Data Min Knowl Disc (2011) 22:372–392
DOI 10.1007/s10618-010-0183-9

Matrix-variate and higher-order probabilistic projections

Shipeng Yu · Jinbo Bi · Jieping Ye

Received: 2 May 2009 / Accepted: 18 June 2010 / Published online: 10 July 2010

Abstract Feature extraction from two-dimensional or higher-order data, such as face images and surveillance videos, have recently been an active research area. There have been several 2D or higher-order PCA-style dimensionality reduction algorithms, but they mostly lack probabilistic interpretations and are difficult to apply with, e.g., incomplete data. It is also hard to extend these algorithms for applications where a certain region of the data point needs special focus in the dimensionality reduction process (e.g., the facial region in a face image). In this paper we propose a probabilistic dimensionality reduction framework for 2D and higher-order data. It specifies a particular generative process for this type of data, and leads to better understanding of some 2D and higher-order PCA-style algorithms. In particular, we show it actually takes several existing algorithms as its (non-probabilistic) special cases. We develop efficient iterative learning algorithms within this framework and study the theoretical properties of the stationary points. The model can be easily extended to handle special regions in the high-order data. Empirical studies on several benchmark data and real-world cardiac ultrasound images demonstrate the strength of this framework.

Responsible editor: Tao Li.

S. Yu (✉)
Siemens Medical Solutions USA, Malvern, PA, USA
e-mail: shipeng.yu@siemens.com

J. Bi
Department of Computer Science and Engineering,
University of Connecticut, Storrs, CT 06269, USA
e-mail: jinbo@engr.uconn.edu

J. Ye
Arizona State University, Tempe, AZ, USA
e-mail: jieping.ye@asu.edu

Keywords Dimensionality reduction · Higher-order principle component analysis · Low-rank matrix factorization · Probabilistic projection

1 Introduction

Recent technological innovations have unleashed a torrent of data with large numbers of dimensions (features or measurements). One of the key issues in such data analysis is the *curse of dimensionality*, i.e., an enormous number of samples is required to perform accurate prediction on problems with large numbers of features. Dimensionality reduction techniques for these vector-valued data, such as principal component analysis (PCA), have been widely applied to overcome this problem.

In recent years there are massive applications which generate *matrix data* where each datum is a two-dimensional (2D) matrix, or *higher-order data* where each datum is a tensor with higher ($\geq$3) dimensions. The latter is also called a *multiway array* in some applications. Hereafter we will use *matrix-variate data* to denote both 2D and higher-order data. Formerly, let an order-O datum be represented as $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_O}$, where I_j, $j = 1, \ldots, O$, is the dimensionality of the jth order of the datum. In this definition, an order-2 datum can be conveniently represented as a matrix $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$. Examples of matrix-variate data include image data (order 2) in face recognition where each datum is a face image, and videos (order 3) from surveillance cameras where each datum is a three-dimensional data cube (width $\times$ height $\times$ time).

Dimensionality reduction for this type of data is an active research area and has strong connections to low-rank tensor approximations. One can certainly convert every datum $\underline{\mathbf{X}}_i$ into a (column) vector by concatenating columns (or rows) and then apply traditional PCA, but such tensor-to-vector conversion may lead to loss of spatial locality information inherent in the data, and it also leads to very high dimensional representation of the data which is not feasible for PCA. Several 2D and higher-order algorithms have been proposed such as PCA-style unsupervised methods (e.g., Yang et al. 2004; Ye 2005; Lu et al. 2008) and multiway data analysis (see, e.g., Kolda and Bader 2007 for a survey). However, so far there lack probabilistic interpretations to these algorithms, and thus it is difficult to apply them incrementally (when new data are available), locally (when some sub-regions are of special interest), robustly (when there are some outlier points which might bias the projected dimensions), and when there are missing entries in some data points.

A probabilistic dimensionality reduction framework for matrix-variate data has many applications. One application in image analysis is to distinguish the region of interest (ROI) from the background in learning the dimensionality reduction mapping. This has great potential in, e.g., face recognition and medical imaging, since ideally the reconstruction should focus more on the ROI (i.e., the face and the organ) instead of the background pixels. Another application is to learn a mixture of mappings which reveal the projection in different perspectives and better reconstruct the datum. We will discuss more about these applications in the experiment section.

In this paper we introduce a family of probabilistic models, the *probabilistic higher-order PCA* (PHOPCA in short), for matrix-variate data. Interestingly, we will show that they recover the optimal solutions of several matrix-variate PCA-style algorithms

under mild conditions (Sect. 3). These models for the first time explicitly specify the generative process of matrix-variate objects. The well-known probabilistic PCA model (Tipping and Bishop 1999b; Roweis and Ghahramani 1999) is shown as the one-dimensional special case of PHOPCA. Efficient expectation-maximization (EM) algorithms are derived for learning the projection mapping, with less time complexity than the non-probabilistic counterparts (Sect. 4). Several extensions of the PHOPCA family are also discussed which show the additional benefits of the proposed probabilistic framework (Sect. 5). Empirical studies are shown using face images, USPS handwritten digits and a real application in cardiac view recognition of echocardiogram (Sect. 6).

2 Related work

PCA is a well-known dimensionality reduction method for one-dimensional data and has been extensively applied in machine learning and data mining (see, e.g., Jolliffe 2002). Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ denote a set of N (column) vectors of input data, PCA computes the eigen-decomposition of the sample covariance matrix of the data, $\frac{1}{N}\sum_i (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$ (with $\bar{\mathbf{x}}$ being the mean vector), and outputs an orthogonal transformation which contains the eigenvector(s) corresponding to the largest eigenvalue(s). Here $(\cdot)^\top$ denote matrix transpose. It is known that PCA captures the largest variance direction(s) of the data, and achieves the minimum reconstruction error (in vector 2-norm) among all projection directions with the same reduced dimensionality. PCA can be seen as a matrix factorization method (as it factorizes the sample covariance matrix), but it is significantly different from other special matrix factorization methods (such as non-negative matrix factorization; Lee and Seung 1999).

In the following we will review various non-probabilistic PCA-style algorithms for matrix-variate data (including both 2D and higher-order data), and the probabilistic PCA which is targeting at one-dimensional data.

2.1 2D/higher-order PCA-style algorithms

Order-2 data like images can be conveniently represented as a matrix and is therefore easier to analyze compared to higher-order data. Yang et al. (2004) and Ding and Ye (2005) proposed 2DPCA (or 2DSVD), in which a one-sided linear transformation, i.e., $\mathbf{X}_i\mathbf{R}$, is applied to each image $\mathbf{X}_i$. Let each image $\mathbf{X}_i$ be of size $m \times n$. By maximizing the data variance in the transformed space, 2DPCA leads to an analytical solution for $\mathbf{R}$ which contains the leading eigenvector(s) of the *right one-sided sample covariance matrix*, $\frac{1}{N}\sum_{i=1}^N (\mathbf{X}_i - \bar{\mathbf{X}})^\top (\mathbf{X}_i - \bar{\mathbf{X}})$, where $\bar{\mathbf{X}}$ is the mean image $\frac{1}{N}\sum_{i=1}^N \mathbf{X}_i$.

Later Ye (2005) introduced GLRAM, which considers two-sided transformation, $\mathbf{L}^\top\mathbf{X}_i\mathbf{R}$, to project each image $\mathbf{X}_i$ from space $\mathbb{R}^{m\times n}$ to $\mathbb{R}^{r\times c}$, where the left-sided mapping matrix $\mathbf{L}(m \times r)$ and right-sided mapping matrix $\mathbf{R}(n \times c)$ are both orthonormal ($r < m, c < n$). GLRAM minimizes the average matrix reconstruction errors of all images, $\frac{1}{N}\sum_{i=1}^N \|\mathbf{X}_i - \mathbf{L}\mathbf{Z}_i\mathbf{R}^\top\|_F^2$, with respect to both the reduced images $\mathbf{Z}_i$

and the mapping matrices $\mathbf{L}$ and $\mathbf{R}$. Here $\|\cdot\|_F$ denote matrix Frobenius norm. It is shown in Ye (2005) that GLRAM solves this optimization problem by choosing an initial $\mathbf{L}$ and $\mathbf{R}$ and iteratively computing the leading eigenvectors of the *left* and *right one-sided sample covariance matrices* as follows until convergence:

$$\mathbf{R} \leftarrow \text{top } c \text{ eigenvectors of } \sum_i \mathbf{X}_i^\top \mathbf{L}\mathbf{L}^\top \mathbf{X}_i \text{ with } \mathbf{L} \text{ fixed};$$
$$\mathbf{L} \leftarrow \text{top } r \text{ eigenvectors of } \sum_i \mathbf{X}_i \mathbf{R}\mathbf{R}^\top \mathbf{X}_i^\top \text{ with } \mathbf{R} \text{ fixed}.$$

It's shown that GLRAM obtains smaller reconstruction error than 2DPCA if maintaining the same compression ratio. There exist other variants of 2D PCA-style algorithms such as Inoue and Urahama (2006) and Shashua and Levin (2001).

