Probabilistic Interpretations and Extensions for a Family of 2D PCA-style Algorithms

Shipeng Yu, Jinbo Bi
CAD and Knowledge Solutions,
Siemens Medical Solutions USA, Inc.
{shipeng.yu, jinbo.bi}@siemens.com

Jieping Ye
Arizona State University
Jieping.Ye@asu.edu

ABSTRACT

Recently 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. We propose a probabilistic framework to better understand the 2D and higher-order PCA-style algorithms, and show that it takes several existing algorithms as its (non-probabilistic) special cases. Efficient learning algorithms are proposed, and the stationary points are theoretically analyzed. Empirical studies on several benchmark data and real-world cardiac ultrasound images demonstrate the strength of this framework.

General Terms

Algorithms, Theory, Performance

Keywords

Probabilistic Dimensionality Reduction, Matrix Factorization, 2DPCA

1. INTRODUCTION

Principal component analysis (PCA) is a well-known dimensionality reduction method for 1D data and has been extensively applied in machine learning and pattern recognition [4]. 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). 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.

In recent years there are massive applications which generate higher-order data (multiway arrays, tensors), such as images (order 2) in face recognition and videos (order 3) from surveillance cameras. Each order-O datum is represented as $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_O}$, and an order-2 datum (e.g., an image) can be conveniently represented as a matrix $\mathbf{X}$. 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 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., [11, 12, 7]) and multiway data analysis (see a recent survey [6]). However, so far there lack probabilistic interpretations to these algorithms, and thus it is difficult to apply them incrementally (with new data), locally (for subregions), robustly (with outliers) and with missing entries.

Another important motivation of this work is the *noise model* which is explicitly or implicitly assumed by most of the previous deterministic work, i.e., they assume the *noise* or the reconstruct error is homogeneous in all the matrix entries. Therefore, the algorithm tries to reconstruct each entry with the same effort (or the reconstruction error of each entry contributes equally to the total loss). But in many matrix factorization problems, we know that some regions of the images are easy to contain higher noise than other parts of the images (e.g., a certain medical scanner might obtain high noise in a certain region of the scanned image due to orientation and light angles), or some regions are more important for the purpose of matrix reconstruction (e.g., the face part in a face image). Other examples include that sometimes the images might be annotated by some people who consistently mark in a specific region. In terms of reconstruction, these regions are likely to have higher noises.

In this paper we introduce a family of probabilistic models for 2D (and higher-order) data called the *probabilistic higher-order PCA* (PHOPCA), and show that they recover the optimal solutions of several PCA-style algorithms under mild conditions (Section 3). These models for the first time explicitly specify the generative process of higher-order objects, and take the probabilistic PCA [10, 8] as its 1D special case. Efficient EM-type algorithms are derived for learning, with less time complexity than the non-probabilistic counterparts (Section 4). Several extensions of PHOPCA are also discussed (Section 5). Some empirical results are shown using face images, USPS handwritten digits and a real application in cardiac view recognition of echocardiogram (Section 6).

DMMT'08, August 24, 2008, Las Vegas, USA.

2. RELATED WORK

2.1 2D/Higher-Order PCA-style Algorithms

For 2D data like images, in [11] 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 $m \times n$. With an orthonormal mapping matrix $\mathbf{R}$ of size $n \times c$ ($c < n$), it projects each image from space $\mathbb{R}^{m\times n}$ to $\mathbb{R}^{m\times c}$. The analytical solution for $\mathbf{R}$ 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. Later GLRAM [12] 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 $\mathbf{R}$. It is shown in [12] that GLRAM works by iteratively computing the leading eigenvectors of the *left* and *right one-sided sample covariance matrices* as follows until convergence:

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

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

GLRAM is further extended to higher-order multilinear PCA in [7], 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 ($j = 1, \ldots, O$).

