Bilinear Probabilistic Principal Component Analysis

Jianhua Zhao, Philip L. H. Yu, and James T. Kwok

***Abstract*— Probabilistic principal component analysis (PPCA) is a popular linear latent variable model for multi-layer performing dimension reduction on 1-D data in a probabilistic manner. However, when used on 2-D data such as images, PPCA suffers from the curse of dimensionality due to the subsequently large number of model parameters. To overcome this problem, we propose in this paper a novel probabilistic model on 2-D data called bilinear PPCA (BPPCA). This allows the establishment of a closer tie between BPPCA and its nonprobabilistic counterpart. Moreover, two efficient parameter estimation algorithms for fitting BPPCA are also developed. Experiments on a number of 2-D synthetic and real-world data sets show that BPPCA is more accurate than existing probabilistic and nonprobabilistic dimension reduction methods.**

***Index Terms*— 2-D data, dimension reduction, expectation maximization, principal component analysis, probabilistic model.**

I. Introduction

MANY REAL-WORLD applications involve high-dimensional data. However, the interesting structure inside the data often lies in a low-dimensional space. Dimension reduction, which aims to find a compact and meaningful data representation, is thus a useful tool for data visualization, interpretation, and analysis [1], [2].

Probabilistic modeling of dimension reduction is an important research topic in data mining, pattern recognition, machine learning, and statistics [3]. Compared with its nonprobabilistic counterparts, probabilistic models enable different sources of data uncertainty to be well studied by means of probability theory. Consequently, statistical inference and Bayesian (or variational Bayesian) methods can be performed, and missing data can be handled in a principled way. Moreover, probabilistic models can be easily extended in various ways. For example, they can be extended to probabilistic mixture models to accommodate for heterogeneous data [4], can be modified to accommodate for discrete data [5], and can also be robustified to handle outliers with the incorporation of a heavy-tailed noise distribution (such as the student t-distribution) [6].

Principal component analysis (PCA) [7] is one of the most popular techniques for dimension reduction. While the standard PCA is nonprobabilistic, Moghaddam and Pentland [8] extended it to a probabilistic framework, and Tipping and Bishop [4] derived the probabilistic PCA (PPCA) from the classical linear latent variable model. In particular, PPCA is an important development since it inherits all the advantages of a probabilistic model while including PCA as a special case.

However, PPCA, like its nonprobabilistic counterpart, is formulated for the 1-D data where observations are vectors. To apply PPCA to 2-D data where the observations are matrices (such as images), one possible solution is to first vectorize the data and then apply PPCA to the resultant 1-D data. However, vectorization destroys the natural matrix structure and may lose potentially useful local structure information among columns/rows [9]. Moreover, for 2-D data such as images, the resultant vectorized data is very high-dimensional (typically over tens of thousands of pixels) and thus suffers from the *curse of dimensionality* [10].

Instead of using vectorization, several nonprobabilistic models have been proposed in recent years that extend PCA directly for 2-D data. Examples include the generalized low-rank approximation of matrices (GLRAM) [11] and 2-DPCA [12]. This overcomes the curse of dimensionality and significantly reduces the computation cost. Moreover, GLRAM achieves a high compression ratio (i.e., much fewer space is needed for storing the data), which is particularly important for large-scale high-dimensional data. Empirically, these methods can achieve competitive or even better recognition performance than PCA, especially when the sample size is small relative to feature dimensionality. Inspired by these encouraging results, attempts have been made to formulate a probabilistic model for GLRAM so that it can enjoy similar advantages as PPCA has over PCA [10], [13], [14]. Following the classical linear latent variable model as in PPCA, they formulated the same model (which is called probabilistic second-order PCA (PSOPCA) in [13]), but with different learning algorithms.

Despite all these successes, the relationship between PSOPCA and GLRAM, unlike that between PPCA and PCA [4], has not been well established. For example, it is shown that the factor loading matrix in PPCA spans the principal subspace of the covariance matrix. However, a similar result for PSOPCA has only been obtained in the special case of zero-noise limit [14]. Moreover, parameter estimation in PPCA can be easily performed by either including the latent variables (i.e., missing data) or not, as closed-form updates are available

Manuscript received April 4, 2011; revised December 31, 2011; accepted January 1, 2012. Date of publication January 23, 2012; date of current version February 29, 2012. The work of J. H. Zhao was supported in part by the Science Fund of Yunnan Province under Grant 2010CD070 and Grant 2011Z010, and the Small Project Fund of YNUFE under Grant YC10D028 and Grant YCT1013. The work of P. L. H. Yu was supported by the Hong Kong Research Grants Councils GRF under Grant HKU 706710P. The work of J. T. Kwok was supported by the Research Grants Council of the Hong Kong Special Administrative Region under Grant 615209.

J. Zhao is with the School of Statistics and Mathematics, Yunnan University of Finance and Economics, Kunming 650221, China (e-mail: jhzhao.ynu@gmail.com).

P. L. H. Yu is with the Department of Statistics and Actuarial Science, University of Hong Kong, Hong Kong (e-mail: plhyu@hku.hk).

J. T. Kwok is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: jamesk@cse.ust.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2183006

for the constituent steps in both cases. In contrast, parameter estimation in PSOPCA [10], [13] requires the inclusion of latent variables. Otherwise, it is unclear if a closed-form update is still possible. This can be of practical significance as estimation algorithms not involving latent variables usually converge faster, as the convergence rate of the expectation maximization (EM) algorithm is determined by the portion of missing information in the complete data [15].

Motivated by PPCA, we proposed in this paper a novel probabilistic model called bilinear PPCA (BPPCA) that addresses these problems. While both BPPCA and PSOPCA are probabilistic models for 2-D data, BPPCA is more advantageous in that it bears closer relationships and similarities with PPCA. In particular: 1) BPPCA performs PPCA in the row and column directions alternately; 2) similar to PPCA, the maximum likelihood estimators (MLE) of BPPCA's model parameters span the principal subspaces of the column and row covariance matrices; and 3) as in PPCA, efficient closed-form expressions are available for the parameter update steps in BPPCA, with or without the use of latent variables.

The remainder of this paper is organized as follows. Section II reviews some related works. Section III proposes the BPPCA model and Section IV is devoted to the maximum likelihood estimation of BPPCA. Section V gives some empirical studies to compare BPPCA with some related methods. Section VI closes this paper with some concluding remarks.

In this paper, the transpose of vector/matrix is denoted by the superscript $'$, and the identity matrix by $\mathbf{I}$. Moreover, $\|\cdot\|_F$ denotes the Frobenius norm, $\mathrm{tr}(\cdot)$ is the matrix trace, $\mathrm{vec}(\cdot)$ is the vectorization operator, $\otimes$ is the Kronecker product, and $\mathcal{N}_d(\boldsymbol{\mu}, \Sigma)$ is the d-dimensional normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ.

II. Related Works

A. Minimum-Error Formulation of PCA

Let $\{\mathbf{x}_n\}_{n=1}^N$ (where each $\mathbf{x}_n \in \mathbb{R}^d$) be a set of observations. We assume that the data has been centered. In the minimum-error formulation [1], PCA finds the optimal projection matrix $\mathbf{U} \in \mathbb{R}^{d\times q}$ (where the latent dimensionality $q < d$ and the projection vectors are orthonormal $\mathbf{U}'\mathbf{U} = \mathbf{I}$) and low-dimensional representations $\mathbf{t}_n \in \mathbb{R}^q$ $(n = 1, \ldots, N)$ that minimize the mean squared error (MSE) of the reconstructed observations $(1/N)\sum_{n=1}^N \|\mathbf{x}_n - \mathbf{U}\mathbf{t}_n\|^2$. The $\mathbf{U}$ solution consists of, up to an arbitrary rotation, the q leading eigenvectors of the sample covariance matrix

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^N \mathbf{x}_n\mathbf{x}_n' \tag{1}$$

and the $\mathbf{t}_n$ solution is $\mathbf{U}'\mathbf{x}_n$.

As $\|\mathbf{x}_1 - \mathbf{x}_2\| \simeq \|\mathbf{U}\mathbf{t}_1 - \mathbf{U}\mathbf{t}_2\| = \|\mathbf{t}_1 - \mathbf{t}_2\|$, the Euclidean distance $\|\mathbf{x}_1 - \mathbf{x}_2\|$ in the d-dimensional space can be approximated by the distance $\|\mathbf{t}_1 - \mathbf{t}_2\|$ in the lower q-dimensional space. This reduces the amount of computation from $O(d)$ to $O(q)$. In addition, classification using the reduced representations usually leads to improved performance.