GLRAM is further extended to higher-order multilinear PCA in Lu et al. (2008), where we consider tensor product $\underline{\mathbf{X}}_i \times_1 \mathbf{U}_1^\top \times \cdots \times_O \mathbf{U}_O^\top$ to map each datum from space $\mathbb{R}^{I_1 \times \cdots \times I_O}$ to $\mathbb{R}^{P_1 \times \cdots \times P_O}$. Here $\underline{\mathbf{X}}_i \times_j \mathbf{U}_j^\top$ is the jth-mode product of tensor $\underline{\mathbf{X}}_i$ by (transpose of) the orthonormal mapping matrix $\mathbf{U}_j \in \mathbb{R}^{I_j \times P_j} (P_j < I_j)$. Minimizing the reconstruction error leads to a similar eigen-decomposition algorithm to iteratively find $\mathbf{U}_j$ with other mapping matrices fixed ($j = 1, \ldots, O$). Other related work include Kolda (2001), Lathauwer et al. (2000), Lu et al. (2006), Sun et al. (2006), Vasilescu and Terzopoulos (2002), and Wang et al. (2005).

These 2D and higher-order methods are able to capture the spatial locality and are in general more efficient and memory-cheaper than standard PCA (after tensor-to-vector unfolding). They also yield lower reconstruction error if a same effective projection dimension is considered. Several other variants of PCA-style algorithms include Ding and Ye (2005) and Inoue and Urahama (2006). But up to now, no probabilistic explanation exists so far for this type of algorithms, therefore there is no principled solution to handle missing entries in the higher-order data.

These methods are also strongly related to the multiway data analysis (Kolda and Bader 2007), where matrix singular value decomposition (SVD) is extended to higher-order tensors using, e.g., Tucker and PARAFAC models. The main difference is that the PCA-style algorithms consider *i.i.d.* tensor samples, whereas multiway data analysis factorizes one big tensor. Our proposed probabilistic models only interpret the former, while a recently published paper (Chu and Ghahramani 2009) addresses the latter.

2.2 Probabilistic PCA

While PCA originates from the analysis of data variances, probabilistic PCA (PPCA) emerges from the statistics community and brings probabilistic explanations to PCA (Tipping and Bishop 1999b; Roweis and Ghahramani 1999). For input data $\mathbf{x} \in \mathbb{R}^d$, PPCA defines a latent variable model (of dimensionality $k < d$) as follows:

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{1}$$

with $\mathbf{z} \in \mathbb{R}^k$ the *latent variables*, $\mathbf{W}(d \times k)$ the *factor loadings*, $\boldsymbol{\mu} \in \mathbb{R}^d$ the mean vector, and $\boldsymbol{\epsilon}$ a noise process which follows a normal distribution with zero mean and isotropic variance, i.e., $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$. We also follow the convention to assume $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k)$ for the latent variables. Here $\mathbf{I}_k$ denote the identity matrix of size $k \times k$.

Conditioned on the latent variable $\mathbf{z}$, $\mathbf{x}$ is seen to follow a normal distribution, i.e., $\mathbf{x}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$. With $\mathbf{z}$ integrated out, $\mathbf{x}$ is also marginally normal distributed as $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I})$. Based on the Bayes' rule, the *a posteriori* distribution of $\mathbf{z}$ given $\mathbf{x}$ is also a normal:

$$\mathbf{z}|\mathbf{x} \sim \mathcal{N}\left((\mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I})^{-1}\mathbf{W}^\top(\mathbf{x} - \boldsymbol{\mu}), \sigma^2(\mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I})^{-1}\right).$$

This gives the projection model for any input data $\mathbf{x}$ under PPCA, with any fixed σ^2. When $\sigma^2 \to 0$, this normal distribution collapses to a single mass at the mean $(\mathbf{W}^\top\mathbf{W})^{-1}\mathbf{W}^\top(\mathbf{x} - \boldsymbol{\mu})$, which can be shown to be equivalent to PCA up to a scaling and rotation factor (Tipping and Bishop 1999b). This indicates that the *principal subspace* obtained from PPCA is the same as that in PCA.

The generative model for PPCA is similar to the factor analysis (Bartholomew and Knott 1999). The only difference is the noise process. In factor analysis the noise levels for different dimensions can be different, leading to a noise process $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$. Both models assume that in the noise process every two dimensions are independent. The different noise models lead to very different behavior in the projection model: factor analysis is covariant under component-wise rescaling of the data variables, whilst PPCA (and PCA) is covariant under rotation of the coordination system. For a detailed comparison of these two models please refer to Tipping and Bishop (1999b).

With a set of observations $\{\mathbf{x}_i\}_{i=1}^N$, the maximum likelihood (ML) estimate of $\mathbf{W}$ can be obtained by eigen-decomposing the sample covariance matrix. There also exists an expectation-maximization (EM) algorithm for $\mathbf{W}$, which is more efficient, memory-cheaper, and allows us to principally handle missing data and PPCA mixtures (see Tipping and Bishop 1999b for more details).

3 Probabilistic higher-order PCA

As will be seen later in Sect. 3.3, it is straightforward to extend the probabilistic model from second-order data to higher-order ($O \geq 3$) data. So for simplicity we mainly focus on second-order data (we call them "images" hereafter).

Given a set of N 2D images, our goal is to find a low dimensional feature representation for each image $\mathbf{X}_i$ which is a $m \times n$ matrix. The canonical solution is to vectorize each image into a vector by concatenating all the columns (or all rows), and then apply standard PCA or PPCA. But this is arguably not the best solution because: 1) concatenating columns (rows) remove the local correlation structure between columns (rows) in the image, and 2) vectorization leads to high dimensional representation of images, causing problems for PCA and PPCA in terms of both time and space complexities. We

propose a matrix-variate probabilistic model in this section, and build its connections to PPCA and other PCA-style algorithms.

Let us start with some matrix notations. For any matrices $\mathbf{A}(m \times n) = (a_{ij})$ and $\mathbf{B}(p \times q) = (b_{kl})$, we define $\text{vec}(\mathbf{A}) = (a_{11}, a_{21}, \ldots, a_{m1}, a_{12}, \ldots, a_{m2}, \ldots, a_{mn})^\top \in \mathbb{R}^{mn}$ the *vectorization* of $\mathbf{A}$, and $\mathbf{A} \otimes \mathbf{B} = (a_{ij}\mathbf{B}) \in \mathbb{R}^{mp \times nq}$ the *Kronecker product* of $\mathbf{A}$ and $\mathbf{B}$. Recall that $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\,\text{vec}(\mathbf{B})$ holds for any matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ with proper dimensions. Finally we denote $\mathbf{\Sigma} \succ 0$ if square matrix $\mathbf{\Sigma}$ is positive definite.

3.1 Preliminaries

Matrix-variate distributions, such as matrix-variate normal and Wishart, are widely used in statistics (Gupta and Nagar 1999). They are in general the 2D extensions of some multi-variate distributions and show interesting characteristics. Among them the matrix-variate normal is the most basic one.

Definition 1 (*Gupta and Nagar 1999*) Random matrix $\mathbf{X}(m \times n)$ is said to follow a *matrix-variate normal distribution* with *mean matrix* $\mathbf{M}(m \times n)$ and *covariance matrices* $\mathbf{\Sigma}(m \times m) \succ 0$ and $\mathbf{\Phi}(n \times n) \succ 0$, if $\text{vec}(\mathbf{X}^\top) \sim \mathcal{N}(\text{vec}(\mathbf{M}^\top), \mathbf{\Sigma} \otimes \mathbf{\Phi})$. This is denoted as $\mathbf{X} \sim \mathcal{N}(\mathbf{M}, \mathbf{\Sigma}, \mathbf{\Phi})$.[1]

Matrix-variate normal is defined through a normal distribution on the vectorized form of the matrix, with a special Kronecker covariance structure. It is not hard to see that the p.d.f. of $\mathbf{X} \sim \mathcal{N}(\mathbf{M}, \mathbf{\Sigma}, \mathbf{\Phi})$ is

$$\frac{1}{(2\pi)^{\frac{1}{2}mn}|\mathbf{\Sigma}|^{\frac{1}{2}n}|\mathbf{\Phi}|^{\frac{1}{2}m}}\,\text{etr}\left[-\tfrac{1}{2}\mathbf{\Sigma}^{-1}(\mathbf{X}-\mathbf{M})\mathbf{\Phi}^{-1}(\mathbf{X}-\mathbf{M})^\top\right],$$

with $\text{etr}(\cdot) = \exp(\text{tr}(\cdot))$ and $\text{tr}(\cdot)$ the matrix trace. Simple properties of matrix-variate normal include: 1) $\mathbf{X}^\top$, the transpose of $\mathbf{X}$, follows $\mathcal{N}(\mathbf{M}^\top, \mathbf{\Phi}, \mathbf{\Sigma})$; 2) The rows and/or columns of $\mathbf{X}$ are independent if $\mathbf{\Sigma}$ and/or $\mathbf{\Phi}$ are diagonal; 3) With $m = 1$ or $n = 1$, matrix-variate normal reduces to multi-variate normal.

Following this definition we can also define a similar normal distribution for higher-order ($O > 2$) tensors, with a special Kronecker covariance $\mathbf{\Sigma}_1 \otimes \cdots \otimes \mathbf{\Sigma}_O$ for the "vectorized" or "unfolded" form of tensor $\underline{\mathbf{X}}$. Note that the vector unfolding should happen from order O back to order 1.