These 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). But in terms of reconstruction error, standard PCA is shown to be however superior if the same effective projection dimensions are used (see, e.g., [12]). So far it is still not well understood how these algorithms relate to standard PCA. Several other variants of PCA-style algorithms include [1, 3].

These methods are strongly related to the multiway data analysis [6], 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.

2.2 Probabilistic PCA

Probabilistic PCA (PPCA) emerges from the statistics community and brings probabilistic explanations to PCA [10, 8]. For input data $\mathbf{x} \in \mathbb{R}^d$, PPCA defines a generative model as $\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$, 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 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$. We also follow the convention to assume $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k)$. To see its connection to PCA, we get the *a posteriori* distribution of $\mathbf{z}$ given $\mathbf{x}$ using the Bayes' rule, which is also a normal distribution. When $\sigma^2 \to 0$, the distribution collapses to a single mass at the mean $(\mathbf{W}^\top\mathbf{W})^{-1}\mathbf{W}^\top(\mathbf{x} - \boldsymbol{\mu})$, which turns out to be equivalent to PCA up to a scaling and rotation factor [10]. This indicates that the *principal subspace* obtained from PPCA is the same as that in PCA.

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 [10].

3. PROBABILISTIC HIGHER-ORDER PCA

We introduce the PHOPCA models in this section. For simplicity we mainly focus on second-order data (we call them "images" hereafter), and briefly mention the extensions to higher-order data in Section 3.3.

We start with some 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 $\boldsymbol{\Sigma} \succ 0$ if square matrix $\boldsymbol{\Sigma}$ is positive definite.

3.1 Preliminaries

Matrix-variate distributions, such as matrix-variate normal and Wishart, are widely used in statistics [2]. 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 (SEE [2]). *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 $\boldsymbol{\Sigma}(m \times m) \succ 0$ *and* $\boldsymbol{\Phi}(n \times n) \succ 0$, *if* $\text{vec}(\mathbf{X}^\top) \sim \mathcal{N}(\text{vec}(\mathbf{M}^\top), \boldsymbol{\Sigma} \otimes \boldsymbol{\Phi})$. *This is denoted as* $\mathbf{X} \sim \mathcal{N}(\mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Phi})$.[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 $\mathbf{X} \sim \mathcal{N}(\mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Phi})$ is

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

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 $\mathcal{N}(\mathbf{M}^\top, \boldsymbol{\Phi}, \boldsymbol{\Sigma})$; 2) The rows and/or columns of $\mathbf{X}$ are independent if $\boldsymbol{\Sigma}$ and/or $\boldsymbol{\Phi}$ 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

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

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, the second-order PHOPCA (or PSOPCA) is based on the matrix-variate normal assumption and assumes the following *two-sided latent variable model*:

$$\mathbf{X} = \mathbf{L}\mathbf{Z}\mathbf{R}^\top + \mathbf{M} + \mathbf{\Upsilon}, \tag{1}$$

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 $\mathbf{\Upsilon}$ is a matrix-variate noise process. In this probabilistic framework, we assume matrix-variate normal for both $\mathbf{Z}$ and $\mathbf{\Upsilon}$: $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_r, \mathbf{I}_c)$, and $\mathbf{\Upsilon} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I}_m, \sigma\mathbf{I}_n)$ with noise level $\sigma > 0$. This noise model will be further discussed in Section 5.

From (1) we see that each 2D 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 PSOPCA are $\{\mathbf{L}, \mathbf{R}, \mathbf{M}, \sigma\}$. When $m = 1$ or $n = 1$, $\mathbf{L}$ or $\mathbf{R}$ is a (positive) scalar, and PSOPCA reduces to standard PPCA. Therefore, PSOPCA 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* PSOPCA.

The definition (1) 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 PSOPCA 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 $\mathbf{B} = (\mathbf{X} - \mathbf{M})\mathbf{R}(\mathbf{R}^\top\mathbf{R} + \sigma^2\mathbf{I})^{-1}$, $\mathbf{S} = \sigma^2(\mathbf{R}^\top\mathbf{R} + \sigma^2\mathbf{I})^{-1}$.

