is a random matrix of errors. In the probabilistic PCA model, it is assumed $E[\boldsymbol{\nu}_i] = 0$, $\text{Cov}(\boldsymbol{\nu}_i)$ is a strictly positive definite diagonal matrix, $E[\boldsymbol{\varepsilon}_i] = 0$, $\text{Cov}(\boldsymbol{\varepsilon}_i) = \sigma^2\boldsymbol{I}_m$, and $\boldsymbol{\varepsilon}_i$ is stochastically independent of $\boldsymbol{\nu}_i$. If without considering the error $\boldsymbol{\varepsilon}_i$, the data cloud $\{\text{vec}(X_i - \mu)\}_{i=1}^n$ lies in the r-dimensional subspace $\text{span}(\boldsymbol{\Gamma})$ of $\mathbb{R}^m$. In this case, $\boldsymbol{\nu}_i$ contains all the information regarding $\text{vec}(X_i - \mu)$, and should be the main variable subspace of interest. Typically, $\boldsymbol{\Gamma}$ is estimated by solving the eigenvalue problem for the sample covariance matrix of $\{\text{vec}(X_i)\}_{i=1}^n$. This can be seen from the eigenvalue decomposition of the population covariance matrix,

$$\boldsymbol{\Sigma} = \text{Cov}\{\text{vec}(X_i)\} = \boldsymbol{\Gamma}\text{Cov}(\boldsymbol{\nu}_i)\boldsymbol{\Gamma}^\top + \sigma^2\boldsymbol{I}_m, \tag{3}$$

which implies that $\boldsymbol{\Gamma}$ consists of the leading r eigenvectors of $\boldsymbol{\Sigma}$.

While the PCA model assumes that $\text{vec}(X_i - \mu)$ ideally lies in $\text{span}(\boldsymbol{\Gamma})$, the MPCA model preserves the matrix structure of X_i and assumes that the columns and rows of $(X_i - \mu)$ lie in the subspaces $\text{span}(\boldsymbol{A}_1) \subseteq \mathbb{R}^p$ and $\text{span}(\boldsymbol{A}_2) \subseteq \mathbb{R}^q$, respectively, where $\boldsymbol{A}_1 \in \mathcal{O}_{p,p_0}$ and $\boldsymbol{A}_2 \in \mathcal{O}_{q,q_0}$ with $(p_0, q_0) \leq (p, q)$. In particular, the MPCA model assumes

$$X_i = \mu + \boldsymbol{A}_1 U_i \boldsymbol{A}_2^\top + \boldsymbol{\varepsilon}_i, \tag{4}$$

where $U_i \in \mathbb{R}^{p_0\times q_0}$ is the random coordinate matrix with $E[U_i] = 0$. One can see from (4) how MPCA utilizes the matrix structure by retaining both the column basis $\boldsymbol{A}_1$ and the row basis $\boldsymbol{A}_2$ for data representation. Under model (4), it is U_i, if errors are ignored, that contains all the information about X_i, and U_i can be naturally constructed by $\boldsymbol{A}_1^\top(X_i - \mu)\boldsymbol{A}_2$, provided we have good estimates of $(\boldsymbol{A}_1, \boldsymbol{A}_2)$. See Section 2 for the

estimation criterion for $(\boldsymbol{A}_1, \boldsymbol{A}_2)$ and its implementation algorithm. Note that (p_0, q_0) is the true dimensionality of the MPCA model (4). Almost always (p_0, q_0) is not known and an estimate or an approximate value $(\widetilde{p}, \widetilde{q})$ is used in practice. See Hung et al. (2012) for a selection criterion of the dimensionality, which is done by hypothesis test based on asymptotic normality derived in that paper. Consistency and asymptotic normality of MPCA are discussed therein. We will further discuss the influence when using an approximate $(\widetilde{p}, \widetilde{q})$ to implement MPCA in Section 3.2.