3.2 Probabilistic second-order PCA

Let each image $\mathbf{X}$ be a $m \times n$ matrix. To directly model the spatial locality of the data, we propose the probabilistic second-order PCA, or the second-order PHOPCA, based on the matrix-variate normal assumption and assume the following *two-sided latent variable model*:

[1] For simplicity we overload symbol $\mathcal{N}$ to denote both multi-variate normal distribution (with 2 parameters) and matrix-variate normal distribution (with 3 parameters).

$$\mathbf{X} = \mathbf{L}\mathbf{Z}\mathbf{R}^\top + \mathbf{M} + \boldsymbol{\Upsilon}, \tag{2}$$

where $\mathbf{L}(m \times r)$ and $\mathbf{R}(n \times c)$ are the *row* and *column loading matrices*, and $\mathbf{Z}(r \times c)$ is the *latent variable core* of $\mathbf{X}$, with $r \leq m$, $c \leq n$ the *row* and *column PCA dimensions*, respectively. $\mathbf{M}(m \times n)$ is the mean matrix, and $\boldsymbol{\Upsilon}$ is a matrix-variate noise process. In this probabilistic framework, we assume matrix-variate normal for both $\mathbf{Z}$ and $\boldsymbol{\Upsilon}$: $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_r, \mathbf{I}_c)$, and $\boldsymbol{\Upsilon} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I}_m, \sigma\mathbf{I}_n)$ with noise level $\sigma > 0$. This noise model will be further discussed in Sect. 5.

From (2) we see that PHOPCA directly models the 2D image $\mathbf{X}$ as a two-sided mapping, and $\mathbf{Z}$ is the low-dimensional mapping. The local structure is kept in $\mathbf{L}$ and $\mathbf{R}$. In other words, the image $\mathbf{X}$ is generated by sampling a (smaller) latent variable core $\mathbf{Z}$, applying the row and column loadings, and adding a mean and some noise to every entry. With fixed dimensions (r, c), all the parameters in second-order PHOPCA are $\{\mathbf{L}, \mathbf{R}, \mathbf{M}, \sigma\}$. When $m = 1$ or $n = 1$, $\mathbf{L}$ or $\mathbf{R}$ is a (positive) scalar, and second-order PHOPCA reduces to standard PPCA. Therefore, second-order PHOPCA is a *2D extension* of PPCA. If $r = m$ or $c = n$, projection happens only on one side of the image, and we call it *one-mode* second-order PHOPCA. In the general case when $r < m$, $c < n$, we call it *two-mode* second-order PHOPCA. In Sect. 4 we will show that one-mode second-order PHOPCA recovers 2DPCA (Yang et al. 2004), and two-mode second-order PHOPCA recovers the GLRAM solution (Ye 2005).

The definition (2) indicates that the image $\mathbf{X}$ follows a matrix-variate normal conditioned on the core $\mathbf{Z}$. But unlike in PPCA, if we integrate $\mathbf{Z}$ out, $\mathbf{X}$ in general *does not* follow a matrix-variate normal. The *a posteriori* distribution of $\mathbf{Z}$ given $\mathbf{X}$ is also in general *not* matrix-variate normal, but for one-mode second-order PHOPCA it *is*. For instance if $\mathbf{L} = \mathbf{I}_m$, the *a posteriori* distribution of the core $\mathbf{Z}$ give image $\mathbf{X}$ is $\mathcal{N}(\mathbf{B}, \mathbf{I}_m, \mathbf{S})$, with $\mathbf{B} = (\mathbf{X} - \mathbf{M})\mathbf{R}(\mathbf{R}^\top\mathbf{R} + \sigma^2\mathbf{I})^{-1}$, $\mathbf{S} = \sigma^2(\mathbf{R}^\top\mathbf{R} + \sigma^2\mathbf{I})^{-1}$.

To see the connection between second-order PHOPCA and PPCA, it is easy to obtain that in the vectorized form, Eq. 2 is a special PPCA model $\text{vec}(\mathbf{X}^\top) = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$, with $\mathbf{z} = \text{vec}(\mathbf{Z}^\top)$, $\boldsymbol{\mu} = \text{vec}(\mathbf{M}^\top)$, $\mathbf{W} = \mathbf{L} \otimes \mathbf{R}$, and $\boldsymbol{\epsilon} = \text{vec}(\boldsymbol{\Upsilon}^\top)$.[2] This means second-order PHOPCA enforces a *Kronecker structure* to the factor loadings $\mathbf{W}$, and this is the key why PHOPCA is able to take into account the row-wise and column-wise correlations in the images. Second-order PHOPCA also has much less free parameters compared to a standard PPCA model on $\text{vec}(\mathbf{X}^\top)$. Learning under PHOPCA model will be discussed in Sect. 4.

3.3 Probabilistic higher-order PCA

We can easily extend the second-order PHOPCA to higher order by realizing that matrix product $\mathbf{L}\mathbf{Z}\mathbf{R}^\top$ in (2) is simply the first and second-mode product of $\mathbf{Z}$ with $\mathbf{L}$ and $\mathbf{R}$ in the tensor form, i.e., $\mathbf{Z} \times_1 \mathbf{L} \times_2 \mathbf{R}$. For an order-$O$ tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_O}$, we propose PHOPCA which assumes an *order-O latent variable model* as:

[2] This can be proved by applying formula $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\,\text{vec}(\mathbf{B})$ on both side of Eq. 2.

$$\underline{\mathbf{X}} = \underline{\mathbf{Z}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times \cdots \times_O \mathbf{U}_O + \underline{\mathbf{M}} + \underline{\boldsymbol{\Upsilon}},$$

with $\mathbf{U}_j \in \mathbb{R}^{I_j \times P_j}$ the jth-mode factor loadings ($P_j \leq I_j$), and $\underline{\mathbf{Z}} \in \mathbb{R}^{P_1 \times \cdots \times P_O}$ the latent variable core of $\underline{\mathbf{X}}$. $\underline{\mathbf{M}}$ is the mean tensor, and $\underline{\boldsymbol{\Upsilon}}$ is an order-O noise process. A tensor extension of matrix-variate normal distribution can be assigned to $\underline{\mathbf{Z}}$ and $\underline{\boldsymbol{\Upsilon}}$, similar to that in second-order PHOPCA. The connection to PPCA also holds for general PHOPCA models with a proper tensor-to-vector unfolding and the special Kronecker factor loadings $\mathbf{W} = \mathbf{U}_1 \otimes \cdots \otimes \mathbf{U}_O$. When projection happens only on one mode of the tensor (i.e., $P_j < I_j$, $P_k = I_k$ when $k \neq j$), we call it *one-mode* PHOPCA.

4 Learning in PHOPCA models

In this section we discuss how to optimize the parameters in the proposed PHOPCA models. For simplicity we again start with second-order PHOPCA, and then extend to general PHOPCA models. Given N images $\{\mathbf{X}_i\}_{i=1}^N$ which we assume are samples from second-order PHOPCA model (2), we focus on learning the loading matrices $\mathbf{L}$ and $\mathbf{R}$ (with fixed PCA dimensions) mainly using EM type algorithms. We will show that there is in general no analytical ML solutions for PHOPCA projections, but for one-mode PHOPCA there is a *global optimal solution* (up to a scaling and rotation factor). These learning algorithms provide insights and probabilistic explanations to existing PCA-style algorithms, and can be easily extended to handle mixture models and other noise models.

By definition (2) we see that $\mathbf{M}$ is the mean of the matrix-variate normal, and an easy derivation shows that its ML estimate is $\mathbf{M} = \frac{1}{N}\sum_i \mathbf{X}_i$. Therefore for simplicity we drop $\mathbf{M}$ in the following derivations.

4.1 Learning in one-mode second-order PHOPCA

Without loss of generality, we consider the right one-mode second-order PHOPCA model,

$$\mathbf{X}_i = \mathbf{Z}_i \mathbf{R}^\top + \boldsymbol{\Upsilon},$$

in which $\mathbf{Z}_i\,(m \times c)$ has the same number of rows as $\mathbf{X}_i\,(m \times n)$. As shown in Sect. 3.2, the *a posteriori* distribution of $\mathbf{Z}_i$ given $\mathbf{X}_i$ given all the model parameters is a matrix-variate normal $\mathcal{N}(\mathbf{B}_i, \mathbf{I}_m, \mathbf{S})$, with

$$\mathbf{B}_i = \mathbf{X}_i \mathbf{R}(\mathbf{R}^\top \mathbf{R} + \sigma^2 \mathbf{I})^{-1},\ \mathbf{S} = \sigma^2 (\mathbf{R}^\top \mathbf{R} + \sigma^2 \mathbf{I})^{-1}. \tag{3}$$

Treating $\mathbf{Z}_i$ as the latent variable, we can derive a standard EM algorithm to learn $\mathbf{R}$ and σ iteratively. In the E-step we calculate the *a posteriori* distribution of $\mathbf{Z}_i$ which gives the sufficient statistics (3), and then in the M-step we maximize the expected log-likelihood of the images with respect to $\mathbf{R}$ and σ, which is:

$$-Nmn\log\sigma^2-\frac{1}{2\sigma^2}\sum_i\mathbb{E}\left(\|\mathbf{X}_i-\mathbf{Z}_i\mathbf{R}^\top\|_F^2\right)-\frac{1}{2}\sum_i\mathbb{E}\left(\|\mathbf{Z}_i\|_F^2\right).$$

Here all the expectations $\mathbb{E}(\cdot)$ are with respect to the matrix-variate normal distribution $\mathbf{Z}_i\sim\mathcal{N}(\mathbf{B}_i,\mathbf{I}_m,\mathbf{S})$. A page of mathematics leads to the following update equations:

$$\mathbf{R}=\left[\frac{1}{N}\sum_i\mathbf{X}_i^\top\mathbf{B}_i\right]\left[\frac{1}{N}\sum_i\mathbf{B}_i^\top\mathbf{B}_i+m\mathbf{S}\right]^{-1}, \tag{4}$$

$$\sigma^2=\frac{1}{mn}\left(\frac{1}{N}\sum_i\|\mathbf{X}_i-\mathbf{B}_i\mathbf{R}^\top\|_F^2+m\,\mathrm{tr}(\mathbf{R}^\top\mathbf{R}\mathbf{S})\right). \tag{5}$$

Finally we run (3), (4) and (5) until convergence. For a test image $\mathbf{X}_*$, (the distribution of) its PHOPCA projection is calculated as in (3). We yield both the mean projection and the covariance of the projection.