B. Maximum-Variance Formulation of PCA

In the maximum-variance formulation [7], PCA tries to sequentially find the projections $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_q$ (where each $\|\mathbf{u}_k\| = 1$) such that the variance of the projected data $\mathbf{u}_k'\mathbf{x}$ $(k = 1, \ldots, q)$ is maximized

$$\max_{\mathbf{u}_k} \mathrm{cov}(\mathbf{u}_k'\mathbf{x}) = \max_{\mathbf{u}_k} \mathbf{u}_k'\Sigma\mathbf{u}_k. \tag{2}$$

Here, $\Sigma = \mathrm{cov}(\mathbf{x}) = \mathbb{E}[\mathbf{x}\mathbf{x}']$ is the population covariance matrix of the (centered) observations $\mathbf{x}$. Again, the $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_q]$ solution consists of the q leading eigenvectors of Σ. If Σ is estimated by its MLE, which is the sample covariance matrix $\mathbf{S}$, then both the minimum-error and maximum-variance formulations lead to the same $\mathbf{U}$ solution (up to an arbitrary rotation).

C. Probabilistic Principal Component Analysis (PPCA)

PPCA [4] is a *restricted factor analysis* model

$$\begin{cases} \mathbf{x} = \mathbf{C}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \\ \mathbf{z} \sim \mathcal{N}_q(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}_d(\mathbf{0}, \sigma^2\mathbf{I}) \end{cases} \tag{3}$$

where $\mathbf{C} \in \mathbb{R}^{d\times q}$ is the factor loading matrix, $\mathbf{z} \in \mathbb{R}^q$ is the latent representation independent of $\boldsymbol{\epsilon}$, $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean vector, and $\sigma^2 > 0$ is the isotropic noise variance. Both the probability density distribution of $\mathbf{x}$ and the conditional probability density distribution of $\mathbf{z}$ given $\mathbf{x}$ are the multivariate normal distribution

$$\begin{aligned} \mathbf{x} &\sim \mathcal{N}_d(\boldsymbol{\mu}, \Sigma), \\ \mathbf{z}|\mathbf{x} &\sim \mathcal{N}_q\left(\mathbf{M}^{-1}\mathbf{C}'(\mathbf{x} - \boldsymbol{\mu}), \sigma^2\mathbf{M}^{-1}\right) \end{aligned} \tag{4}$$

where

$$\Sigma = \mathbf{C}\mathbf{C}' + \sigma^2\mathbf{I}, \quad \mathbf{M} = \mathbf{C}'\mathbf{C} + \sigma^2\mathbf{I}. \tag{5}$$

Given a set of observations $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$, the MLE of $\boldsymbol{\mu}$ is simply the sample mean $\bar{\mathbf{x}}$. As in Section II-A, we assume that $\bar{\mathbf{x}}$ is zero, and the sample covariance matrix is then given by (1). The MLE of $\boldsymbol{\theta} = (\mathbf{C}, \sigma^2)$ can be obtained by maximizing the log likelihood,[1] which is, up to a constant

$$\mathcal{L}(\boldsymbol{\theta}|\mathcal{X}) = -\frac{N}{2}\left\{\ln|\Sigma| + \mathrm{tr}\left(\Sigma^{-1}\mathbf{S}\right)\right\}. \tag{6}$$

Setting its derivative with respect to $\boldsymbol{\theta}$ to zero, and assuming that $\mathrm{rank}(\mathbf{S}) > q$, we obtain

$$\mathbf{C} = \mathbf{U}\left(\Lambda - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}\mathbf{V}' \tag{7}$$

$$\sigma^2 = \frac{1}{d-q}\sum_{i=q+1}^d \lambda_i \tag{8}$$

where $\mathbf{V}$ is an arbitrary orthogonal matrix, $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_q]$, and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_q)$, with $\{\mathbf{u}_i\}_{i=1}^d, \{\lambda_i\}_{i=1}^d$ $(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d)$ being the eigenvectors and eigenvalues of $\mathbf{S}$.

Alternatively, (6) can be maximized by using the well-known EM algorithm [15]. This requires the introduction of missing data and consists of an E-step and a M-step.

[1]For an alternative Bayesian framework for PPCA, interested readers are referred to [16] and [17].

E-Step: Let the missing data be $\mathcal{Z} = \{\mathbf{z}_n\}_{n=1}^N$. The complete data log likelihood is $\sum_{n=1}^N \ln\{p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n)\}$. Its expectation (up to a constant) with respect to the distribution $p(\mathcal{Z}|\mathcal{X})$ leads to the so-called Q-function

$$Q(\boldsymbol{\theta}) = -\frac{1}{2}\sum_{n=1}^{N}\left\{d\ln\sigma^2 + \sigma^{-2}\mathbb{E}\left[\|\mathbf{x}_n - \mathbf{C}\mathbf{z}_n\|^2|\mathbf{x}_n\right]\right\}$$

where the involved expectations $\mathbb{E}[\mathbf{z}_n|\mathbf{x}_n]$ and $\mathbb{E}[\mathbf{z}_n\mathbf{z}_n'|\mathbf{x}_n]$ can be easily obtained from (4) as

$$\begin{aligned}\mathbb{E}[\mathbf{z}_n|\mathbf{x}_n] &= \mathbf{M}^{-1}\mathbf{C}'\mathbf{x}_n, \\ \mathbb{E}[\mathbf{z}_n\mathbf{z}_n'|\mathbf{x}_n] &= \sigma^2\mathbf{M}^{-1} + \mathbb{E}[\mathbf{z}_n|\mathbf{x}_n]\mathbb{E}[\mathbf{z}_n'|\mathbf{x}_n].\end{aligned} \tag{9}$$

M-Step: We maximize Q with respect to $\mathbf{C}$ and σ^2, yielding

$$\widetilde{\mathbf{C}} = \sum\nolimits_{n=1}^{N}\mathbf{x}_n\mathbb{E}[\mathbf{z}_n'|\mathbf{x}_n]\left(\sum\nolimits_{n=1}^{N}\mathbb{E}[\mathbf{z}_n\mathbf{z}_n']\right)^{-1},$$

$$\widetilde{\sigma}^2 = \frac{1}{Nd}\sum\nolimits_{n=1}^{N}\left(\|\mathbf{x}_n\|^2 - \mathbb{E}[\mathbf{z}_n'|\mathbf{x}_n]\widetilde{\mathbf{C}}'\mathbf{x}_n\right).$$

Similar to the maximum-variance formulation of PCA, we may classify observations based on the expected latent representation $\mathbb{E}[\mathbf{z}|\mathbf{x}]$. Note that since the small eigenvalues of the covariance matrix Σ tend to be underestimated [18], PPCA regularizes Σ automatically by increasing its small eigenvalues to σ^2 in (5). Consequently, a regularized latent representation $\mathbb{E}[\mathbf{z}|\mathbf{x}] = (\mathbf{C}'\mathbf{C} + \sigma^2\mathbf{I})^{-1}\mathbf{C}'\mathbf{x}$ is produced in (9).

D. Minimum-Error Formulation for Bilinear Dimension Reduction

Inspired by the minimum-error formulation of 1-D PCA, several techniques have been proposed in recent years that perform dimension reduction on the 2-D data directly. Examples include the GLRAM [11] and 2-DPCA [12].

Let $\{\mathbf{X}_n\}_{n=1}^N$ (where each $\mathbf{X}_n \in \mathbb{R}^{d_c\times d_r}$) be a set of 2-D centered observations. GLRAM finds the optimal transformation matrices $\mathbf{U}_c \in \mathbb{R}^{d_c\times q_c}$, $\mathbf{U}_r \in \mathbb{R}^{d_r\times q_r}$ (where the columns are orthogonal and $q_c < d_c$, $q_r < d_r$) and low-dimensional representations $\mathbf{T}_n \in \mathbb{R}^{q_c\times q_r}$ $(n = 1,\ldots,N)$ such that the MSE

$$\frac{1}{N}\sum_{n=1}^{N}\|\mathbf{X}_n - \mathbf{U}_c\mathbf{T}_n\mathbf{U}_r'\|_F^2 \tag{10}$$

of the reconstructed observations $\{\mathbf{U}_c\mathbf{T}_n\mathbf{U}_r'\}_{n=1}^N$ is minimized. Given an initial $\mathbf{U}_r$, (10) can be minimized by iterating the following two steps until convergence.