3.2. Characteristics of MPCA

Since MPCA preserves the matrix structure of X_i (i.e., it involves more constraints when searching for the dimension reduction subspace), the adopted MPCA subspace cannot be smaller than PCA. To see this, we equivalently express model (4) as

$$\mathrm{vec}(X_i) = \mathrm{vec}(\mu) + (\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)\mathrm{vec}(U_i) + \mathrm{vec}(\boldsymbol{\varepsilon}_i). \tag{5}$$

Thus, MPCA model (4) is equivalent to imposing a Kronecker product structure to the dimension reduction subspace $\mathrm{span}(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)$ for $\mathrm{vec}(X_i)$. Unlike model (2), however, $\mathrm{Cov}\{\mathrm{vec}(U_i)\}$ of the random coordinate $\mathrm{vec}(U_i)$ may not be of full-rank nor diagonal. Consequently, there must exist a matrix $G \in \mathbb{R}^{m_0 \times r}$, $r \le m_0 = p_0 q_0$, and a random vector $\boldsymbol{\nu}_i \in \mathbb{R}^r$ with $\mathrm{Cov}(\boldsymbol{\nu}_i)$ being diagonal and strictly positive definite such that $\mathrm{vec}(U_i) = G\boldsymbol{\nu}_i$. Substituting this expression of $\mathrm{vec}(U_i)$ into (5) gives

$$\mathrm{vec}(X_i) = \mathrm{vec}(\mu) + \{(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)G\}\boldsymbol{\nu}_i + \mathrm{vec}(\boldsymbol{\varepsilon}_i), \tag{6}$$

which is nothing but a structured PCA model (2) with $\boldsymbol{\Gamma} = (\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)G$. It is now clear that MPCA uses a larger subspace $\mathrm{span}(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)$ for dimension reduction, since $\mathrm{span}(\boldsymbol{\Gamma}) = \mathrm{span}\{(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)G\}$ must be a subspace of $\mathrm{span}(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)$. That is, by ignoring the matrix structure of X_i, it is possible for further dimension reduction for $\mathrm{vec}(X_i)$, due to the effect of G in (6). Although MPCA cannot achieve the minimal dimension reduction, by preserving the matrix structure of observations, MPCA does achieve a more efficient estimation for the dimension reduction subspace as described below.

The minimal dimension reduction subspace for $\text{vec}(X_i)$ is $\text{span}(\boldsymbol{\Gamma})$ with dimensionality r, and PCA is able to target it directly. On the other hand, (5)-(6) imply that MPCA targets $\text{span}(\boldsymbol{\Gamma})$ indirectly by targeting the *Kronecker envelope* of $\text{span}(\boldsymbol{\Gamma})$, i.e., $\text{span}(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)$ with dimensionality $m_0 = p_0 q_0$. The idea of Kronecker envelope was first introduced in Li et al. (2010). They showed that for any $\boldsymbol{\Gamma} \in \mathcal{O}_{m,r}$ and predetermined dimension-folding (p, q) such that $pq = m$, there must exist $\boldsymbol{A}_1 \in \mathcal{O}_{p\times p_0}$ and $\boldsymbol{A}_2 \in \mathcal{O}_{q\times q_0}$ with minimum dimensions (p_0, q_0) such that $\text{span}(\boldsymbol{\Gamma}) \subseteq \text{span}(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)$. It is this Kronecker envelope that makes MPCA possible to better parsimoniously parameterize the target dimension reduction subspace $\text{span}(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)$ than PCA does for $\text{span}(\boldsymbol{\Gamma})$, even $\text{span}(\boldsymbol{\Gamma}) \subseteq \text{span}(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)$. In particular, the number of required parameters for $\text{span}(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)$ is of the order $O(pp_0 + qq_0)$, while it is $O(pqr)$ for $\text{span}(\boldsymbol{\Gamma})$. As a result, MPCA adopts a larger but tensor-structured subspace for dimension reduction, and this tensor structure greatly reduces the number of parameters required to specify such a subspace. Thus, it leads to efficiency gain in estimation and prediction. This is a common phenomenon in trading bias for variance.

***Remark* 1**. In Hung et al. (2012), the authors named r the *effective dimension*, which is the minimal dimension required by PCA for representing $\text{vec}(X_i)$, and named (p_0, q_0) the *Kronecker dimension*, which is the minimal dimension required by MPCA for representing X_i. From (6), we have $r \leq m_0 \leq m$.