An important fact about this EM algorithm is that it leads to the *global optimal projection subspace* for one-mode second-order PHOPCA, as summarized in the following theorem.

Theorem 1 *Let* $\mathbf{G}=\frac{1}{N}\sum_i\mathbf{X}_i^\top\mathbf{X}_i$, *and* $\lambda_1\geq\cdots\geq\lambda_n$ *be its eigenvalues with eigenvectors* $\mathbf{u}_1,\ldots,\mathbf{u}_n$. *The EM algorithm for one-mode second-order PHOPCA leads to the following ML solutions for* $\mathbf{R}$ *and* σ^2*:*

$$\mathbf{R}=\mathbf{U}_c\left(\frac{1}{m}\mathbf{\Lambda}_c-\sigma^2\mathbf{I}\right)^{\frac{1}{2}}\mathbf{V},\quad\sigma^2=\frac{1}{m(n-c)}\sum_{j=c+1}^{n}\lambda_j,$$

where $\mathbf{\Lambda}_c=\mathrm{diag}(\lambda_1,\ldots,\lambda_c)$, $\mathbf{U}_c=[\mathbf{u}_1,\ldots,\mathbf{u}_c]$, *and* $\mathbf{V}$ *is an arbitrary* $c\times c$ *orthogonal matrix.*

Proof We give a sketch here. We plug (3) into (4) to find the stationary point of the EM updates. At convergence we have $\mathbf{R}=\sigma^2\mathbf{GRS}(\mathbf{SR}^\top\mathbf{GRS}+\sigma^4m\mathbf{S})^{-1}$. Let $\mathbf{R}=\mathbf{UDV}$ be its SVD, we have $\mathbf{S}=\sigma^2\mathbf{V}^\top(\mathbf{D}^2+\sigma^2\mathbf{I})^{-1}\mathbf{V}$. Then after some mathematics we obtain $\mathbf{U}^\top\mathbf{GU}=m(\mathbf{D}^2+\sigma^2\mathbf{I})$, which means $\mathbf{U}$ contains the eigenvectors of $\mathbf{G}$. Let $\mathbf{G}=\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ be its SVD, we have $\mathbf{D}=(\frac{1}{m}\mathbf{\Lambda}-\sigma^2\mathbf{I})^{\frac{1}{2}}$. Finally a similar study as that in Tipping and Bishop (1999b) shows that $\mathbf{D}$ corresponds to the largest c eigenvalues. This gives $\mathbf{R}$. Plug this into (5) leads to the solution for σ^2. □

This theorem generalizes the optimal solution of PPCA (Tipping and Bishop 1999b) for which $m=1$. It is seen that the ML estimate of noise level σ^2 is the average of the rest $n-c$ eigenvalues divided by the number of rows m. A similar result exists for $\mathbf{L}$ if we do the left one-mode second-order PHOPCA with $\mathbf{R}=\mathbf{I}_n$. If the arbitrary rotation matrix $\mathbf{V}$ needs to be identified, one can run SVD to $\mathbf{R}^\top\mathbf{R}$ to recover it.

For a test image $\mathbf{X}_*$, its PHOPCA projection converges to a point mass $\mathbf{B}_*=\mathbf{X}_*\mathbf{U}_c\sqrt{m}\mathbf{\Lambda}_c^{-\frac{1}{2}}\mathbf{V}$ when $\sigma\to 0$. Note that matrix $\mathbf{G}$ in Theorem 1 is precisely the right one-sided sample covariance used in Yang et al. (2004). Therefore, we have the following corollary:

Corollary 2 *The right one-mode second-order PHOPCA recovers the 2DPCA algorithm proposed in Yang et al. (2004) when* $\sigma \to 0$*, up to a scaling and rotation factor.*

This indicates that one-mode second-order PHOPCA provides a *probabilistic explanation* to the 2DPCA algorithm (Yang et al. 2004). The EM algorithm provides another way of calculating those projection directions.

4.2 Learning in general second-order PHOPCA

In the general two-mode second-order PHOPCA model, we encounter some difficulty since the *a posteriori* distribution of $\mathbf{Z}_i$ given $\mathbf{X}_i$, $P(\mathbf{Z}_i|\mathbf{X}_i, \mathbf{L}, \mathbf{R}) \propto P(\mathbf{X}_i|\mathbf{Z}_i, \mathbf{L}, \mathbf{R})P(\mathbf{Z}_i)$, is in general not matrix-variate normal. In this subsection we solve this optimization problem using the *variational EM* (Jordan et al. 1999), in which we maximize a lower bound of the data log-likelihood with respect to some *variational parameters* in the E-step, and with respect to $\mathbf{L}$ and $\mathbf{R}$ in the M-step. Please refer to Jordan et al. (1999) for more details on this type of algorithms.

In variational EM we need to choose a *variational distribution* $Q(\mathbf{Z}_i)$ to approximate the true posterior, which is here $P(\mathbf{Z}_i|\mathbf{X}_i, \mathbf{L}, \mathbf{R})$. In the following we use a matrix-variate normal, $Q \triangleq \mathcal{N}(\mathbf{B}_i, \mathbf{T}, \mathbf{S})$, with mean $\mathbf{B}_i(r \times c)$ and covariances $\mathbf{T}(r \times r) \succ 0$, $\mathbf{S}(c \times c) \succ 0$ being variational parameters. Then the lower-bound to be maximized is $\sum_i \int Q \log \frac{P}{Q} d\mathbf{Z}_i$, i.e., the sum of the KL-divergence between Q and P for each image i. We omit the derivations and summarize the results in the following.

In the variational E-step, the lower bound is maximized with respect to the variational parameters $\mathbf{B}_i$, $\mathbf{T}$ and $\mathbf{S}$. It turns out that:

$$\mathbf{T} = c\,\sigma^2 \left[\operatorname{tr}(\mathbf{R}^\top \mathbf{R}\mathbf{S})\mathbf{L}^\top \mathbf{L} + \sigma^2 \operatorname{tr}(\mathbf{S})\,\mathbf{I}_r\right]^{-1}, \tag{6}$$

$$\mathbf{S} = r\,\sigma^2 \left[\operatorname{tr}(\mathbf{L}^\top \mathbf{L}\mathbf{T})\mathbf{R}^\top \mathbf{R} + \sigma^2 \operatorname{tr}(\mathbf{T})\,\mathbf{I}_c\right]^{-1}, \tag{7}$$

and each $\mathbf{B}_i$ needs to satisfy $\mathbf{L}^\top \mathbf{L}\mathbf{B}_i\mathbf{R}^\top \mathbf{R} + \sigma^2 \mathbf{B}_i = \mathbf{L}^\top \mathbf{X}_i \mathbf{R}$. To solve this we need to make a vectorization on both sides and solve a (big) linear equation:

$$\left[(\mathbf{R}^\top \mathbf{R}) \otimes (\mathbf{L}^\top \mathbf{L}) + \sigma \mathbf{I}_c \otimes \sigma \mathbf{I}_r\right] \operatorname{vec}(\mathbf{B}_i) = \operatorname{vec}(\mathbf{L}^\top \mathbf{X}_i \mathbf{R}) \tag{8}$$

with respect to $\operatorname{vec}(\mathbf{B}_i)$, and then reshape it back to get $\mathbf{B}_i$. When σ is small (e.g., $\sigma < 0.001$), however, the σ term in the equation corresponds to a small add-on (σ^2) to the diagonal entries of matrix $(\mathbf{R}^\top \mathbf{R}) \otimes (\mathbf{L}^\top \mathbf{L})$. In this case we might safely ignore the σ term and get

$$\mathbf{B}_i = (\mathbf{L}^\top \mathbf{L})^{-1} \mathbf{L}^\top \mathbf{X}_i \mathbf{R} (\mathbf{R}^\top \mathbf{R})^{-1} = \mathbf{L}^+ \mathbf{X}_i \mathbf{R}^{+\top} \tag{9}$$

to remove the computational burden. The entry-wise difference is at most $\mathcal{O}(\sigma^2)$. Here $\mathbf{L}^+$ ($\mathbf{R}^+$) denote the pseudo-inverse of $\mathbf{L}$ ($\mathbf{R}$).