1) $\mathbf{U}_c \leftarrow$ the q_c leading eigenvectors of

$$\mathbf{G}_c = \frac{1}{N}\sum_{n=1}^{N}\mathbf{X}_n\mathbf{U}_r\mathbf{U}_r'\mathbf{X}_n'. \tag{11}$$

2) $\mathbf{U}_r \leftarrow$ the q_r leading eigenvectors of

$$\mathbf{G}_r = \frac{1}{N}\sum_{n=1}^{N}\mathbf{X}_n'\mathbf{U}_c\mathbf{U}_c'\mathbf{X}_n. \tag{12}$$

After convergence, $\mathbf{T}_n$ is obtained as $\mathbf{U}_c'\mathbf{X}_n\mathbf{U}_r$.

On the other hand, 2-DPCA only applies a linear transformation on the right side of the data matrix. Hence, it can be viewed as a special case of GLRAM.

E. Probabilistic Extensions of GLRAM

Recently, several works attempt to formulate a probabilistic model for GLRAM so that it can enjoy similar advantages as PPCA has over PCA [10], [13], [14]. Following the classical linear latent variable model as in PPCA, they formulate the following model, which is called PSOPCA in [13]:

$$\begin{cases}\mathbf{X} = \mathbf{C}\mathbf{Z}\mathbf{R}' + \mathbf{W} + \boldsymbol{\epsilon}, \\ \mathbf{Z} \sim \mathcal{N}_{q_c,q_r}(\mathbf{0},\mathbf{I},\mathbf{I}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}_{d_c,d_r}(\mathbf{0},\sigma^2\mathbf{I},\sigma^2\mathbf{I})\end{cases} \tag{13}$$

where $\mathcal{N}_{q_c,q_r}$ and $\mathcal{N}_{d_c,d_r}$ are matrix-variate normal distributions,[2] $\mathbf{C} \in \mathbb{R}^{d_c\times q_c}$ and $\mathbf{R} \in \mathbb{R}^{d_r\times q_r}$ are the column and row factor loading matrices, respectively, $\mathbf{W} \in \mathbb{R}^{d_c\times d_r}$ is the mean matrix, and $\sigma^2 > 0$ is the noise variance. Different learning algorithms for this model are proposed in [10], [13], and [14].

As mentioned in Section I, the relationship between PSOPCA and GLRAM is not as well-established as that between PPCA and PCA. For example, for PPCA, it can be seen from (7) that the factor loading matrix $\mathbf{C}$ spans the principal subspace of the covariance matrix. In the special case of zero-noise limit, a similar result for PSOPCA is obtained in [14]. Specifically, they showed that the column and row factor loading matrices $\mathbf{C}$ and $\mathbf{R}$ span the principal subspaces of the respective covariance matrices in GLRAM. However, it is unclear how to extend this for the general noise case. Moreover, as seen in Section II-C, parameter estimation in PPCA can be efficiently performed by either including the latent variables (i.e., missing data) or not, as closed-form updates are available for the constituent steps in both cases. In contrast, parameter estimation in PSOPCA [10], [13] requires the inclusion of latent variables. Otherwise, it is unclear if a closed-form update will still be available.

III. BPPCA

A. Proposed Model

In this section, we extend PPCA in (3) to 2-D data. The proposed model, which will be called BPPCA, is defined as

$$\begin{cases}\mathbf{X} = \mathbf{C}\mathbf{Z}\mathbf{R}' + \mathbf{W} + \mathbf{C}\boldsymbol{\epsilon}_r + \boldsymbol{\epsilon}_c\mathbf{R}' + \boldsymbol{\epsilon}, \\ \mathbf{Z} \sim \mathcal{N}_{q_c,q_r}(\mathbf{0},\mathbf{I},\mathbf{I}), \quad \boldsymbol{\epsilon}_r \sim \mathcal{N}_{q_c,d_r}(\mathbf{0},\mathbf{I},\sigma_r^2\mathbf{I}), \\ \boldsymbol{\epsilon}_c \sim \mathcal{N}_{d_c,q_r}(\mathbf{0},\sigma_c^2\mathbf{I},\mathbf{I}), \boldsymbol{\epsilon} \sim \mathcal{N}_{d_c,d_r}(\mathbf{0},\sigma_c^2\mathbf{I},\sigma_r^2\mathbf{I})\end{cases} \tag{14}$$

where $\mathbf{Z}$ is the latent matrix, $\boldsymbol{\epsilon}_c \in \mathbb{R}^{d_c\times q_r}$ is the column noise, $\boldsymbol{\epsilon}_r \in \mathbb{R}^{q_c\times d_r}$ is the row noise, $\boldsymbol{\epsilon} \in \mathbb{R}^{d_c\times d_r}$ is the common noise (which are assumed to be independent of each other), $\mathbf{C} \in \mathbb{R}^{d_c\times q_c}$ and $\mathbf{R} \in \mathbb{R}^{d_r\times q_r}$ are the column and row factor loading matrices, respectively. $\sigma_c^2 > 0$ and $\sigma_r^2 > 0$ are the column and row noise variances, respectively. $\mathbf{W} \in \mathbb{R}^{d_c\times d_r}$ is the mean matrix. Obviously, when $d_r = 1$ or $d_c = 1$, the BPPCA model in (14) reduces to the PPCA model in (3). Similarly, if we remove the $\boldsymbol{\epsilon}_c$ and $\boldsymbol{\epsilon}_r$ terms, (14) reduces to the PSOPCA model in (13) when $\sigma_c = \sigma_r$. As will be seen in Section IV, the introduction of the $\boldsymbol{\epsilon}_c$ and $\boldsymbol{\epsilon}_r$ terms enables the model to have a number of interesting characteristics that are not available under PSOPCA.

[2] A review on matrix-variate normal distributions is in Appendix VI.

From (14), it is easy to obtain that

$$\begin{aligned}\mathbf{CZR}' &\sim \mathcal{N}_{d_c,d_r}(\mathbf{0}, \mathbf{CC}', \mathbf{RR}'),\\ \mathbf{C}\boldsymbol{\epsilon}_r &\sim \mathcal{N}_{d_c,d_r}(\mathbf{0}, \mathbf{CC}', \sigma_r^2\mathbf{I}),\\ \boldsymbol{\epsilon}_c\mathbf{R}' &\sim \mathcal{N}_{d_c,d_r}(\mathbf{0}, \sigma_c^2\mathbf{I}, \mathbf{RR}').\end{aligned}$$

Consequently, $\mathbf{X}$ follows the matrix-variate normal distribution $\mathcal{N}_{d_c,d_r}(\mathbf{W}, \Sigma_c, \Sigma_r)$, where

$$\Sigma_c = \mathbf{CC}' + \sigma_c^2\mathbf{I}; \quad \Sigma_r = \mathbf{RR}' + \sigma_r^2\mathbf{I}. \tag{15}$$

Thus, as in PPCA, BPPCA is characterized by a normal distribution on $\mathbf{X}$ and a low-rank covariance structure [see (5) and (15)]. Note that this can be extended to other constrained covariance structures and nonnormal distributions. Another characteristic of BPPCA is the use of a separable covariance structure via the $\mathbf{CZR}'$ term in (14). This will be studied in more detail in Section III-B.

Similar to PPCA [4], not all the BPPCA parameters can be uniquely identified. However, as a subspace learning method [19], the subspaces of interest (that are spanned by the columns of $\mathbf{C}$ and $\mathbf{R}$) can still be uniquely identified up to: 1) orthogonal rotations of the factor loading matrices, latent matrix, column and row noise matrices; and 2) scaling of the column and row factor loading matrices. Interested readers are referred to Appendix VI for details.

B. Bilinear Transformation and Separable Covariance

In the BPPCA model (14), the observed 2-D data is related to a lower-dimensional latent matrix $\mathbf{Z}$ via the transformation

$$\tilde{\mathbf{X}} \equiv \mathbf{CZR}'. \tag{16}$$

This is called a *bilinear* transformation as $\tilde{\mathbf{X}}$ is linear with respect to $\mathbf{C}$ (resp. $\mathbf{R}$) when $\mathbf{R}$ (resp. $\mathbf{C}$) is fixed. Note that a similar modeling assumption is also used in (13) for the PSOPCA model.

Proposition 1: The use of the bilinear transformation (16) is equivalent to the assumption of a *separable* (Kronecker product) covariance matrix on $\tilde{\mathbf{X}}$

$$\text{cov}(\text{vec}(\tilde{\mathbf{X}})) = \Sigma_r \otimes \Sigma_c \tag{17}$$

where $\Sigma_r \in \mathbb{R}^{d_r \times d_r}$ and $\Sigma_c \in \mathbb{R}^{d_c \times d_c}$ are the row and column covariance matrices of $\tilde{\mathbf{X}}$, respectively.