In calculating PCA, there is no need to pre-specify the effective dimension r. In calculating MPCA one has to specify a dimensionality pair $(\widetilde{p}, \widetilde{q})$. Suppose the corresponding estimates are $(\widehat{\boldsymbol{A}}_1, \widehat{\boldsymbol{A}}_2)$. Although the Kronecker envelope of $\text{span}(\boldsymbol{\Gamma})$ must exist, there is no guarantee to prevent it from the trivial case, $(p_0, q_0) = (p, q)$, where the Kronecker envelope is the entire space $\mathbb{R}^m$ to encapsulate $\text{span}(\boldsymbol{\Gamma})$. We usually have $n \ll pq$, when dealing with tensor objects. This small n phenomenon leads us to use $(\widetilde{p}, \widetilde{q}) < (p_0, q_0)$. In this situation, the estimated MPCA subspace $\text{span}(\widehat{\boldsymbol{A}}_2 \otimes \widehat{\boldsymbol{A}}_1)$ might not contain the true $\text{span}(\boldsymbol{\Gamma})$. As a result, MPCA under $(\widetilde{p}, \widetilde{q}) < (p_0, q_0)$ is no longer an exact true model, but an approximate one. Comparing with PCA targeting $\text{span}(\boldsymbol{\Gamma})$ directly (using parameters in the order $O(pqr)$), there is a trade-off between approximation bias and estimation efficiency in using MPCA. Again, the efficiency gain of MPCA mainly comes from the parsimonious usage of parameters (in the order $O(p\widetilde{p} + q\widetilde{q})$) to

approximate span($\mathbf{\Gamma}$). It is our empirical experience that this efficiency gain usually dominates the loss from approximation bias, even when there exists no clear Kronecker product structure to represent vec(X_i). That is, MPCA can easily outperform PCA especially when the sample size is small. With limited sample size, it is preferable to more efficiently estimate an approximate tensor-structured subspace span($\widehat{\boldsymbol{A}}_2 \otimes \widehat{\boldsymbol{A}}_1$), instead of unstably estimating the optimal subspace span($\mathbf{\Gamma}$). See Figure 4 for a conceptual display for the dimension reduction subspaces of MPCA and PCA.

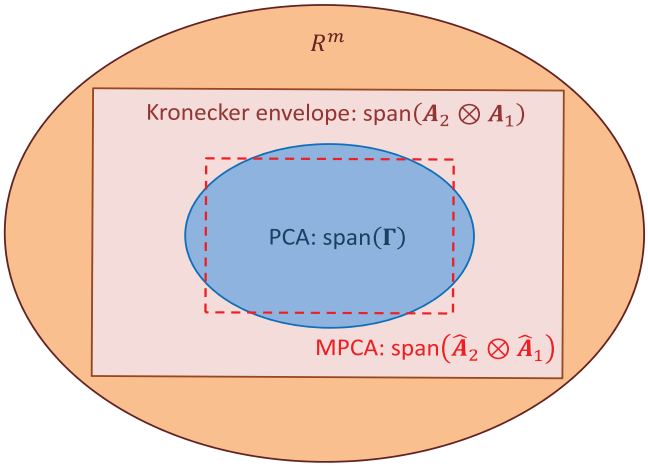

Figure 4: The conceptual display for the dimension reduction subspaces of MPCA and PCA. PCA uses the minimal dimension reduction subspace span($\mathbf{\Gamma}$) for vec(X_i) $\in \mathbb{R}^m$. MPCA under the true dimension (p_0, q_0) aims to use a larger Kronecker envelope span($\boldsymbol{A}_2 \otimes \boldsymbol{A}_1$) to encapsulate span($\mathbf{\Gamma}$). In empirical implementation, it can be the case that MPCA has used $(\widetilde{p}, \widetilde{q}) < (p_0, q_0)$ leading to dimension reduction subspace span($\widehat{\boldsymbol{A}}_2 \otimes \widehat{\boldsymbol{A}}_1$) (the dashed rectangle) to approximate span($\mathbf{\Gamma}$), with the benefit of using fewer parameters than PCA.

4. Applications

When the data are a set of two or higher order tensors (or arrays), like a gray or RGB image set, the traditional dimension reduction approach is to vectorize each array as a column vector, and combine them as a data matrix, and then apply PCA on it. Here, we call this approach the vectorized PCA. Employing vectorized PCA

has the convenient advantage that theoretical and application properties of PCA have been well studied and understood. However, vectorized PCA often creates a huge number of parameters and leads to the situation of large p and small n such that the behavior of the estimates (i.e., eigenvectors and eigenvalues) become unpredictable. MPCA avoids this large p and small n problem. Borrowing the model framework of vectorized PCA, MPCA imposes a tensor structure to the model, i.e., the Kronecker envelope structure to encapsulate the effective eigen-subspace, as discussed in Section 3.2. Through this modeling, one may pay price for the bias, but save a lot of variation on top of computation simplicity. For example, as to a set of $p \times q$ data matrix, vectorized PCA needs to construct the $pq \times pq$ covariance matrix, but MPCA alternatively deals with $p \times p$ and $q \times q$ mode-wise covariance matrices.