Another strong connection between PSOPCA and PPCA is revealed via the following proposition (we put all the proofs into Appendix for clarity).

PROPOSITION 1. *In the vectorized form, PSOPCA* (1) *is a PPCA model* $\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$, *with* $\mathbf{x} = \mathrm{vec}(\mathbf{X}^\top)$, $\mathbf{z} = \mathrm{vec}(\mathbf{Z}^\top)$, $\boldsymbol{\mu} = \mathrm{vec}(\mathbf{M}^\top)$, $\mathbf{W} = \mathbf{L} \otimes \mathbf{R}$, *and* $\boldsymbol{\epsilon} = \mathrm{vec}(\mathbf{\Upsilon}^\top)$. *In this PPCA model* $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{rc})$ *and* $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_{mn})$.

This proposition states that PSOPCA is a special PPCA model (in its vectorized form) and enforces a *Kronecker structure* to the factor loadings $\mathbf{W}$. This Kronecker structure is the key why PSOPCA is able to take into account the row-wise and column-wise correlations in the images. PSOPCA also has much less free parameters compared to a standard PPCA model on the vectorized form of $\mathbf{X}$.

3.3 Probabilistic Higher-Order PCA

We can easily extend PSOPCA to PHOPCA by realizing that matrix product $\mathbf{L}\mathbf{Z}\mathbf{R}^\top$ in (1) 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 $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_O}$, PHOPCA assumes the *order-O latent variable model* as:

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

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 $\underline{\mathbf{\Upsilon}}$ is an order-O noise process. A tensor extension of matrix-variate normal distribution can be assigned to $\underline{\mathbf{Z}}$ and $\underline{\mathbf{\Upsilon}}$, similar to that in PSOPCA. Proposition 1 also holds for PHOPCA 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

For simplicity we again start with PSOPCA, the 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 the PSOPCA model (1), 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 PSOPCA (and PHOPCA) projections, but for one-mode PSOPCA (and 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 missing data, mixture models and other noise models.

By definition (1) 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 and assume $\mathbf{M}$ is subtracted from each image $\mathbf{X}_i$ before learning.

4.1 Learning in One-mode PSOPCA

Without loss of generosity, we consider right one-mode PSOPCA model $\mathbf{X}_i = \mathbf{Z}_i\mathbf{R}^\top + \mathbf{\Upsilon}$, in which $\mathbf{Z}_i(m \times c)$ has the same number of rows as $\mathbf{X}_i(m \times n)$. As shown in Section 3.2, the *a posteriori* distribution of $\mathbf{Z}_i$ given $\mathbf{X}_i$ is a matrix-variate normal $\mathcal{N}(\mathbf{B}_i, \mathbf{I}_m, \mathbf{S})$, with

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

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 (2), and then in the M-step we maximize the expected log-likelihood of the images with respect to $\mathbf{R}$ and σ, which is: $-\frac{1}{2\sigma^2}\sum_i \mathbb{E}\left(\|\mathbf{X}_i - \mathbf{Z}_i\mathbf{R}^\top\|_F^2\right) - \frac{1}{2}\sum_i \mathbb{E}\left(\|\mathbf{Z}_i\|_F^2\right) - Nmn\log\sigma^2$. Here all the expectations $\mathbb{E}(\cdot)$ are with respect to $\mathbf{Z}_i \sim \mathcal{N}(\mathbf{B}_i, \mathbf{I}_m, \mathbf{S})$. A page of mathematics leads to the following update equations:

$$\mathbf{R} = \left[\frac{1}{N}\sum_i \mathbf{X}_i^\top\mathbf{B}_i\right]\left[\frac{1}{N}\sum_i \mathbf{B}_i^\top\mathbf{B}_i + m\mathbf{S}\right]^{-1}, \tag{3}$$

$$\sigma^2 = \frac{1}{mn}\left(\frac{1}{N}\sum_i \|\mathbf{X}_i - \mathbf{B}_i\mathbf{R}^\top\|_F^2 + m\,\mathrm{tr}(\mathbf{R}^\top\mathbf{R}\mathbf{S})\right). \tag{4}$$