In the variational M-step, we maximize the lower bound with respect to factor loadings $\mathbf{L}$ and $\mathbf{R}$ to get:

$$\mathbf{L} = \left[\frac{1}{N}\sum_i \mathbf{X}_i\mathbf{R}\mathbf{B}_i^\top\right]\left[\frac{1}{N}\sum_i \mathbf{B}_i\mathbf{R}^\top\mathbf{R}\mathbf{B}_i^\top + \mathrm{tr}(\mathbf{R}^\top\mathbf{R}\mathbf{S})\,\mathbf{T}\right]^{-1}, \tag{10}$$

$$\mathbf{R} = \left[\frac{1}{N}\sum_i \mathbf{X}_i^\top\mathbf{L}\mathbf{B}_i\right]\left[\frac{1}{N}\sum_i \mathbf{B}_i^\top\mathbf{L}^\top\mathbf{L}\mathbf{B}_i + \mathrm{tr}(\mathbf{L}^\top\mathbf{L}\mathbf{T})\,\mathbf{S}\right]^{-1}. \tag{11}$$

Finally we iterate (6)–(11) until convergence. Note that updates for $\mathbf{T}$ and $\mathbf{S}$ are coupled (so as for $\mathbf{L}$ and $\mathbf{R}$). σ can also be optimized if desired. It is easy to check that these equations lead to one-mode second-order PHOPCA updates when we fix $\mathbf{L} = \mathbf{I}_m$ (or $\mathbf{R} = \mathbf{I}_n$).

Unlike the one-mode model, the general second-order PHOPCA does not have an analytical global solution. When $\sigma \to 0$, the variational parameters $\mathbf{T}$ and $\mathbf{S}$ tend to be zero matrices, and the variational posterior of $\mathbf{Z}_i$ tend to decouple to a single mass (9) (this is also how to calculate the PHOPCA projection for a test image $\mathbf{X}_*$). In this case the iterative updates (9), (10) and (11) lead to the following important result:

Theorem 3 *Let* $\mathbf{G}(\mathbf{L}) = \frac{1}{N}\sum_i \mathbf{X}_i^\top\mathbf{L}\mathbf{L}^+\mathbf{X}_i$ *and* $\mathbf{H}(\mathbf{R}) = \frac{1}{N}\sum_i \mathbf{X}_i\mathbf{R}\mathbf{R}^+\mathbf{X}_i^\top$ *be two matrix-valued functions with input matrix* $\mathbf{L}(m \times r)$ *and* $\mathbf{R}(n \times c)$. *Let* $\mathbf{U}_c(\mathbf{L})$ *and* $\mathbf{V}_r(\mathbf{R})$ *contain eigenvectors of* $\mathbf{G}(\mathbf{L})$ *and* $\mathbf{H}(\mathbf{R})$ *with leading c and r eigenvalues, respectively. Then the stationary point of zero-noise, general second-order PHOPCA algorithm* (9)–(11) *satisfies:*

$$\mathbf{R}\mathbf{R}^+ = \mathbf{U}_c(\mathbf{L})\mathbf{U}_c(\mathbf{L})^\top,\ \mathbf{L}\mathbf{L}^+ = \mathbf{V}_r(\mathbf{R})\mathbf{V}_r(\mathbf{R})^\top. \tag{12}$$

Proof We give a sketch here. When $\sigma = 0$, $\mathbf{S}$ and $\mathbf{T}$ are zero matrices. With $\mathbf{L}$ fixed, plugging (9) into (11) yields $\mathbf{R} = \mathbf{G}\mathbf{R}\left[\mathbf{R}^\top\mathbf{G}\mathbf{R}\right]^{-1}\mathbf{R}^\top\mathbf{R}$ at the stationary point. Let $\mathbf{R} = \mathbf{E}\mathbf{D}\mathbf{F}^\top$ be its SVD, we have $\mathbf{E}\mathbf{E}^\top\mathbf{G}\mathbf{E} = \mathbf{G}\mathbf{E}$. To get $\mathbf{E}$ we eigendecompose $\mathbf{E}^\top\mathbf{G}\mathbf{E} = \mathbf{P}\boldsymbol{\Psi}\mathbf{P}^\top$ and obtain $\mathbf{E}\mathbf{P}\boldsymbol{\Psi} = \mathbf{G}\mathbf{E}\mathbf{P}$. This indicates $\mathbf{E}\mathbf{P}$ is the eigenvector of $\mathbf{G}$, and the only stable stationary solution is $\mathbf{E}\mathbf{P} = \mathbf{U}_c$. Then we have $\mathbf{R}\mathbf{R}^+ = \mathbf{E}\mathbf{E}^\top = \mathbf{U}_c\mathbf{U}_c^\top$. Similarly we can obtain $\mathbf{L}\mathbf{L}^+$ with $\mathbf{R}$ fixed. □

Theorem 3 builds an important connection between general second-order PHOPCA models and the GLRAM (Ye 2005). If we write

$$\mathbf{G}(\mathbf{L}) = \frac{1}{N}\sum_i \mathbf{X}_i^\top\mathbf{V}_r(\mathbf{R})\mathbf{V}_r(\mathbf{R})^\top\mathbf{X}_i,\quad \mathbf{H}(\mathbf{R}) = \frac{1}{N}\sum_i \mathbf{X}_i\mathbf{U}_c(\mathbf{L})\mathbf{U}_c(\mathbf{L})^\top\mathbf{X}_i^\top$$

by plugging in (12) at the stationary point of second-order PHOPCA, this is exactly the stationary point of repeatedly calculating the SVD of $\mathbf{G}(\mathbf{L})$ and $\mathbf{H}(\mathbf{R})$ in GLRAM (cf. Sect. 2.1, also see Ye 2005). Therefore, we actually proved the following:

Corollary 4 *Zero-noise second-order PHOPCA models and the GLRAM (Ye 2005) have the same stationary point.*

Based on the connection between PHOPCA and PPCA, we also see that GLRAM (in its vectorized form) is indeed a PCA model. Actually when both $\mathbf{L}$ and $\mathbf{R}$ are constrained to be column orthonormal (as in GLRAM), the two-sided factorization $\mathbf{X} = \mathbf{L}\mathbf{Z}\mathbf{R}^\top$ has vectorized form $\text{vec}(\mathbf{X}^\top) = (\mathbf{L} \otimes \mathbf{R})\,\text{vec}(\mathbf{Z}^\top)$, which is indeed a PCA model since $\mathbf{L} \otimes \mathbf{R}$ is also orthonormal. Thus GLRAM defines a PCA-style factorization of the vectorized images, and constrains that *the orthonormal mapping matrix in PCA is a Kronecker product of two smaller-sized orthonormal matrices*. This explains why: 1) GLRAM has less space requirement than PCA in the vectorized space, and 2) GLRAM gets better reconstruction than its one-sided counterpart (Yang et al. 2004). Ye (2005) also suggests applying a PCA to $\text{vec}(\mathbf{Z}^\top)$ after GLRAM, i.e., a GLRAM + PCA, and it's clear from here that GLRAM + PCA is still a PCA model.

4.3 Learning in general PHOPCA

The learning algorithms for second-order PHOPCA can be extended to higher-order PHOPCA. For one-mode PHOPCA where projection only happens at the jth-mode, we can "unfold" the other modes to yield a big matrix $\underline{\mathbf{X}}_i^{(j)} \in \mathbb{R}^{I_j \times (I_{j+1} I_{j+2} \ldots I_O I_1 I_2 \ldots I_{j-1})}$ for each tensor $\underline{\mathbf{X}}_i$, and apply the algorithm in Sect. 4.1. A similar theorem like Theorem 1 exists, and after convergence we find the globally optimal projection subspace in mode j. For general PHOPCA where we need to project in at least two modes, no global optimum exists and we can turn to variational EM algorithms similar to those described in Sect. 4.2. In the most interesting case where the noise level $\sigma \to 0$, the E-step (9) is extended to get the core $\underline{\mathbf{B}}_i$ as the all-mode product:

$$\underline{\mathbf{B}}_i = \underline{\mathbf{X}}_i \times_1 \mathbf{U}_1^{+\top} \times_2 \mathbf{U}_2^{+\top} \times \cdots \times_O \mathbf{U}_O^{+\top}, \tag{13}$$

and the M-step to update the factor loadings is now

$$\mathbf{U}_j = \left[\sum_i \underline{\mathbf{X}}_i^{(j)} \cdot \underline{\mathbf{E}}_i^{(j)\top}\right] \left[\sum_i \underline{\mathbf{E}}_i^{(j)} \cdot \underline{\mathbf{E}}_i^{(j)\top}\right]^{-1},$$

where $\underline{\mathbf{E}}_i = \underline{\mathbf{B}}_i \times_1 \mathbf{U}_1 \times \cdots \times_{j-1} \mathbf{U}_{j-1} \times_{j+1} \mathbf{U}_{j+1} \times \cdots \times_O \mathbf{U}_O \in \mathbb{R}^{I_1 \ldots I_{j-1} P_j I_{j+1} \ldots I_O}$ is the tensor product of $\underline{\mathbf{B}}_i$ with all-mode factor loadings except $\mathbf{U}_j$. A similar result like Theorem 3 can also be proved for general PHOPCA models, which indicates that this iterative algorithm yields the same stationary point as the higher-order multilinear PCA (Lu et al. 2008). A corollary is that after tensor-to-vector unfolding, multilinear PCA is still a PCA model with Kronecker-type factor loading matrix. For a test tensor $\underline{\mathbf{X}}_*$, the PHOPCA projection can be calculated using (13).