The authors of Hung et al. (2012) use the Olivetti faces data set to demonstrate how MPCA is applied to do the dimension reduction. They randomly sampled 100 images out of the whole set (400 images) as the training set and let the rest be the test set. Figure 5 shows the reconstruction performance of MPCA and vectorized PCA on 6 random test images. It is quite clear from Figure 5 that MPCA has better generalization ability than PCA in test image reconstruction.

Figure 5: The reconstruction of 6 random test images; (top) original; (middle) PCA reconstruction; (bottom) MPCA reconstruction.

If we define the projection matrix $P_{\boldsymbol{A}_i} = \boldsymbol{A}_i \boldsymbol{A}_i^\top$, the reconstruction for a test image X_{tst} can be written as $P_{\boldsymbol{A}_1} X_{tst} P_{\boldsymbol{A}_2}$ or $\boldsymbol{A}_1 U_{tst} \boldsymbol{A}_2^\top$, where $U_{tst} = \boldsymbol{A}_1^\top X_{tst} \boldsymbol{A}_2$ is the score matrix of X_{tst} projected to the column space $\boldsymbol{A}_1$ and the row space $\boldsymbol{A}_2$. $\boldsymbol{A}_1 U_{tst} \boldsymbol{A}_2^\top$ can be further decomposed as

$$\boldsymbol{A}_1 U_{tst} \boldsymbol{A}_2^\top = \sum_{i=1}^{\widetilde{p}} \sum_{j=1}^{\widetilde{q}} U_{tst}[i,j] \boldsymbol{a}_{1i} \boldsymbol{a}_{2j}^\top, \tag{7}$$

where $U_{tst}[i,j]$ is the $(i,j)^{th}$ matrix element of U_{tst}, $\boldsymbol{a}_{1i}$ is the i^{th} column of $\boldsymbol{A}_1$, and $\boldsymbol{a}_{2j}$ is the j^{th} column of $\boldsymbol{A}_2$. Equation (7) gives a description that the reconstruction matrix is equal to the sum of the linear combination of the basis matrix $\boldsymbol{a}_{1i} \boldsymbol{a}_{2j}^\top$ with the projection score $U_{tst}[i,j]$. In Figure 6, we show the first 10×10 bases of one training set in the Olivetti face images. Looking at the basis images in Figure 6, one may concern that all the basis images contain various sizes of blocks, which are composed of vertical and horizontal lines. The line structure is a result of the Kronecker product of a column vector and a row vector.

One intuitive question is how the blocks can compose a face. It may be helpful to look at the SVD basis for image compression. Given a $p \times q$ image matrix $\boldsymbol{X}$, SVD does the decomposition as

$$\boldsymbol{X} = \boldsymbol{A}_1 \boldsymbol{D} \boldsymbol{A}_2^\top = \sum_{i=1}^{\min(p,q)} d_i \boldsymbol{a}_{1i} \boldsymbol{a}_{2i}^\top \approx \sum_{i=1}^{r} d_i \boldsymbol{a}_{1i} \boldsymbol{a}_{2i}^\top, \tag{8}$$

where $\boldsymbol{D}$ is the diagonal matrix with elements $d_1 \geq d_2 \geq \cdots \geq d_{\min(p,q)}$, and r is the reduced rank. SVD decomposes $\boldsymbol{X}$ to many rank-one matrix with dimension $p \times q$, where each is a form of $\boldsymbol{a}_{1i} \boldsymbol{a}_{2i}^\top$. This is similar to the basis matrix of MPCA except that SVD only includes the diagonal indices. Here, we use Miller's drawing "Gleaner" as a demonstration. The drawing is documented as a 196×257 matrix $\boldsymbol{X}$ as shown in Figure 7. We show three compressed images from (8) whose reduced rank r's are 25, 50, and 100. Pairwise comparisons on these three images suggest that the 25 leading bases catch the main structure, and the next 25 bases polish the surface, and the further 50 bases take care of the curve.