Finally we run (2), (3) and (4) until convergence. An important fact about this EM algorithm is that it leads to the *global optimal projection subspace* for one-mode PSOPCA, as summarized in the following theorem.

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

solutions for $\mathbf{R}$ *and* σ^2*:*

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

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

This theorem generalizes the optimal solution of PPCA [10] 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 PSOPCA with $\mathbf{R} = \mathbf{I}_n$. If the arbitrary rotation matrix $\mathbf{V}$ needs to be identified, one can eigen-decompose $\mathbf{R}^\top\mathbf{R}$ to recover it.

For a test image $\mathbf{X}_*$, (the distribution of) its PSOPCA projection is calculated as in (2), which is 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 2 is precisely the right one-sided sample covariance used in [11], thus we have:

COROLLARY 3. *The (right) one-mode PSOPCA algorithm recovers the 2D PCA-style solution* [11] *when* $\sigma \to 0$*, up to a scaling and rotation factor.*

Corollary 3 indicates that one-mode PSOPCA provides a *probabilistic explanation* to the algorithm in [11]. The EM algorithm provides another way of calculating those projection directions.

4.2 Learning in General PSOPCA

In the general PSOPCA 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* [5], 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. Refer to [5] 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. Another (longer) page of mathematics leads to the update equations 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:

$$\mathbf{T} = c\,\sigma^2\left[\,\mathrm{tr}(\mathbf{R}^\top\mathbf{R}\mathbf{S})\mathbf{L}^\top\mathbf{L} + \sigma^2\,\mathrm{tr}(\mathbf{S})\,\mathbf{I}_r\right]^{-1}, \tag{5}$$

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

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

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

with respect to $\mathrm{vec}(\mathbf{B}_i)$, and then reshape it back to get $\mathbf{B}_i$. When σ is small (e.g., < 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 yield

$$\mathbf{B}_i = (\mathbf{L}^\top\mathbf{L})^{-1}\mathbf{L}^\top\mathbf{X}_i\mathbf{R}(\mathbf{R}^\top\mathbf{R})^{-1} = \mathbf{L}^+\mathbf{X}_i\mathbf{R}^{+\top} \tag{7}$$

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:

$$\mathbf{L} = \left[\frac{1}{N}\sum_i \mathbf{X}_i\mathbf{R}\mathbf{B}_i^\top\right]\left[\frac{1}{N}\sum_i \mathbf{B}_i\mathbf{R}^\top\mathbf{R}\mathbf{B}_i^\top + \mathrm{tr}(\mathbf{R}^\top\mathbf{R}\mathbf{S})\,\mathbf{T}\right]^{-1} \tag{8}$$

$$\mathbf{R} = \left[\frac{1}{N}\sum_i \mathbf{X}_i^\top\mathbf{L}\mathbf{B}_i\right]\left[\frac{1}{N}\sum_i \mathbf{B}_i^\top\mathbf{L}^\top\mathbf{L}\mathbf{B}_i + \mathrm{tr}(\mathbf{L}^\top\mathbf{L}\mathbf{T})\,\mathbf{S}\right]^{-1} \tag{9}$$

Finally we iterate (5)$\sim$(9) 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 PSOPCA updates when we fix $\mathbf{L} = \mathbf{I}_m$ (or $\mathbf{R} = \mathbf{I}_n$).