5 Discussions and extensions

The proposed PHOPCA framework extends PPCA to model matrix-variate objects, and is shown to take several 2D and higher-order PCA-style algorithms as (deterministic) special cases. The probabilistic interpretations provide additional insights to these algorithms (e.g., show that they are special PCA models after unfolding).

PHOPCA also enjoys less time complexity than the deterministic counterparts, and less space complexity than PPCA (after unfolding). The general second-order PHOPCA has time complexity $\mathcal{O}(tNmn\max(r,c))$ and space complexity $\mathcal{O}(mn)$, where r and c are the row-wise and column-wise projected dimensions. The higher-order PHOPCA has time complexity $\mathcal{O}(tN\prod_j I_j \max_j\{P_j\})$. Here t is the number of EM iterations.

PHOPCA fundamentally assumes a matrix-variate normal distribution for the latent variable model. In general, one can define various projection models with other matrix-variate distributions such as matrix-variate t distribution and Wishart distribution (Gupta and Nagar 1999). A similar formulation such as (2) can be defined with these alternative distributions, but learning and inference might be harder as they do not have the nice conjugate property as matrix-variate normal.

As of PPCA to PCA, PHOPCA provides additional benefits and possible extensions to the matrix-variate PCA-style algorithms:

- *Incremental learning* with newly obtained data is easy via the EM algorithm (which is crucial for real applications of matrix-variate PCA-style algorithms).
- *Missing data* can now be handled elegantly with an additional E-step (to estimate these missing values).
- *Other noise models* can be introduced for or against certain projection dimensions (or factors).
- *Mixture of PHOPCA* models can be easily derived (similar to Tipping and Bishop 1999a) for matrix-variate object clustering and (local) projections.
- *Robust matrix-variate PCA projections* can be introduced (via, e.g., a student't model instead of normal) when there are "outlier" matrix-variate objects.

In this paper we briefly go into two of these extensions, leaving the other extensions for future work. Some empirical results are shown in the next section.

5.1 Matrix-variate factor analysis

In PHOPCA the noise level σ is fixed for all input dimensions. If we allow them to differ, we are more in the family of factor analysis (FA) models for matrix-variate data. Let us take the second-order PHOPCA as an example in this subsection. We first give some motivation of this extension.

In some applications we want to focus on some specific sub-region(s) of the images, e.g., for face images we might only care about the facial region (eye, nose, mouse, etc.) and ignore the background. Or on the contrary we know some sub-regions contain higher noise, for instance, a medical scanner might obtain high noise in a certain region of the scanned image due to orientation and light angles. In terms of projection or feature extraction, we want the regions in focus to have lower reconstruction errors. Standard PHOPCA model cannot distinguish sub-regions (neither can 2DPCA and GLRAM), but we can easily extend PHOPCA to solve this problem if each of such regions is in a rectangular shape. This is not a limitation since we can always find a rectangular coverage of the actual shape.

In second-order PHOPCA the noise model for $\boldsymbol{\Upsilon}$ is $\boldsymbol{\Upsilon} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I}_m, \sigma\mathbf{I}_n)$, where the noise level for both covariance matrices is the same σ. This means the noise level for

every entry in the image is σ^2. If we change the noise model to be $\boldsymbol{\Upsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_0, \boldsymbol{\Phi}_0)$ such that both $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Phi}_0$ are diagonal matrices with $\boldsymbol{\Sigma}_0(k,k) = \sigma_{kk} > 0$ and $\boldsymbol{\Phi}_0(\ell,\ell) = \phi_{\ell\ell} > 0$, then the noise level at a specific entry (k,ℓ) in the image is effectively $\sigma_{kk}\phi_{\ell\ell}$. Therefore by choosing (or adapting) different (σ_{kk}) and $(\phi_{\ell\ell})$, we can make PHOPCA for or against certain regions in the images.

The EM (for one-mode PHOPCA) and variational EM (for general PHOPCA) can be easily derived for this new model (with fixed noise levels). For simplicity we only put the variational EM updates in the following. We need to iterate these equations until convergence. The derivation is straightforward.

$$\mathbf{T} = c\left[\operatorname{tr}(\mathbf{R}^\top \boldsymbol{\Phi}_0^{-1}\mathbf{R}\mathbf{S})\,\mathbf{L}^\top\boldsymbol{\Sigma}_0^{-1}\mathbf{L} + \operatorname{tr}(\mathbf{S})\,\mathbf{I}_r\right]^{-1},$$

$$\mathbf{S} = r\left[\operatorname{tr}(\mathbf{L}^\top \boldsymbol{\Sigma}_0^{-1}\mathbf{L}\mathbf{T})\,\mathbf{R}^\top\boldsymbol{\Phi}_0^{-1}\mathbf{R} + \operatorname{tr}(\mathbf{T})\,\mathbf{I}_c\right]^{-1},$$

$$\text{Solve } \mathbf{B}_i \text{ from } \mathbf{L}^\top\boldsymbol{\Sigma}_0^{-1}\mathbf{L}\mathbf{B}_i\mathbf{R}^\top\boldsymbol{\Phi}_0^{-1}\mathbf{R} + \mathbf{B}_i = \mathbf{L}^\top\boldsymbol{\Sigma}_0^{-1}\mathbf{X}_i\boldsymbol{\Phi}_0^{-1}\mathbf{R},$$

$$\mathbf{R} = \left[\sum_i \mathbf{X}_i^\top\boldsymbol{\Sigma}_0^{-1}\mathbf{L}\mathbf{B}_i\right]\left[\sum_i \mathbf{B}_i^\top\mathbf{L}^\top\boldsymbol{\Sigma}_0^{-1}\mathbf{L}\mathbf{B}_i + N\operatorname{tr}(\mathbf{L}^\top\boldsymbol{\Sigma}_0^{-1}\mathbf{L}\mathbf{T})\,\mathbf{S}\right]^{-1},$$

$$\mathbf{L} = \left[\sum_i \mathbf{X}_i\boldsymbol{\Phi}_0^{-1}\mathbf{R}\mathbf{B}_i^\top\right]\left[\sum_i \mathbf{B}_i\mathbf{R}^\top\boldsymbol{\Phi}_0^{-1}\mathbf{R}\mathbf{B}_i^\top + N\operatorname{tr}(\mathbf{R}^\top\boldsymbol{\Phi}_0^{-1}\mathbf{R}\mathbf{S})\,\mathbf{T}\right]^{-1}.$$

This model is a generalization of factor analysis to matrix-variate data. We recommend to fix $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Phi}_0$ before training or do cross-validation to learn these noise levels.

5.2 Mixture of matrix-variate projections

PHOPCA also allows us to consider MPHOPCA, a mixture of matrix-variate projections, where for each datum we sample one PHOPCA model from a pool of, say, K candidate models, and then sample the datum from that specific PHOPCA model. This leads to a clustering structure of the data, and following the same discussions in Tipping and Bishop (1999a) we can show that it is actually a Gaussian mixture model (after tensor-to-vector unfolding) with a specific covariance structure. Learning in MPHOPCA is basically (a weighted) PHOPCA learning with an additional E-step estimating the (soft) weights that each datum belongs to these component models. The details are omitted here.

6 Empirical study

In this section we validate the proposed PHOPCA models on some benchmark data and present a real-world application in dimensionality reduction for automatic cardiac view recognition. We will evaluate one-mode PHOPCA, general two-mode PHOPCA, the matrix-variate factor analysis and mixture of PHOPCA models on these problems.

Due to lack of higher-order ($O \geq 3$) benchmark data, we mainly present results on 2D matrix-variate data.

6.1 Benchmark data

We first test the proposed PHOPCA models on face image benchmark data sets ORL and AR. ORL[3] is a well-known data set for face recognition. It contains the face images of 40 persons, for a total of 400 images. The image size is 92×112. The major challenge here is the variation of the face pose. We use the whole image as an instance (i.e., the dimension is $92 \times 112 = 10304$). AR[4] is a large face image dataset. The instance of each face may contain large areas of occlusion, due to the presence of sun glasses and scarves. We use a subset of AR which contains 1638 face images of 126 persons. Its image size is 768×576. We first crop the image from row 100 to 500 and column 200 to 550, and then subsample the cropped images down to a size of $101 \times 88 = 8888$.

We first show some reconstructed images in Fig. 1 from ORL. As expected, general PHOPCA is better than (right) one-mode PHOPCA, and both FA-type noise model (row 4) and mixture of PHOPCAs (row 5) yield even better reconstructed images. For the FA-type noise model suppose the focus is on the facial region (rows [12, 80] and columns [40, 100]), and we assume low noise level (10^{-4}) to the diagonals in focused rows/columns whereas putting high noise level (10^{-3}) to the other diagonals. For the mixture model, 5 components are used with random image-to-component initialization. For all the testings we run 20 EM iterations. A typical learning curve is shown in Fig. 2.