Proof: Since the covariance of $\text{vec}(\mathbf{Z})$ is $\mathbf{I}$, the covariance of $\tilde{\mathbf{X}}$ is given by (17), where $\Sigma_r = \mathbf{RR}'$ and $\Sigma_c = \mathbf{CC}'$. Conversely, if the covariance matrix of $\tilde{\mathbf{X}}$ is separable as in (17), there exist $\mathbf{C}$ and $\mathbf{R}$ such that $\Sigma_c = \mathbf{CC}'$ and $\Sigma_r = \mathbf{RR}'$. Then (16) holds with $\mathbf{Z} = \mathbf{C}^{-1}\tilde{\mathbf{X}}\mathbf{R}'^{-1}$. ∎

The separable covariance assumption has been successfully used in a variety of applications. Examples include the spatial-temporal modeling of environmental data [20], channel modeling in multiple-input multiple-output communications [21], and signal modeling of MEG/EEG data [22]. This covariance structure arises when the variables can be cross-classified by two (or, in general, three or more) vector-valued factors [23], [24]. For the 2-D data considered here, this corresponds to $\text{vec}(\tilde{\mathbf{X}}) = \mathbf{a} \otimes \mathbf{b}$, where $\mathbf{a}$ and $\mathbf{b}$ are variables in the row and column directions, respectively, and with $\text{cov}(\mathbf{a}) = \Sigma_r, \text{cov}(\mathbf{b}) = \Sigma_c$.

Obviously, a separable covariance is more restrictive than a full covariance matrix. For example, in the simplest case where $d_r = d_c = 2$, it can be shown that separability imposes the following constraints on the covariance matrix $\Sigma = [\sigma_{ij}]$ and the associated correlation matrix $\boldsymbol{\rho} = [\rho_{ij}]$ [23]:

$$\begin{aligned}&\frac{\sigma_{11}}{\sigma_{33}} = \frac{\sigma_{22}}{\sigma_{44}},\\ &\rho_{12} = \rho_{34}, \quad \rho_{13} = \rho_{24}, \quad \rho_{23} = \rho_{14}, \quad \text{and } \rho_{14} = \rho_{12}\rho_{13}.\end{aligned}$$

Despite such a restrictive covariance structure, separability can significantly reduce the number of parameters in the model (from $1/2 d_c d_r (d_c d_r + 1)$ for the full covariance to $(1/2)\,(d_c(d_c+1) + d_r(d_r+1))$ for the separable covariance[3]), leading to reduced algorithm complexity and often more accurate estimators [24]. This can be attributed to the bias-variance tradeoff [25], in which one can trade bias for lower variance, leading to better generalization. As will be seen in Section V-E, empirical results also confirm that our BPPCA model (which is based on separable covariance) outperforms PPCA (which uses a nonrestrictive covariance). Note that the use of a restrictive covariance structure is common in machine learning. For example, for linear discriminant analysis (LDA), the even more restrictive diagonal covariance assumption leads to the diagonal LDA [26], which is found to perform well on high-dimensional microarray data.

Recently, Dryden *et al.* [27] proposed a related dimension reduction technique for 2-D data called factored PCA (FPCA) which also assumes separable covariance. Indeed, it can be seen from (15) that FPCA can be regarded as a special case of BPPCA with $\sigma_c^2 \to 0, \sigma_r^2 \to 0$ and $q_c = d_c, q_r = d_r$.

C. Probabilistic Graphical Models for BPPCA

To further understand model (14), it is helpful to rewrite it as

$$\begin{cases}\mathbf{X} = \mathbf{CY}^r + \mathbf{W} + \mathbf{Y}_\epsilon^r,\\ \mathbf{Y}^r = \mathbf{ZR}' + \boldsymbol{\epsilon}_r,\\ \mathbf{Y}_\epsilon^r = \boldsymbol{\epsilon}_c\mathbf{R}' + \boldsymbol{\epsilon}\end{cases} \tag{18}$$

where $\mathbf{Y}^r \in \mathbb{R}^{q_c \times d_r}$ and $\mathbf{Y}_\epsilon^r \in \mathbb{R}^{d_c \times d_r}$ are latent matrices. This can be interpreted as a two-stage representation of BPPCA. From the projection point of view, $\mathbf{X}$ is first projected onto $\mathbf{Y}^r$ in the column direction. Then $\mathbf{Y}^r$ and residual $\mathbf{Y}_\epsilon^r$ are further projected in the row direction onto $\mathbf{Z}$ and $\boldsymbol{\epsilon}_c$, respectively. From the generative model point of view, $\mathbf{Y}^r$ and $\mathbf{Y}_\epsilon^r$ are first generated in the row direction and $\mathbf{X}$ is then generated in the column direction.

Alternatively, by introducing another two latent matrices $\mathbf{Y}^c \in \mathbb{R}^{d_c \times q_r}$ and $\mathbf{Y}_\epsilon^c \in \mathbb{R}^{d_c \times d_r}$, model (14) can be rewritten as first projecting (resp. generating) in the row (resp. column) direction and then projecting (resp. generating) in the column (resp. row) direction

$$\begin{cases}\mathbf{X} = \mathbf{Y}^c\mathbf{R}' + \mathbf{W} + \mathbf{Y}_\epsilon^c,\\ \mathbf{Y}^c = \mathbf{CZ} + \boldsymbol{\epsilon}_c,\\ \mathbf{Y}_\epsilon^c = \mathbf{C}\boldsymbol{\epsilon}_r + \boldsymbol{\epsilon}.\end{cases} \tag{19}$$

[3]For example, when $d_c = d_r = 20$, the full covariance has 80, 200 parameters while the separable covariance has only 420.

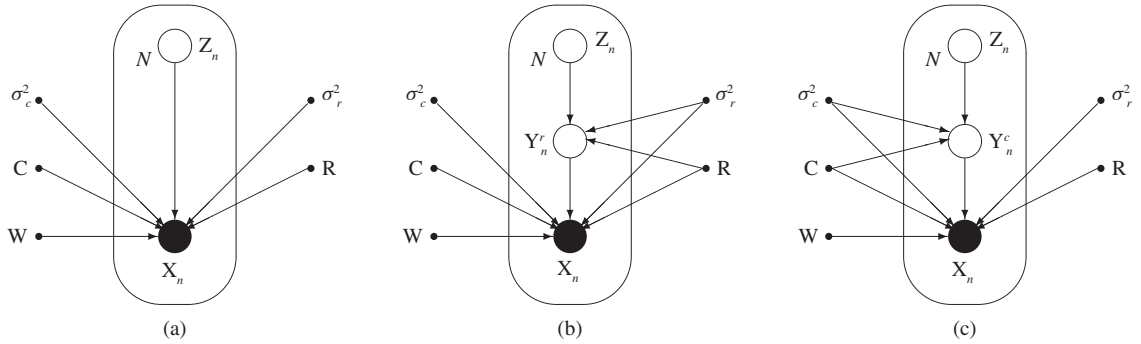

Fig. 1. Probabilistic graphical models for BPPCA. (a) Original generative model (14). (b) Two-stage generative model (18), with row followed by column. (c) Two-stage generative model (19), with column followed by row.

Fig. 1 shows the probabilistic graphical models of BPPCA corresponding to the three ways of generating $\mathbf{X}$ [(14), (18), and (19)].