Unlike one-mode PSOPCA, the general PSOPCA 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 (7) (this is also how to calculate the PSOPCA projection for a test image $\mathbf{X}_*$). In this case the iterative updates (7), (8) and (9) lead to the following important result:

THEOREM 4. *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 PSOPCA algorithm* (7)$\sim$(9) *satisfies:*

$$\mathbf{R}\mathbf{R}^+ = \mathbf{U}_c(\mathbf{L})\mathbf{U}_c(\mathbf{L})^\top,\ \mathbf{L}\mathbf{L}^+ = \mathbf{V}_r(\mathbf{R})\mathbf{V}_r(\mathbf{R})^\top. \tag{10}$$

Theorem 4 builds an important connection between general PSOPCA models and the GLRAM [12]. If we write

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

and

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

by plugging in (10) at the stationary point of PSOPCA, this is exactly the stationary point of repeatedly calculating the SVD of $\mathbf{G}(\mathbf{L})$ and $\mathbf{H}(\mathbf{R})$ in GLRAM (cf. Section 2.1, also see [12]). Therefore, we see GLRAM can be viewed as a special case of general PSOPCA models when $\sigma \to 0$, which is summarized as follows:

COROLLARY 5. *Zero-noise PSOPCA models and the GLRAM [12] have the same stationary point.*

Combined with Proposition 1, we 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; 2) GLRAM gets better reconstruction than the one-sided algorithm [11]; and 3) GLRAM cannot yield better reconstruction than PCA with the same effective dimensions (as empirically verified in [12]). [12] also suggests applying a PCA to $\text{vec}(\mathbf{Z}^\top)$ after GLRAM, and it's clear from here that GLRAM + PCA is still a PCA model.

4.3 Learning in PHOPCA

The learning algorithms for PSOPCA can be extended to learning in PHOPCA. For one-mode PHOPCA where projection only happens at the jth-mode, we can "unfold" the other modes to yield a big matrix

$$\underline{\mathbf{X}}_i^{(j)} \in \mathbb{R}^{I_j \times (I_{j+1} I_{j+2} \cdots I_O I_1 I_2 \cdots I_{j-1})}$$

for each tensor $\underline{\mathbf{X}}_i$, and apply the algorithm in Section 4.1. A similar theorem like Theorem 2 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 Section 4.2. In the most interesting case where the noise level $\sigma \to 0$, the E-step (7) is extended to get the core $\underline{\mathbf{B}}_i$ as the all-mode product:

$$\underline{\mathbf{B}}_i = \underline{\mathbf{X}}_i \times_1 \mathbf{U}_1^{+\top} \times_2 \mathbf{U}_2^{+\top} \times \cdots \times_O \mathbf{U}_O^{+\top}, \qquad (11)$$

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 $\underline{\mathbf{E}}_i = \underline{\mathbf{B}}_i \times_1 \mathbf{U}_1 \times \cdots \times_{j-1} \mathbf{U}_{j-1} \times_{j+1} \mathbf{U}_{j+1} \times \cdots \times_O \mathbf{U}_O \in \mathbb{R}^{I_1 \cdots I_{j-1} P_j I_{j+1} \cdots I_O}$ is the tensor product of $\underline{\mathbf{B}}_i$ with all-mode factor loadings except $\mathbf{U}_j$. A similar result like Theorem 4 can also be proved for PHOPCA, which indicates that this iterative algorithm yields the same stationary point as the higher-order multilinear PCA [7]. 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 (11).

5. DISCUSSIONS AND EXTENSIONS

The proposed PHOPCA framework extends PPCA to model random higher-order 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 various 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 PSOPCA has time complexity $\mathcal{O}(tNmn\max(r,c))$ and space complexity $\mathcal{O}(mn)$. The higher-order PHOPCA has $\mathcal{O}(tN\prod_j I_j \max_j\{P_j\})$ as the time complexity. Here t is the number of EM iterations.