A thorough comparison of reconstruction errors are shown in Fig. 3 for ORL and AR, respectively. The projection dimensions are (r, c) for general PHOPCA (and its variants), $\lceil rc/m \rceil$ for right one-mode PHOPCA, and $\lceil \frac{Nrc+nr+mc}{N+mn} \rceil$ for PCA (with N the number of images in the dataset). Note that we choose the projection dimensions for PCA and one-mode PHOPCA such that they have the same overall compression ratio. The performance is measured in *root mean square error* (RMSE) which is the square root of the mean reconstruction error. As suggested in Ye (2005), we just show the results with $r = c$. As expected, general PHOPCA outperforms one-mode PHOPCA and PCA in almost all dimensions, showing its great benefit for image compression. The mixture model (with 5 components) yields even better performance, but mainly in the region of smaller projection dimensions.

We confirmed experimentally that the one-mode second-order PHOPCA yields exactly the same results as the 2DPCA model (Yang et al. 2004), and general second-order PHOPCA model yields the same results as the GLRAM algorithm (Ye 2005). This is why we didn't plot the results for these two algorithms in Fig. 3. We also tested the efficiency of the proposed EM algorithms. General PHOPCA only takes about half the time as the GLRAM algorithm. On the ORL data set, for example, general

[3] http://www.uk.research.att.com/facedatabase.html.

[4] http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html.

Fig. 1 Reconstructions of some ORL images (92×112). From top to bottom: (1) original images; (2) using general PHOPCA (equivalent to GLRAM; Ye 2005); (3) using (right) one-mode PHOPCA (equivalent to 2DPCA; Yang et al. 2004); (4) using PHOPCA with FA noise model; (5) Mixture of PHOPCA with 5 components. Projection dimensions are $r = c = 15$

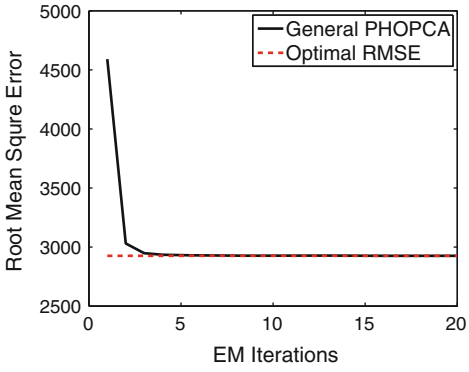

Fig. 2 A typical learning curve for PHOPCA. The root mean square reconstruction error converges to the optimal value after around 10 iterations

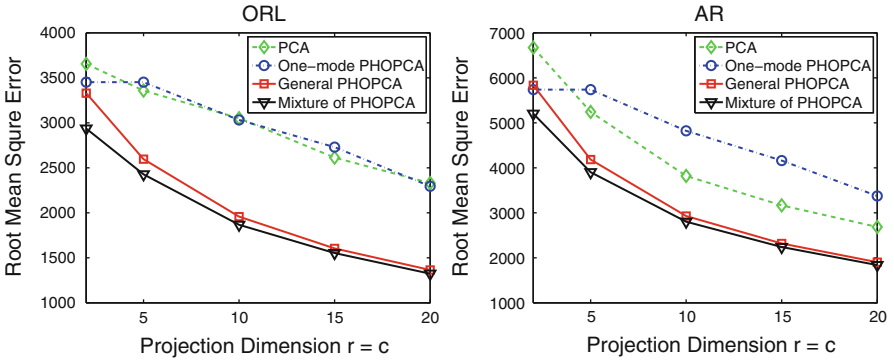

Fig. 3 Reconstruction performance comparison on two image data sets ORL (*left*) and AR (*right*). The performance is calculated using root mean square error (RMSE) of the reconstruction. The GLRAM algorithm (Ye 2005) yields exactly the same results as the general PHOPCA model, as shown in Theorem 3

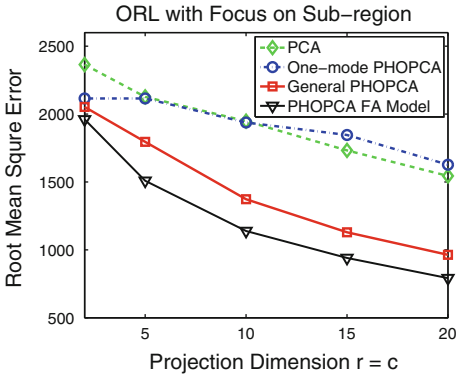

Fig. 4 Reconstruction performance comparison on the image data set ORL focusing on sub-regions. The performance is calculated using root mean square error (RMSE) of the reconstruction on rows [12,80] and columns [40,100]

PHOPCA takes in average 0.56 s to converge for $r = c = 5$, whilst GLRAM takes 0.90 s.

In Fig. 4, we compare different methods on the specified sub-region of ORL dataset. For the factor analysis model the focus is on the facial region (rows [12, 80] and columns [40, 100]), and the calculated RMSE value is only calculated in this subregion. It can be seen that the PHOPCA model with FA-type noise model yields the best performance.

6.2 Automatic cardiac view recognition

Ultrasound images of the heart are usually taken as 2D slice of the 3D heart from standardized 15 different angles. Diagnostic analysis of these images requires, as the first step, recognizing the pose of the heart so that spatial cardiac structures can be identified. Cardiac views are imaged from 4 windows: the parasternal, apical, subcostal and suprasternal windows, which leads up to 15 basic views. Automatic view recognition is the problem of automatically classifying cardiac ultrasound images with respect

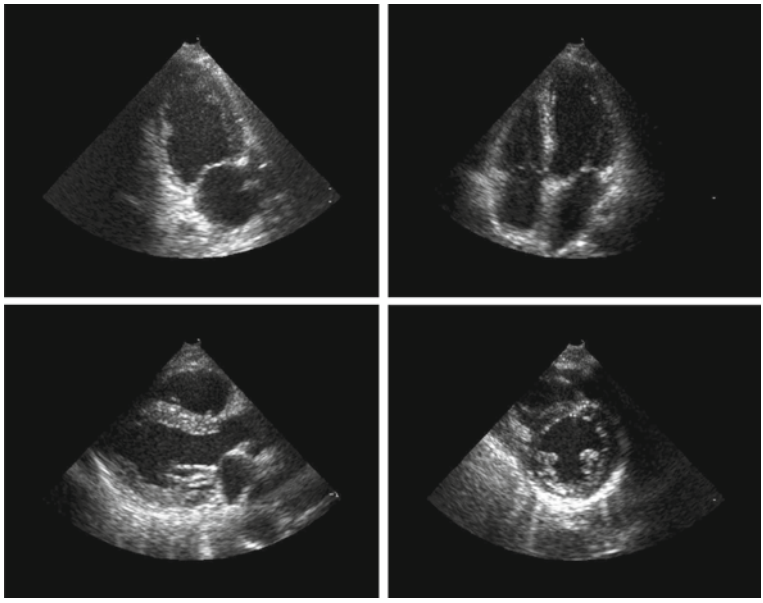

Fig. 5 Four views of the heart: top-left, apical 2 chamber (a2c); top-right, apical 4 chamber (a4c); bottom-left, parasternal long axis (plax); bottom-right, parasternal short axis (psax)

to their views. We collected patient data with various image quality from St. Francis Hospital at New York which contain largely 4 views. Figure 5 shows an example of each of them.

Ultrasound videos of 100 patients were collected where some patients had multiple images for one view or certain views missing, resulting in 72 a2c, 87 a4c, 80 plax and 83 psax clips of 480×640 image frames. Due to the high number of raw features, various dimensional reduction methods can be employed to extract useful features for discrimination from raw images or evaluated image features. In Fig. 6 we show one original image of each view and the reconstructed images from different methods. The projection dimension of second-order PHOPCA was $(r, c) = (10, 10)$, equivalent to 100 features. For PCA and one-mode PHOPCA, we determined the dimensions similarly as in previous experiments.