D. Probability Distributions

In the following, we list the various probability distributions that can be obtained from the BPPCA model. Derivations can be found in Appendix VI

$$\begin{aligned}&\mathbf{Y}^r \sim \mathcal{N}_{q_c,d_r}(\mathbf{0}, \mathbf{I}, \Sigma_r), \quad \mathbf{Y}^r_\epsilon \sim \mathcal{N}_{d_c,d_r}(\mathbf{0}, \sigma_c^2\mathbf{I}, \Sigma_r),\\ &\mathbf{Y}^c \sim \mathcal{N}_{d_c,q_r}(\mathbf{0}, \Sigma_c, \mathbf{I}), \quad \mathbf{Y}^c_\epsilon \sim \mathcal{N}_{d_c,d_r}(\mathbf{0}, \Sigma_c, \sigma_r^2\mathbf{I}),\\ &\mathbf{X} \sim \mathcal{N}_{d_c,d_r}(\mathbf{W}, \Sigma_c, \Sigma_r) \qquad (20)\\ &\mathbf{Z}|\mathbf{Y}^r \sim \mathcal{N}_{q_c,q_r}(\mathbf{Y}^r\mathbf{R}\mathbf{M}_r^{-1}, \mathbf{I}, \sigma_r^2\mathbf{M}_r^{-1}) \qquad (21)\\ &\mathbf{Y}^r|\mathbf{X} \sim \mathcal{N}_{q_c,d_r}\left(\mathbf{M}_c^{-1}\mathbf{C}'(\mathbf{X}-\mathbf{W}), \sigma_c^2\mathbf{M}_c^{-1}, \Sigma_r\right) \qquad (22)\\ &\mathbf{Y}^c|\mathbf{X} \sim \mathcal{N}_{d_c,q_r}\left((\mathbf{X}-\mathbf{W})\mathbf{R}\mathbf{M}_r^{-1}, \Sigma_c, \sigma_r^2\mathbf{M}_r^{-1}\right) \qquad (23)\end{aligned}$$

where Σ_c and Σ_r are given by (15) and

$$\mathbf{M}_c = \mathbf{C}'\mathbf{C} + \sigma_c^2\mathbf{I}; \quad \mathbf{M}_r = \mathbf{R}'\mathbf{R} + \sigma_r^2\mathbf{I}. \qquad (24)$$

IV. Maximum Likelihood Estimation of BPPCA

In this section, we show how the BPPCA parameters are estimated from a given set of observations $\mathcal{X} = \{\mathbf{X}_n\}_{n=1}^N$. From (20), the MLE of $\mathbf{W}$ is obviously the sample mean $(1/N)\sum_{n=1}^N \mathbf{X}_n$. As in PPCA, we assume that the data has been centered. The MLE of the remaining parameters, $\boldsymbol{\theta} = (\mathbf{C}, \sigma_c^2, \mathbf{R}, \sigma_r^2)$, can be obtained by maximizing the (incomplete-data) log likelihood of the BPPCA model, which is, up to a constant

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}|\mathcal{X}) = -\frac{1}{2}\sum_{n=1}^N \{&d_r \ln|\Sigma_c| \\ &+ d_c \ln|\Sigma_r| + \operatorname{tr}\left(\Sigma_c^{-1}\mathbf{X}_n\Sigma_r^{-1}\mathbf{X}_n'\right)\Big\}. \qquad (25)\end{aligned}$$

Due to the bilinear nature of BPPCA, it is natural to develop iterative procedures for the maximization of $\mathcal{L}$. As in PPCA, it will be seen that the parameter estimation here can be easily performed by either including the latent variables (i.e., missing data) or not. Specifically, we will first present in Section IV-A a procedure based on the conditional maximization (CM) algorithm [28], which does not require the inclusion of latent variables. Then, in Section IV-B, an EM-type algorithm, which involves latent variables, will be proposed.

A. CM Algorithm

The CM algorithm is a special case of the coordinate ascent algorithm in the optimization literature [29], where the objective function [which is the incomplete-data log likelihood $\mathcal{L}$ (25) here] is maximized with respect to a subset of the variables at each iteration. In the following, we divide the parameters into two subsets, $\{\mathbf{C}, \sigma_c^2\}$ and $\{\mathbf{R}, \sigma_r^2\}$.

1) *CM-Step 1:* We maximize $\mathcal{L}$ with respect to $\mathbf{C}$ and σ_c^2, with $\{\mathbf{R}, \sigma_r^2\}$ fixed. Equation (25) is then reduced to

$$\mathcal{L}_c\left(\mathbf{C}, \sigma_c^2|\mathcal{X}\right) = -\frac{Nd_r}{2}\left\{\ln|\Sigma_c| + \operatorname{tr}\left(\Sigma_c^{-1}\mathbf{S}_c\right)\right\} \qquad (26)$$

where

$$\begin{aligned}\mathbf{S}_c &= \frac{1}{Nd_r}\sum_{n=1}^N \mathbf{X}_n\Sigma_r^{-1}\mathbf{X}_n' \qquad (27)\\ &= \frac{1}{Nd_r}\sum\nolimits_{n=1}^N\sum_{i=1}^{d_r}\mathbf{x}_{ni}\Sigma_r^{-1}\mathbf{x}_{ni}'\end{aligned}$$

is the sample covariance matrix of the columns of $\mathbf{X}_n$'s. Note that (26) is similar to (6). Hence, using the same derivation as in Section II-C, we obtain

$$\widetilde{\mathbf{C}} = \mathbf{U}_c\left(\Lambda_c - \widetilde{\sigma}_c^2\mathbf{I}\right)^{\frac{1}{2}}\mathbf{V}_c' \qquad (28)$$

$$\widetilde{\sigma}_c^2 = \frac{1}{d_c - q_c}\sum_{i=q_c+1}^{d_c}\lambda_{ci} \qquad (29)$$

where $\mathbf{U}_c, \mathbf{V}_c$, and Λ_c are defined similarly as their counterparts in Section II-C (i.e., $\mathbf{V}_c$ is an arbitrary orthogonal matrix, $\mathbf{U}_c = [\mathbf{u}_{c1}, \ldots, \mathbf{u}_{cq_c}]$ and $\Lambda_c = \operatorname{diag}(\lambda_{c1}, \ldots, \lambda_{cq_c})$, with $\{\mathbf{u}_{ci}\}_{i=1}^{d_c}, \{\lambda_{ci}\}_{i=1}^{d_c}$ $(\lambda_{c1} \geq \lambda_{c2} \geq \cdots \geq \lambda_{cd_c})$ being the eigenvectors and eigenvalues of $\mathbf{S}_c$).

2) CM-step 2: We maximize $\mathcal{L}$ with respect to $\mathbf{R}$ and σ_r^2, with $\{\mathbf{C}, \sigma_c^2\}$ fixed. This maximization is analogous to that in CM-step 1. Define the sample covariance matrix of the rows of $\mathbf{X}_n$'s as

$$\mathbf{S}_r = \frac{1}{Nd_c}\sum\nolimits_{n=1}^N \mathbf{X}_n'\widetilde{\Sigma}_c^{-1}\mathbf{X}_n. \qquad (30)$$

Then (25) becomes

$$\mathcal{L}_r(\boldsymbol{\theta}|\mathcal{X}) = -\frac{Nd_c}{2}\left\{\ln|\Sigma_r| + \operatorname{tr}\left(\Sigma_r^{-1}\mathbf{S}_r\right)\right\}$$

Algorithm 1 CM algorithm for BPPCA

Input: Data $\mathcal{X}$ and (random) initialization of $\mathbf{R}, \sigma_r^2$.

1: Compute the sample mean $\bar{\mathbf{X}}$ and center the data as $\mathbf{X}_n \leftarrow \mathbf{X}_n - \bar{\mathbf{X}}$.
2: **repeat**
3: *CM-step 1:* Compute $\mathbf{S}_c$ via (27). Update $\mathbf{C}$ and σ_c^2 via (28) and (29).
4: *CM-step 2:* Compute $\mathbf{S}_r$ via (30). Update $\mathbf{R}$ and σ_r^2 via (31) and (32).
5: **until** change of $\mathcal{L}$ is smaller than a threshold.

Output: $(\mathbf{C}, \mathbf{R}, \sigma_c^2, \sigma_r^2)$.

and the optimal solution is

$$\widetilde{\mathbf{R}} = \mathbf{U}_r \left(\Lambda_r - \widetilde{\sigma}_r^2 \mathbf{I}\right)^{\frac{1}{2}} \mathbf{V}_r' \tag{31}$$

$$\widetilde{\sigma}_r^2 = \frac{1}{d_r - q_r} \sum_{i=q_r+1}^{d} \lambda_{ri} \tag{32}$$

where $\mathbf{U}_r, \mathbf{V}_r$, and Λ_r are defined similarly as in CM-step 1 (but based on (30)).

The whole CM algorithm is shown in Algorithm 1. Since the CM algorithm is based on coordinate descent, both CM-steps 1 and 2 will increase the log likelihood $\mathcal{L}$. Moreover, it can be easily seen that the so-called "space filling" condition[4] is satisfied here. Hence, the CM algorithm is guaranteed to converge to a stationary point of $\mathcal{L}$ under the same convergence conditions as for standard EM [30].