As of PPCA to PCA, PHOPCA provides additional benefits to the higher-order PCA-style algorithms:

- **Incremental learning** with newly obtained images is easy via the EM algorithm (which is crucial for real applications of higher-order PCA-style algorithms).
- **Missing data** can now be handled nicely 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 [9]) for higher-order object clustering and (local) projections.
- **Robust higher-order PCA projections** can be introduced (via, e.g., a student't model instead of normal) when there are "outlier" tensor objects.

We only briefly go into two of these extensions, with some empirical results shown in the next section.

5.1 Higher-Order 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 higher-order data. Take PSOPCA as an example. If we change the noise model for $\boldsymbol{\Upsilon}$ to be $\boldsymbol{\Upsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_0, \boldsymbol{\Phi}_0)$, where $\boldsymbol{\Sigma}_0(k,k) = \sigma_{kk} > 0$ and $\boldsymbol{\Phi}_0(\ell,\ell) = \phi_{\ell\ell} > 0$, then the noise level at a specific entry (k,ℓ) in the image can be computed as $\sigma_{kk}\phi_{\ell\ell}$. Therefore by choosing (or adapting) different (σ_{kk}) and $(\phi_{\ell\ell})$, we can make PHOPCA for or against certain regions in the images (in, e.g., applications where we want to focus on the facial area and ignore the background, or where we know a higher noise was introduced from a certain medical scanner). The learning algorithms for higher-order FA-type noise model can be easily derived (with the noise model fixed). The details are in the Appendix.

5.2 Mixture of Higher-Order Projections

PHOPCA allows us to consider MPHOPCA, a mixture of higher-order 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 PHOPCA model. This leads to a clustering structure of the data, and following the same discussions in [9] 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

6.1 Benchmark Image Data

Here we mainly illustrate the results of second-order PHOPCA, i.e., PSOPCA, since it's easier to show and compare. We first test on several image benchmark data sets (face image data sets ORL, AR and the USPS handwritten digit images). ORL is a well-known data set for face recognition. It contains the face images of 40 persons, for a total of 400 images (size 92×112). AR is a large face image dataset, of which we use a subset containing 1638 face images of 126 persons (size 101×88 after cropping and subsampling). USPS is an

Figure 1: Reconstructions of some ORL face images. row1: original images; row2: using general PSOPCA; row3: using (right) one-mode PSOPCA; row4: using PSOPCA with FA noise model. row5: Mixture of PSOPCA with 5 components. The projection dimensions are $r = c = 15$.

image data set consisting of 9298 handwritten digits of "0" through "9". We use a subset of USPS (image size 16×16).

For all the experiments we run 20 EM iterations (a typical learning curve is shown in Figure 2 left). We first illustrate some reconstructed images in Figure 1 from ORL. As expected, the general PSOPCA is better than the (right) one-mode PSOPCA, and both the FA-type noise model (row 4) and mixture of PSOPCAs (row 5) yield even better reconstructed images. For the FA-type noise model suppose we want to focus on the facial region (rows 12 to 80 and columns 40 to 100), and assume low noise level (0.0001) to the diagonals in focused rows and columns whereas putting high noise level (0.001) to the other diagonals. For the mixture model, 5 components are used with random initialization.

A thorough comparison of reconstruction errors are shown in Figure 2, where projection dimensions are (r, c) for general PSOPCA and $\lceil rc/m \rceil$ for right one-mode PSOPCA. Note that we choose the projection dimension for one-mode PSOPCA such that the compressed images have the same overall dimensionality as that in the general PSOPCA (i.e., they have the same compression ratio). The metric is the root mean square error which is the square root of the mean reconstruction error. As suggested in [12], we just show the results with $r = c$. As expected, general PSOPCA completely recovers the GLRAM results, and clearly outperforms one-mode PSOPCA. The mixture model yields better performance, but mainly in the region of smaller projection dimensions. PPCA was also tried on these data (with $r \times c = 225$ projection dimensions) and yielded the smallest reconstruction error (as expected).