For the problem of automatic view recognition, we illustrate the potentials of PHOPCA by distinguishing psax clips from a4c clips. These images were randomly split into the training set (44 a4c vs. 42 psax) and test set (43 a4c vs. 41 psax). One-mode PHOPCA, general PHOPCA and mixture of PHOPCA were applied to the psax clips to produce the projection matrices $\mathbf{L}$ and $\mathbf{R}$. Then features for each clip were generated by projecting the first frame of the clip along the projection matrices. Moreover, PHOPCA with a focus area ($[150, 350] \times [250, 450]$), where noise level is 10^{-6} in contrast to 10^{-4} in the remaining area, was also used to emphasize the area that is the most discriminative for the psax view. The projection dimension of PHOPCA was $(r, c) = (10, 10)$ which generated acceptable accuracy as in Fig. 7. For PCA and one-mode PHOPCA, we determined the dimensions similarly as in previous

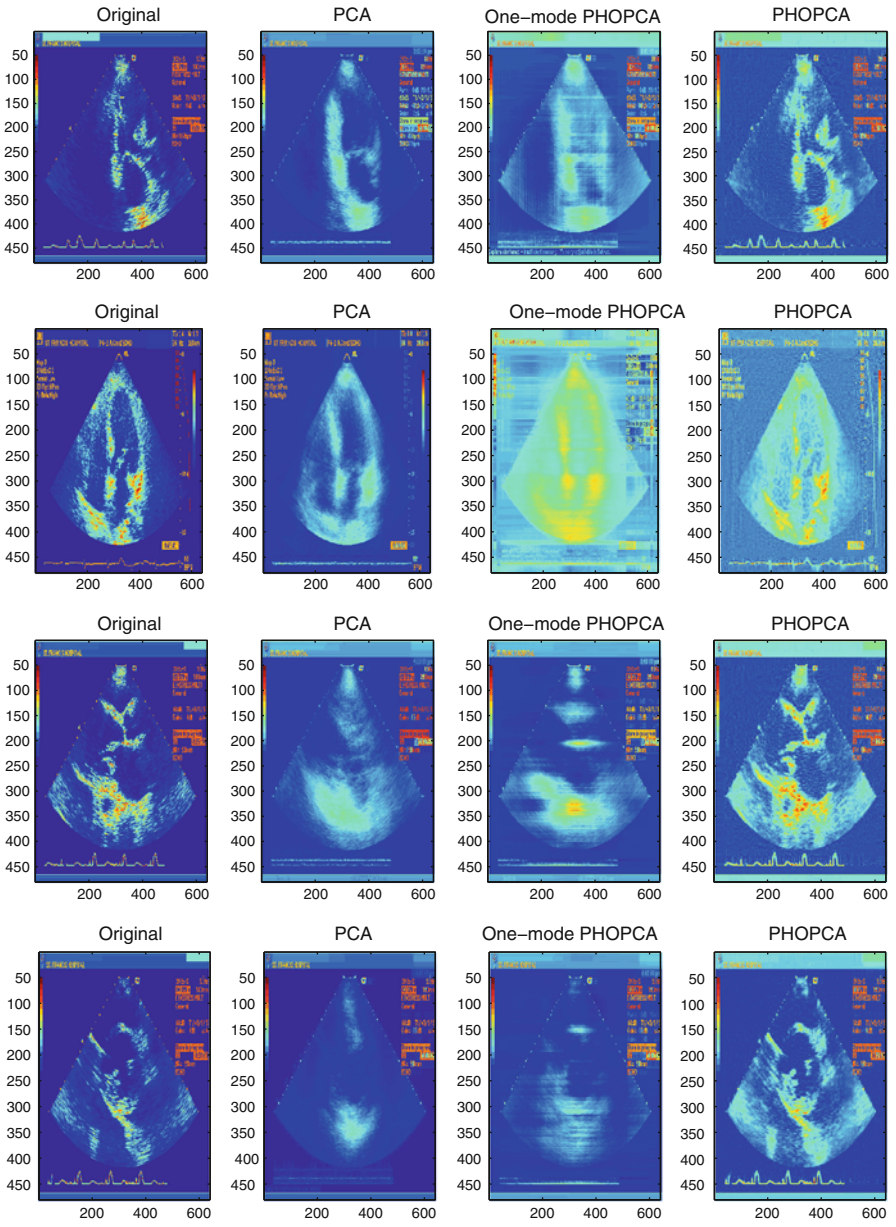

Fig. 6 One original image of each view and the reconstructed images from PCA, one-mode PHOPCA (equivalent to 2DPCA; Yang et al. 2004) and general PHOPCA (equivalent to GLRAM; Ye 2005). The views from top to bottom: a2c, a4c, plax, psax

experiments. Any suitable classification methods can then be used to construct a classifier using these features. In our experiments, we used least squares support vector machine (LSSVM) with threefold cross-validation. As shown in Fig. 7, mixture of

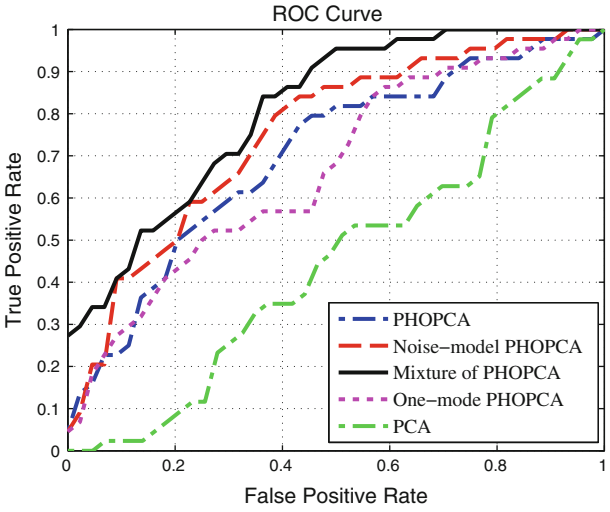

Fig. 7 The ROC curves of least-square support vector machines using different dimensionality reduction methods for feature extraction

PHOPCA and PHOPCA with a focus area outperformed other approaches. General PHOPCA is better than one-mode PHOPCA, and they are both significantly better than PCA.

7 Conclusion and future work

In this paper we introduced a family of probabilistic dimensionality reduction models called PHOPCA for matrix-variate data, including both 2D data as well as higher-order data. These models for the first time explicitly specify the generative process of matrix-variate objects, and the proposed optimization framework recovers the optimal solutions of several matrix-variate PCA-style algorithms under mild conditions. PHOPCA also takes the well-known probabilistic PCA model (for vectorized data) as a special case. Efficient EM algorithms are derived for learning the projection mapping, with less time complexity than the non-probabilistic counterparts. Several extensions of the PHOPCA family, including a factor analysis model for matrix-variate data and a mixture of PHOPCA model, are also discussed which show the additional benefits of the proposed probabilistic framework.

For future work we plan to apply the proposed models on data from other applications including surveillance video and gene arrays. The other extensions mentioned in this paper will also be explored with specific applications in medical imaging. For instance, for computer-aided detection in lung CT images we often see some patients with only partial lung CT scans. These images should be taken as the "outlier" images for the data sets and should be down-weighted in a feature extraction process. An interesting future work is to design a robust matrix-variate projection model to tackle this problem.

References

Bartholomew D, Knott M (1999) Latent variable models and factor analysis. Oxford University Press, New York

Chu W, Ghahramani Z (2009) Probabilistic models for incomplete multi-dimensional arrays. In: Proceedings of the international conference on artificial intelligence and statistics (AISTATS-12)

Ding C, Ye J (2005) 2-Dimensional singular value decomosition for 2D maps and images. In: SDM

Gupta AK, Nagar DK (1999) Matrix variate distributions. Chapman and Hall/CRC, Boca Raton, FL

Inoue K, Urahama K (2006) Equivalence of non-iterative algorithms for simultaneous low rank approximations of matrices. In: CVPR, pp 154–159

Jolliffe IT (2002) Principal component analysis. Springer Verlag, New York

Jordan MI, Ghahramani Z, Jaakkola T, Saul LK (1999) An introduction to variational methods for graphical models. Mach Learn 37(2):183–233

Kolda TG (2001) Orthogonal tensor decompositions. SIAM J Matrix Anal Appl 23(1):243–255

Kolda TG, Bader BW (2007) Tensor decompositions and applications. Technical Report SAND2007-6702. Sandia National Laboratories

Lathauwer LD, Moor BD, Vandewalle J (2000) On the best rank-1 and rank-(r1,r2,. . .,rn) approximation of higher-order tensors. SIAM J Matrix Anal Appl 21(4):1324–1342

Lee DD, Seung HS (1999) Learning the parts of objects with nonnegative matrix factorization. Nature 401:788–791

Lu H, Plataniotis KN, Venetsanopoulos AN (2006) Multilinear principal component analysis of tensor objects for recognition. In: Proceedings of the 18th international conference on pattern recognition, pp 776–779

Lu H, Plataniotis KN, Venetsanopoulos AN (2008) MPCA: multilinear principal component analysis of tensor objects. IEEE Trans Neural Netw 19(1):18–39

Roweis S, Ghahramani Z (1999) A unifying review of linear gaussian models. Neural Comput 11:305–345

Shashua A, Levin A (2001) Linear image coding for regression and classification using the tensor-rank principle. In: Proceedings of the international conference on computer vision and pattern recogniton, pp 42–49

Sun J, Tao D, Faloutsos C (2006) Beyond streams and graphs: dynamic tensor analysis. In: KDD, pp 374–383

Tipping ME, Bishop CM (1999a) Mixtures of probabilistic principal component analysers. Neural Comput 11(2):443–482

Tipping ME, Bishop CM (1999b) Probabilistic principal component analysis. J R Stat Soc B 61:611-622

Vasilescu MAO, Terzopoulos D (2002) Multilinear analysis of image ensembles: Tensorfaces. In: Proceedings of the 7th European conference on computer vision-Part I, pp 447–460

Wang H, Wu Q, Shi L, Yu Y, Ahuja N (2005) Out-of-core tensor approximation of multi-dimensional matrices of visual data. In: SIGGRAPH

Yang J, Zhang D, Frangi AF, Yang J-Y (2004) Two-dimensional pca: a new approach to appearance-based face representation and recognition. IEEE Trans Pattern Anal Mach Intell 26(1):131–137

Ye J (2005) Generalized low rank approximation of matrices. Mach Learn 61:167–191