1) Remarks: Note that the two CM-steps are equivalent to performing PPCA. Recall from (20) that $\mathbf{X}_n \sim \mathcal{N}_{d_c,d_r}(\mathbf{0}, \Sigma_c, \Sigma_r)$. In CM-step 1, with $\mathbf{R}$ and σ_r^2 fixed, $\mathbf{X}_n \Sigma_r^{-1/2} \sim \mathcal{N}_{d_c,d_r}(\mathbf{0}, \Sigma_c, \mathbf{I})$ and its columns $\{\mathbf{x}_{ni} \Sigma_r^{-1/2}\}_{i=1,\ldots,d_r, n=1,\ldots,N}$ are i.i.d. and follow $\mathcal{N}_{d_c}(\mathbf{0}, \Sigma_c)$. The covariance of these Nd_r transformed observations is $\mathbf{S}_c$ in (27). Thus, CM-step 1 performs PPCA on these transformed observations. Similarly, CM-step 2 performs PPCA on the transformed rows of $\mathbf{X}_n$ (Nd_c i.i.d. transformed observations). On the other hand, PSOPCA fails to provide such an important connection.

Moreover, similar to PPCA, (28) and (31) show that the MLE of the factor loading matrices $\mathbf{C}$ and $\mathbf{R}$ are principal subspaces of the column and row covariance matrices $\mathbf{S}_c$ and $\mathbf{S}_r$, respectively (up to scaling and rotation).

Comparing (11), (12), (27), and (30), we can find that $\mathbf{G}_c$ and $\mathbf{G}_r$ are different from $\mathbf{S}_c$ and $\mathbf{S}_r$. Therefore, the principal components by BPPCA and GLRAM are in general different.

2) Computational Complexity: The most expensive computations are on the formations of $\mathbf{S}_c$ in (27), $\mathbf{S}_r$ in (30) and their eigen-decompositions. Using

$$\Sigma_c^{-1} = \frac{1}{\sigma_c^2}(\mathbf{I} - \mathbf{C}\mathbf{M}_c^{-1}\mathbf{C}'). \tag{33}$$

[4]Loosely speaking, this means unconstrained maximization is allowed over the whole parameter space [30].

$\mathbf{S}_c$ can be computed as

$$\mathbf{S}_c = \frac{1}{Nd_r\sigma_c^2} \sum_n \left[\mathbf{X}_n\mathbf{X}_n' - (\mathbf{X}_n\mathbf{C})\mathbf{M}_c^{-1}(\mathbf{X}_n\mathbf{C})'\right].$$

Computing $\mathbf{X}_n\mathbf{X}_n'$ and $\mathbf{X}_n\mathbf{C}$ take $O(d_c^2 d_r)$ and $O(d_c d_r q_c)$ time, respectively. Given $\mathbf{X}_n\mathbf{C}$, computing $(\mathbf{X}_n\mathbf{C})\mathbf{M}_c^{-1}(\mathbf{X}_n\mathbf{C})'$ takes $O(d_c^2 q_c)$ time. Let t be the number of CM iterations. The total cost of forming all the $\mathbf{S}_c$'s is $O(Nd_c^2 d_r) + O(Nt(d_c d_r q_c + d_c^2 q_c))$. Similarly, the cost of computing all the $\mathbf{S}_r$'s is $O(Nd_r^2 d_c) + O(Nt(d_r d_c q_r + d_r^2 q_r))$. Eigen-decompositions of $\mathbf{S}_c$ and $\mathbf{S}_r$ take $O(td_c^3)$ and $O(td_r^3)$, respectively. Hence, the total cost is $O(N[d_c d_r(d_c + d_r)]) + O(Nt[(d_c + d_r)^2 \max(q_c, q_r)]) + O(t[d_c^3 + d_r^3])$. This is similar to that of GLRAM [11] except for the extra first term.

B. Alternating Expectation Conditional Maximization (AECM) Algorithm

In this section, we fit the BPPCA model by an EM-type algorithm called AECM algorithm [31]. Compared to the CM algorithm developed in Section IV-A, EM-type algorithms often enjoy lower computation complexity [14], though their convergence can be slower due to the inclusion of missing information [15].

The AECM algorithm is a flexible and powerful generalization of the standard EM [31]. It is well-known that EM performs an E-step to obtain the so-called Q function followed by a M-step to maximize Q with respect to all parameters. In some cases, the M-step in EM is difficult to solve while it is possible to sequentially and conditionally maximize Q (CMQ) with respect to subsets of parameters. This yields the ECM algorithm [30] that replaces the M-step by a sequence of CMQ steps. In some cases, instead of maximizing Q, some CMQ steps can be performed through less data augmentation with the advantage of faster convergence. This leads to the AECM algorithm that replaces the E-step by several E-steps. The salient feature of AECM is that the augmented complete data is allowed to vary between E-steps yet convergence is guaranteed [31]. A specific application of ECM and AECM to mixtures of factor analyzers can be found in [32].

The AECM algorithm for BPPCA consists of two cycles, each with its own E-step and CM-step. As in Section IV-A, we divide the parameters into the two subsets $\boldsymbol{\theta}_1 = (\mathbf{C}, \sigma_c^2)$ and $\boldsymbol{\theta}_2 = (\mathbf{R}, \sigma_r^2)$.

1) In cycle 1, its E-step treats $(\mathcal{X}, \mathcal{Y}^r) = \{\mathbf{X}_n, \mathbf{Y}_n^r\}_{n=1}^N$ as the complete data, which is then maximized with respect to $\boldsymbol{\theta}_1$ (given $\boldsymbol{\theta}_2$) in its CM-step.
E-Step: The complete data log likelihood is

$$\mathcal{L}_{com,c}(\boldsymbol{\theta}_1 | \mathcal{X}, \mathcal{Y}^r) = \sum_{n=1}^{N} \ln\left\{p(\mathbf{X}_n|\mathbf{Y}_n^r)\, p(\mathbf{Y}_n^r)\right\}.$$

Given $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, we compute the expected $\mathcal{L}_{com,c}$ (up to a constant) with respect to the distribution $p(\mathcal{Y}^r|\mathcal{X}, \boldsymbol{\theta})$

$$Q_c(\boldsymbol{\theta}_1) = -\frac{1}{2} \sum_{n=1}^{N} \Big\{ d_r d_c \ln \sigma_c^2 + \sigma_c^{-2} \mathrm{tr}\{\mathbb{E}[(\mathbf{X}_n - \mathbf{C}\mathbf{Y}_n^r)\Sigma_r^{-1}(\mathbf{X}_n - \mathbf{C}\mathbf{Y}_n^r)'|\mathbf{X}_n]\}\Big\}.$$

From (22), it is easy to obtain the required expectations

$$\mathbb{E}[\mathbf{Y}_n^r|\mathbf{X}_n] = \mathbf{M}_c^{-1}\mathbf{C}'\mathbf{X}_n \tag{34}$$

and

$$\mathbb{E}[\mathbf{Y}_n^r \Sigma_r^{-1}\mathbf{Y}_n^{r\prime}|\mathbf{X}_n] = d_r\sigma_c^2\mathbf{M}_c^{-1} + \mathbb{E}[\mathbf{Y}_n^r|\mathbf{X}_n]\Sigma_r^{-1}\mathbb{E}[\mathbf{Y}_n^{r\prime}|\mathbf{X}_n]. \tag{35}$$

CM-Step: Given $\boldsymbol{\theta}_2$, we maximize Q_c with respect to $\boldsymbol{\theta}_1$ and obtain

$$\widetilde{\mathbf{C}} = \sum_{n=1}^{N}\mathbf{X}_n\Sigma_r^{-1}\mathbb{E}[\mathbf{Y}_n^{r\prime}|\mathbf{X}_n] \cdot\left(\sum_{n=1}^{N}\mathbb{E}[\mathbf{Y}_n^r\Sigma_r^{-1}\mathbf{Y}_n^{r\prime}|\mathbf{X}_n]\right)^{-1} \tag{36}$$

$$\widetilde{\sigma}_c^2 = \frac{1}{Nd_rd_c}\mathrm{tr}\left\{\sum_{n=1}^{N}\mathbf{X}_n\Sigma_r^{-1}\mathbf{X}_n' - \mathbf{X}_n\Sigma_r^{-1}\mathbb{E}[\mathbf{Y}_n^{r\prime}|\mathbf{X}_n]\widetilde{\mathbf{C}}'\right\}. \tag{37}$$

2) In cycle 2, its E-step treats $(\mathcal{X}_n, \mathcal{Y}_n^c) = \{\mathbf{X}_n, \mathbf{Y}_n^c\}_{n=1}^N$ as the complete data, which is then maximized with respect to $\boldsymbol{\theta}_2$ (given $\boldsymbol{\theta}_1$) in its CM-step.