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. Cardia 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 to their views. We collected patient data with various image quality from St. Francis Hospital at New York which contain largely 4 views (Figure 3 shows two of the views).

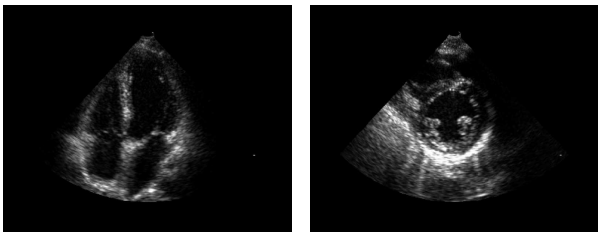

Figure 3: Two views of the heart: left, apical 4 chamber (a4c); right, parasternal short axis (psax).

Ultrasound videos of 100 patients were collected where some patients had multiple images for one view or certain views missing, resulting in 87 a4c and 83 psax clips of 480×640 image frames. PCA can be employed to extract features useful for discrimination from raw images or evaluated image features. We illustrate the potentials of PSOPCA 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 PSOPCA, general PSOPCA and mixture PSOPCA 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. PPCA simply could not run on 480×640 images so we had to down-sample the images to 200×200, which resulted in inferior classification accuracy. Moreover, PSOPCA with a focus area ($[150, 350] \times [250, 450]$), where noise level is 1e-6 in contrast to 1e-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 PSOPCA was $(r, c) = (10, 10)$, equivalent to 100 features, which generated acceptable accuracy as in Figure 4. For one-mode PSOPCA, we used $c = 1$ which generates a lot more features 480×1.

Any suitable classification methods can be then utilized to construct a classifier using these features. In our experiments, we used least squares support vector machine (LSSVM) with a tuning parameter, the regularization factor μ, which was optimized to 800 according to a 3-fold cross-validation on training data. As shown in Figure 4, mixture PSOPCA and PSOPCA with a focus area outperformed other approaches.

7. CONCLUSION

A family of PHOPCA models were proposed which provide probabilistic interpretations to many existing PCA-style algorithms. Several extensions were discussed, and some remain for future work.

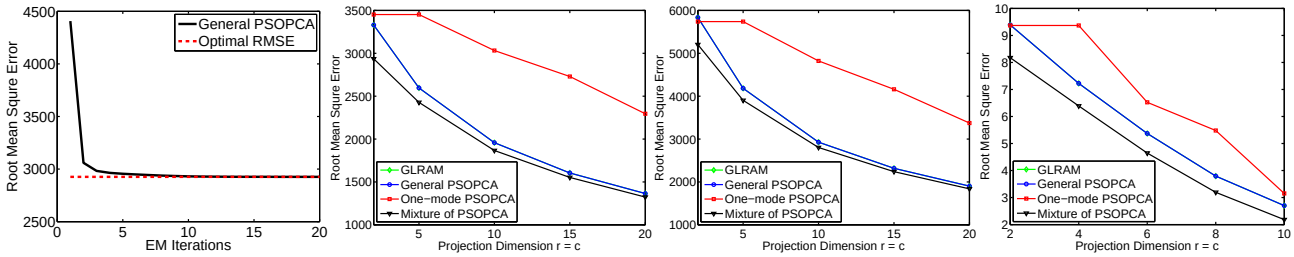

Figure 2: Typical learning curve for PSOPCA (left), and RMSE comparisons on ORL, AR and USPS (right). As expected, GLRAM yields completely the same results as General PSOPCA.