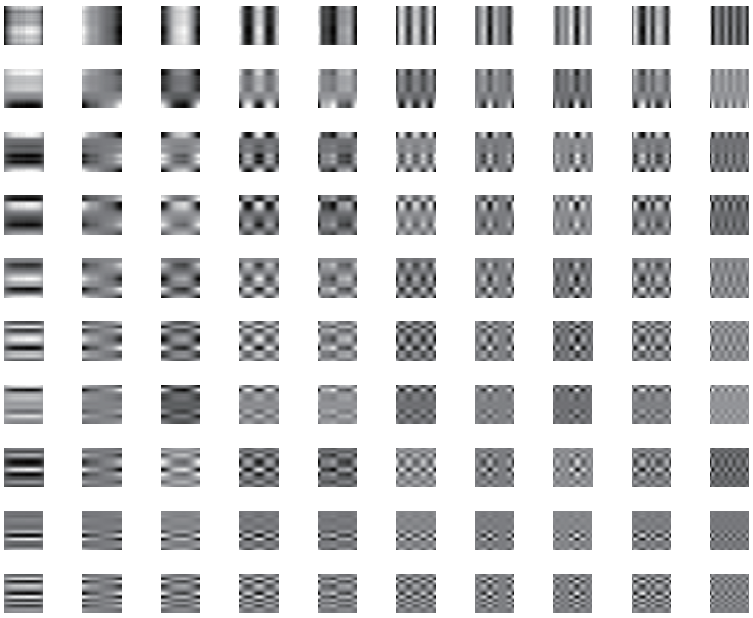

Figure 6: The first 10 × 10 bases of MPCA on a training set of Olivetti faces data.

Figure 7: The data matrix is 196 × 257. Ranks for the three compressed images are 25, 50, and 100.

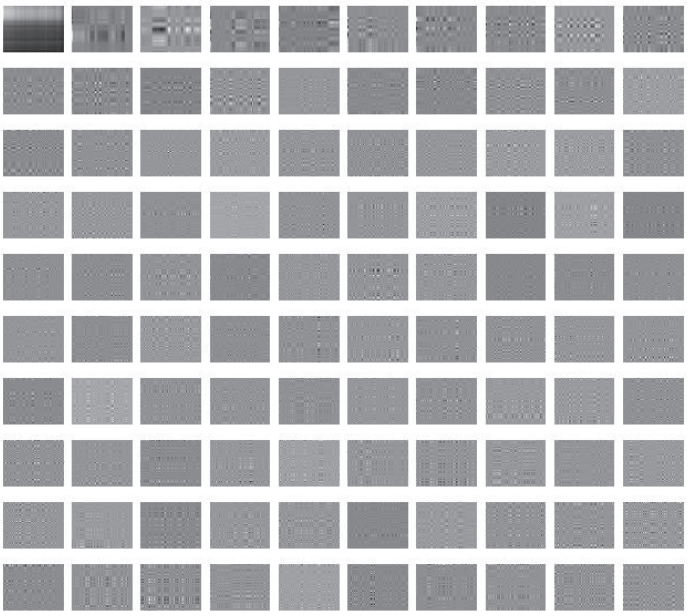

Figure 8: The first 100 rank-one bases of SVD on the Gleaners data matrix.

Figure 8 shows the first 100 eigen bases of SVD, where each basis image is also composed of various sizes of blocks. If we use the analogy of Lego, it seems reasonable to say that the leading bases generate large size Lego to construct the main structure, and the tail bases for tiny size, and the between for medium. The tiny size is so delicate that the reconstruction image will equal the documented image if we include all bases. This elaboration can also explain how MPCA works. Other successful stories about applying MPCA to array data include an EEG data set (Hung and Wang, 2013), cryo-EM image set (Chen et al., 2013) and Cardiology Ultrasound image set (Hung, 2013; Liu, 2013).

5. Concluding remarks and discussion

In this article, we introduced the method MPCA. When the data to be analyzed is high order array with spatial correlation, the traditional PCA, which treats each

data point as a long vector, is not efficient. MPCA, which models data with tensor structure, needs much fewer samples to present a satisfactory approximation. We first introduced the idea from the traditional PCA to MPCA. Then the exact algorithm was presented, and the properties were discussed. We also pointed out some applications of MPCA-based analyses, which have demonstrated the power of MPCA when the data is with tensor structures.

There is a convenient Matlab tensor toolbox (Bader et al., 2012) for general tensor objects handling and analysis. However, if the purpose is only to do MPCA, we recommend a fast and simple Matlab code: GLRAM.m provided by Ye (2005), which is based on iterative alternating eigenvalue decomposition. The original GLRAM is for order 2 tensors. However, it can be easily modified to extend to higher order. In addition to the data, the only inputs required for running GLRAM.m are the target size of dimensionality for the reduced subspace and a control parameter for stopping criterion.