E-Step: The complete data log likelihood is

$$\mathcal{L}_{com,r}(\boldsymbol{\theta}_2|\mathcal{X},\mathcal{Y}^c) = \sum_{n=1}^{N}\ln\{p\left(\mathbf{X}_n|\mathbf{Y}_n^c\right)p(\mathbf{Y}_n^c)\}.$$

Given the updated $\boldsymbol{\theta}_1$, we compute the expected $\mathcal{L}_{\mathrm{com},r}$ with respect to the distribution $p(\mathcal{Y}^c|\mathcal{X},\tilde{\boldsymbol{\theta}}_1,\boldsymbol{\theta}_2)$, up to a constant, as

$$Q_r(\boldsymbol{\theta}_2) = -\frac{1}{2}\sum_{n=1}^{N}\Big\{d_rd_c\ln\sigma_r^2 + \sigma_r^{-2}\mathrm{tr}\{\mathbb{E}[(\mathbf{X}_n-\mathbf{Y}^c\mathbf{R}')'\widetilde{\Sigma}_c^{-1}(\mathbf{X}_n-\mathbf{Y}^c\mathbf{R}')|\mathbf{X}_n]\}\Big\}.$$

From (23), the required expectations can be obtained as

$$\mathbb{E}[\mathbf{Y}_n^c|\mathbf{X}_n] = \mathbf{X}_n\mathbf{R}\mathbf{M}_r^{-1} \tag{38}$$

and

$$\mathbb{E}[\mathbf{Y}_n^{c\prime}\widetilde{\Sigma}_c^{-1}\mathbf{Y}_n^c|\mathbf{X}_n] = d_c\sigma_r^2\mathbf{M}_r^{-1} + \mathbb{E}[\mathbf{Y}_n^{c\prime}|\mathbf{X}_n]\widetilde{\Sigma}_c^{-1}\mathbb{E}[\mathbf{Y}_n^c|\mathbf{X}_n]. \tag{39}$$

CM-Step: Given $\tilde{\boldsymbol{\theta}}_1$, we maximize Q_r with respect to $\boldsymbol{\theta}_2$ and obtain

$$\widetilde{\mathbf{R}} = \sum_{n=1}^{N}\mathbf{X}_n'\widetilde{\Sigma}_c^{-1}\mathbb{E}[\mathbf{Y}_n^c|\mathbf{X}_n] \cdot\left(\sum_{n=1}^{N}\mathbb{E}[\mathbf{Y}_n^{c\prime}\widetilde{\Sigma}_c^{-1}\mathbf{Y}_n^c|\mathbf{X}_n]\right)^{-1} \tag{40}$$

$$\widetilde{\sigma}_r^2 = \frac{1}{Nd_rd_c}\mathrm{tr}\left\{\sum_{n=1}^{N}\mathbf{X}_n'\widetilde{\Sigma}_c^{-1}\mathbf{X}_n - \mathbf{X}_n'\widetilde{\Sigma}_c^{-1}\mathbb{E}[\mathbf{Y}_n^c|\mathbf{X}_n]\widetilde{\mathbf{R}}'\right\}. \tag{41}$$

The whole algorithm is summarized in Algorithm 2. It can be observed that cycles 1 and 2 are guaranteed to increase the log likelihood $\mathcal{L}$ of BPPCA. Under standard regularity conditions and the space-filling condition, the AECM algorithm is also guaranteed to converge to a stationary point of $\mathcal{L}$ [31].

Algorithm 2 AECM algorithm for BPPCA

Input: Data $\mathcal{X}$ and (random) initialization of $(\mathbf{C}, \mathbf{R}, \sigma_c^2, \sigma_r^2)$.

1: Compute the sample mean $\bar{\mathbf{X}}$ and center the data as $\mathbf{X}_n \leftarrow \mathbf{X}_n - \bar{\mathbf{X}}$.
2: **repeat**
3: *E-step of cycle 1*: Compute the conditional expectations $\mathbb{E}[\mathbf{Y}_n^r|\mathbf{X}_n]$ and $\mathbb{E}[\mathbf{Y}_n^r\Sigma_r^{-1}\mathbf{Y}_n^{r\prime}|\mathbf{X}_n]$ via (34) and (35).
4: *CM-step of cycle 1*: Update $\mathbf{C}$ and σ_c^2 via (36) and (37).
5: *E-step of cycle 2*: Compute the conditional expectations $\mathbb{E}[\mathbf{Y}_n^c|\mathbf{X}_n]$ and $\mathbb{E}[\mathbf{Y}_n^{c\prime}\widetilde{\Sigma}_c^{-1}\mathbf{Y}_n^c|\mathbf{X}_n]$ via (38) and (39).
6: *CM-step of cycle 2*: Update $\mathbf{R}$ and σ_r^2 via (40) and (41).
7: **until** change of $\mathcal{L}$ is smaller than a threshold.

Output: $(\mathbf{C}, \mathbf{R}, \sigma_c^2, \sigma_r^2)$.

1) Computational Complexity: The most expensive computations are on the formations of matrices $\sum_{n=1}^N\mathbf{X}_n\Sigma_r^{-1}\mathbb{E}[\mathbf{Y}_n^{r\prime}|\mathbf{X}_n]$ in (28) and $\sum_{n=1}^N\mathbf{X}_n'\widetilde{\Sigma}_c^{-1}\mathbb{E}[\mathbf{Y}_n^c|\mathbf{X}_n]$ in (31). The cost of $\mathbb{E}[\mathbf{Y}_n^{r\prime}|\mathbf{X}_n]$ is $O(d_cd_rq_c)$. Using (33), computation of $\mathbf{X}_n\Sigma_r^{-1}\mathbb{E}[\mathbf{Y}_n^{r\prime}|\mathbf{X}_n]$ can be reduced to $O(d_cd_r(q_c+q_r))$. Hence, the total cost of AECM is $O(Ntd_cd_r(q_c+q_r))$. Note that its per-iteration complexity is typically lower than that of the CM algorithm, especially when one or both data dimensionalities (d_c and d_r) is high.

C. Compression and Reconstruction

In this section, we compare the compressed representations and reconstructions under PPCA [4] and the proposed BPPCA. The key difference is that the operators in PPCA are linear while those in BPPCA are *bilinear*.

In the following, we let $\hat{\boldsymbol{\theta}}$ be the MLE of $\boldsymbol{\theta}$, $\hat{\mathbf{x}}$ and $\widehat{\mathbf{X}}$ be the reconstructed values of $\mathbf{x}$ and $\mathbf{X}$, respectively.

1) PPCA:

a) *Compression*: Given an observation $\mathbf{x}$, we take $\mathbb{E}[\mathbf{z}|\mathbf{x}]$ in (9) in the low-dimensional latent space as the compressed representation.

b) *Linear reconstruction*: Given the compressed representation $\mathbb{E}[\mathbf{z}|\mathbf{x}]$, we can reconstruct $\hat{\mathbf{x}} = \widehat{\mathbf{C}}\mathbb{E}[\mathbf{z}|\mathbf{x}]+\widehat{\boldsymbol{\mu}}$ from (3). Using (9), $\hat{\mathbf{x}} - \widehat{\boldsymbol{\mu}} = \widehat{\mathbf{C}}\widehat{\mathbf{M}}^{-1}\widehat{\mathbf{C}}'(\mathbf{x}-\widehat{\boldsymbol{\mu}})$. In general, $\widehat{\mathbf{C}}\widehat{\mathbf{M}}^{-1}\widehat{\mathbf{C}}'$ is not a projection matrix [25], except when $\sigma^2 \to 0$.

c) *Orthogonal linear reconstruction*: We can also reconstruct as $\hat{\mathbf{x}}_{orth} = \widehat{\mathbf{C}}(\widehat{\mathbf{C}}'\widehat{\mathbf{C}})^{-1}\widehat{\mathbf{M}}\mathbb{E}[\mathbf{z}|\mathbf{x}] + \widehat{\boldsymbol{\mu}}$ [4]. Using (9), we have $\hat{\mathbf{x}}_{orth} - \widehat{\boldsymbol{\mu}} = \widehat{\mathbf{C}}(\widehat{\mathbf{C}}'\widehat{\mathbf{C}})^{-1}\widehat{\mathbf{C}}'(\mathbf{x}-\widehat{\boldsymbol{\mu}})$, in which $\widehat{\mathbf{C}}(\widehat{\mathbf{C}}'\widehat{\mathbf{C}})^{-1}\widehat{\mathbf{C}}'$ is a projection matrix [4].