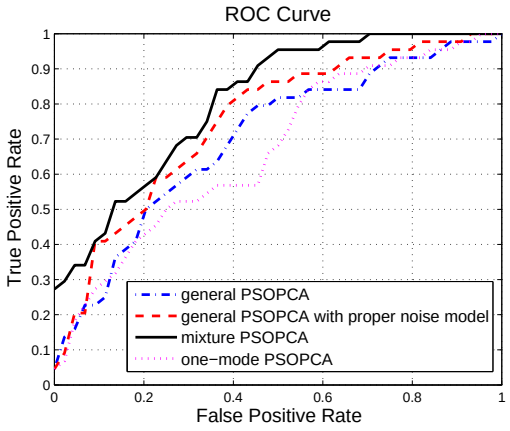

Figure 4: Test classification ROC curves of LSSVM.

8. REFERENCES

[1] C. Ding and J. Ye. 2-Dimensional singular value decomosition for 2D maps and images. In *SDM*, 2005.

[2] A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*. Chapman and Hall-CRC, 1999.

[3] K. Inoue and K. Urahama. Equivalence of non-iterative algorithms for simultaneous low rank approximations of matrices. In *CVPR*, pages 154–159, 2006.

[4] I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 2002.

[5] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[6] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. Technical Report SAND2007-6702, Sandia National Laboratories, 2007.

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

[8] S. Roweis and Z. Ghahramani. A unifying review of linear gaussian models. *Neural Computaion*, 11, 1999.

[9] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[10] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statisitical Scoiety*, B(61):611–622, 1999.

[11] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang. Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1), 2004.

[12] J. Ye. Generalized low rank approximation of matrices. *Machine Learning*, 61:167–191, 2005.

Appendix

PROOF OF PROPOSITION 1. The reformulation from (1) to PPCA is done by applying the formula $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\,\text{vec}(\mathbf{B})$ to the matrix transpose of both sides of (1). Note that by definition the matrix-variate normal in MVPP for $\mathbf{Z}$ and $\boldsymbol{\Upsilon}$ leads to the multi-variate normal in this PPCA for $\mathbf{z}$ and $\boldsymbol{\epsilon}$. □

PROOF OF THEOREM 2. We give a sketch here. We plug (2) into (3) to find the stationary point of the EM updates. At convergence we have $\mathbf{R} = \sigma^2\mathbf{GRS}(\mathbf{SR}^\top\mathbf{GRS} + \sigma^4 m\mathbf{S})^{-1}$. 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 $\mathbf{G} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ be its SVD, we have $\mathbf{D} = (\frac{1}{m}\boldsymbol{\Lambda} - \sigma^2\mathbf{I})^{\frac{1}{2}}$. Finally a similar study as that in [10] shows that $\mathbf{D}$ corresponds to the largest c eigenvalues. This gives $\mathbf{R}$. Plug this into (4) leads to the solution for σ^2. □

PROOF OF THEOREM 4. We give a sketch here. When $\sigma = 0$, $\mathbf{S}$ and $\mathbf{T}$ are zero matrices. With $\mathbf{L}$ fixed, plugging (7) into (9) yields $\mathbf{R} = \mathbf{GR}\left[\mathbf{R}^\top\mathbf{GR}\right]^{-1}\mathbf{R}^\top\mathbf{R}$ at the stationary point. Let $\mathbf{R} = \mathbf{EDF}^\top$ be its SVD, we have $\mathbf{EE}^\top\mathbf{GE} = \mathbf{GE}$. To get $\mathbf{E}$ we eigendecompose $\mathbf{E}^\top\mathbf{GE} = \mathbf{P}\boldsymbol{\Psi}\mathbf{P}^\top$ and obtain $\mathbf{EP}\boldsymbol{\Psi} = \mathbf{GEP}$. This indicates $\mathbf{EP}$ is the eigenvector of $\mathbf{G}$, and the only stable stationary solution is $\mathbf{EP} = \mathbf{U}_c$. Then we have $\mathbf{RR}^+ = \mathbf{EE}^\top = \mathbf{U}_c\mathbf{U}_c^\top$. Similarly we can obtain $\mathbf{LL}^+$ with $\mathbf{R}$ fixed. □

EM updates for general PSOPCA with FA noise model:

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

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

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

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

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