References

Bader, B.W., Kolda, T.G. and others (2012). *MATLAB Tensor Toolbox Version 2.5,* Available online, http://www.sandia.gov/∼ tgkolda/TensorToolbox/

Chen, T.L., Hsieh, D.N., Hung, H., Tu, I.P., Wu, P.S., Wu, Y.M., Chang, W. and Huang, S.Y. (2013). γ-SUP: a clustering algorithm for cryo-electron microscopy images of asymmetric particles. *Annals of Applied Statistics*, to appear.

De Lathauwer, L., De Moor, B. and Vandewalle, J. (2000a). A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.,* 21, 1253-1278.

De Lathauwer, L., De Moor, B. and Vandewalle, J. (2000b). On the best rank-1 and rank-$(R_1, R_2, ..., R_N)$ approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.,* 21, 1324-1342.

Hung, H., Wu, P.S., Tu, I.P. and Huang, S.Y. (2012). On multilinear principal component analysis of order-two tensors. *Biometrika*, 99, 569-583.

Hung, H. and Wang, C.C. (2013). Matrix-variate logistic regression model with application to EEG data. *Biostatistics*, 14, 189-202.

Hung, M.H. (2013). *Comparison on Discriminant Analysis Using Logistic Regression and KNN Algorithm Based on Multilinear Principal Component Analysis for Study of Cardiology Ultrasound in Left Ventricle.* Master Thesis, supervised by Mong-Na Lo Huang and Kai-Hsien Hsieh, Department of Applied Mathematics, National Sun Yat-Sen University.

Jolliffe, I.T. (2002). *Principal Component Analysis.* Second edition, Springer-Verlag, New-York.

Kolda, T.G. and Bader, B.W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3), 455-500.

Li, B., Kim, M.K. and Altman, N. (2010). On dimension folding of matrix- or array-valued statistical objects. *Annals of Statistics*, 38, 1094-1121.

Liu, W.Y. (2013). *Comparison of Classification Effects between Principal Component and Multilinear Principal Component Analysis for Study of Cardiology Ultrasound in Left Ventricle.* Master Thesis, supervised by Mong-Na Lo Huang and Kai-Hsien Hsieh, Department of Applied Mathematics, National Sun Yat-Sen University.

Lu, H., Plataniotis, K.N. and Venetsanopoulos, A.N. (2008). MPCA: Multilinear principal component analysis of tensor objects. *IEEE Transactions on Neural Networks*, 19, 18-39.

Ye, J. (2005). Generalized low rank approximations of matrices. *Machine Learning*, 61, 167-191.

[Received December 2013; accepted January 2014.]

Journal of the Chinese Statistical Association
Vol. 52, (2014) 24–43

多重線性主成分分析簡介

陳定立[a]、陳素雲[a]、洪弘[b]、杜憶萍[a]

[a] 中央研究院統計科學研究所
[b] 國立台灣大學流行病學與預防醫學研究所

摘　要

在統計資料分析中, 主成分分析 (PCA) 是一個簡單且廣被使用的方法。它可以藉由共變異矩陣的特徵值分解來達成。傳統 PCA 處理向量變數, 每個觀察值都以向量形式表示。當觀察值是張量 (tensor) 物件時, 例如圖片、影像、 EEG 訊號、或是基因交互作用等, 傳統的 PCA 首先將這些張量物件向量化, 然後對一個大的共變異矩陣進行特徵值分解。這種對張量物件向量化的主成分分析, 可能會是困難而且效率差的。主要的原因是, 當樣本數比向量化資料的維度小時, 主成分分析的估計過程並不穩定。多重線性主成分分析 (MPCA) 是主成分分析的一種改良。在尋找主成分時, 它保留了觀察值本身的張量結構。保留此結構的主要優點在於節省了用以決定主成分的子空間所需的參數, 這減輕了高維度帶來的不利影響, 因而提高了估計及預測的效率。在本文中, 我們對 MPCA 的基本概念和技巧提供了一個易懂的介紹。從統計的觀點, 並根據一些眞實資料的應用, 讀者可以看到 MPCA 得以成功的理由。

關鍵詞: 維度縮減、高階奇異值分析、克羅內克積、多重線性主成分分析、主成分分析、奇異值分析、張量物件。

JEL classification: C18.