2) BPPCA:

a) *Compression*: Similar to PPCA, we take $\mathbb{E}[\mathbf{Z}|\mathbf{X}]$ as the compressed representation. Using (21) and (22), this can be computed as

$$\mathbb{E}[\mathbf{Z}|\mathbf{X}] = \mathbb{E}[\mathbb{E}[\mathbf{Z}|\mathbf{Y}^r]|\mathbf{X}] = \widehat{\mathbf{M}}_c^{-1}\widehat{\mathbf{C}}'(\mathbf{X}-\widehat{\mathbf{W}})\widehat{\mathbf{R}}\widehat{\mathbf{M}}_r^{-1} \tag{42}$$

where the inner expectation is with respect to the distribution $p(\mathbf{Z}|\mathbf{Y}^r)$ and the outer one is with respect to $p(\mathbf{Y}^r|\mathbf{X})$.

b) *Bilinear reconstruction*: Given the compressed representation $\mathbb{E}[\mathbf{Z}|\mathbf{X}]$, we can reconstruct $\mathbf{X}$ from (14) as

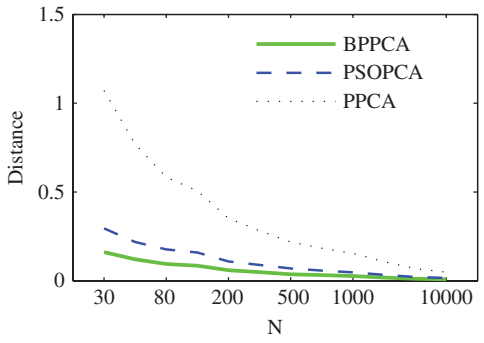

Fig. 2. Arc length distance between the estimated and true principal subspaces at different sample sizes.

$\widehat{\mathbf{X}} = \widehat{\mathbf{C}}\mathbb{E}[\mathbf{Z}|\mathbf{X}]\widehat{\mathbf{R}}' + \widehat{\mathbf{W}}$. Using (42), we have $\widehat{\mathbf{X}} - \widehat{\mathbf{W}} = \widehat{\mathbf{C}}\widehat{\mathbf{M}}_c^{-1}\widehat{\mathbf{C}}'(\mathbf{X} - \widehat{\mathbf{W}})\widehat{\mathbf{R}}\widehat{\mathbf{M}}_r^{-1}\widehat{\mathbf{R}}$. In general, this is not a biorthogonal projection since $\widehat{\mathbf{C}}\widehat{\mathbf{M}}_c^{-1}\widehat{\mathbf{C}}'$ and $\widehat{\mathbf{R}}\widehat{\mathbf{M}}_r^{-1}\widehat{\mathbf{R}}$ are not projection matrices, except when $\sigma_c^2 \to 0$ and $\sigma_r^2 \to 0$.

c) *Biorthogonal bilinear reconstruction*: We can also reconstruct $\mathbf{X}$ as $\widehat{\mathbf{X}}_{orth} = \widehat{\mathbf{C}}(\widehat{\mathbf{C}}'\widehat{\mathbf{C}})^{-1}\widehat{\mathbf{M}}_c\mathbb{E}[\mathbf{Z}|\mathbf{X}]\widehat{\mathbf{M}}_r(\widehat{\mathbf{R}}'\widehat{\mathbf{R}})^{-1}\widehat{\mathbf{R}} + \widehat{\mathbf{W}}$. Using (42), we have $\widehat{\mathbf{X}}_{orth} - \widehat{\mathbf{W}} = \widehat{\mathbf{C}}(\widehat{\mathbf{C}}'\widehat{\mathbf{C}})^{-1}\widehat{\mathbf{C}}'(\mathbf{X} - \widehat{\mathbf{W}})\widehat{\mathbf{R}}(\widehat{\mathbf{R}}'\widehat{\mathbf{R}})^{-1}\widehat{\mathbf{R}}'$, which is a biorthogonal projection since $\widehat{\mathbf{C}}(\widehat{\mathbf{C}}'\widehat{\mathbf{C}})^{-1}\widehat{\mathbf{C}}'$ and $\widehat{\mathbf{R}}(\widehat{\mathbf{R}}'\widehat{\mathbf{R}})^{-1}\widehat{\mathbf{R}}'$ are projection matrices.

V. Experiments

In this section, we perform experiments on a number of synthetic and real-world data sets. Unless otherwise stated, the CM algorithm with fast convergence (Section IV-A) is used for BPPCA. For BPPCA and GLRAM, iteration is stopped when the relative change in the objective ($|1 - \mathcal{L}^{(t)}/\mathcal{L}^{(t+1)}|$) is smaller than a threshold tol ($= 10^{-5}$ in the experiments) or the number of iterations exceeds a certain maximum $t_{\max}$ ($= 20$). For PSOPCA, we use the variational EM learning algorithm in [14] for the general noise case and follow their experimental setting to set $t_{\max} = 20$.

A. Accuracies of the Estimators

In this experiment, we sample a 2-D synthetic data set from a 10×10-D matrix-variate normal distribution. The column and row covariance matrices have different eigenvalues $(5, 4.5, 4, 1, \ldots, 1$ and $10, 9, 8, 2, \ldots, 2$, respectively, of which the first three are dominant), and their leading principal components are the same ($[1/\sqrt{2}, -1/\sqrt{2}, 0, \ldots, 0]'$, $[0, 0, 1/\sqrt{2}, -1/\sqrt{2}, 0, \ldots, 0]'$ and $[0, 0, 0, 0, 1/\sqrt{2}, -1/\sqrt{2}, 0, \ldots, 0]'$).

We compare the accuracies of the following methods in estimating the dominant principal subspace of the data.

1) BPPCA, with $q_c = q_r = 3$;
2) PSOPCA [14], with $q_c = q_r = 3$; and
3) PPCA [4] on the vectorized 1-D data, with $q = q_c q_r = 9$.

The subspaces of BPPCA and PSOPCA are spanned by the columns of $\mathbf{R} \otimes \mathbf{C}$ in (14) and (13), respectively. The performance criterion is the arc length distance between the estimated subspace and the true one [33]. Let $\hat{\mathbf{P}}, \mathbf{P} \in \mathbb{R}^{100\times 9}$ be the orthogonal bases of the two subspaces, respectively. The arc length between them is defined as $\|\boldsymbol{\theta}\|_2$, where $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_9]'$, with $\{\cos(\theta_i)\}_{i=1}^9$ being the singular values of $\hat{\mathbf{P}}'\mathbf{P}$. To reduce statistical variability, results for all the methods are averaged over 50 repetitions.

Fig. 2 shows the arc length distances obtained at different sample sizes (N). It can be observed that: 1) as N increases, the principal subspaces obtained by all three methods all converge to the true one; and 2) with limited sample size, BPPCA performs best, which is then followed by PSOPCA, and (as expected) PPCA is the worst.

TABLE I
Negative Log-Likelihood Values and Arc Length Distances Obtained for Ten Different Initializations of CM

Trial	Iteration 1	2	3	4	Distance
1	76714.2	60168.4	60168.0	60168.0	0
2	71073.9	60168.2	60168.0	60168.0	7.74e–08
3	73446.2	60168.2	60168.0	60168.0	7.15e–08
4	72262.2	60168.2	60168.0	60168.0	1.00e–07
5	72716.3	60168.2	60168.0	60168.0	1.17e–07
6	72890.9	60168.2	60168.0	60168.0	1.50e–07
7	73390.6	60168.2	60168.0	60168.0	8.94e–08
8	70523.1	60168.2	60168.0	60168.0	1.10e–07
9	72355.2	60168.2	60168.0	60168.0	1.29e–07
10	72554.5	60168.2	60168.0	60168.0	1.30e–07

B. Sensitivity to Initialization

Recall that random initialization is used in the CM and AECM algorithms (Algorithms 1 and 2). Our experience suggests that such a simple scheme works well in practice, and almost identical stationary points of the likelihood are obtained with different random initializations. To illustrate this, we report in the following an experiment on sensitive analysis, using the data set in Section V-A (with sample size $N = 200$). The setup follows that used for the GLRAM in [11]. In the first trial for CM, $\mathbf{R}$ is initialized as $[\mathbf{I}, \mathbf{0}']'$ and σ_r^2 as 0.01. For the other nine trials of CM and all ten trials of AECM, the initializations are random. To measure the differences among solutions obtained with different initializations, we measure the arc length distance between the principal subspace for the solution obtained with CM's first trial and those from the other random initializations.

Results for CM and AECM are shown in Tables I and II, respectively. As can be seen, different initializations converge to the same log-likelihood value and almost identical principal subspace (up to rotation).

C. Convergence of CM and AECM

In this experiment, we compare the convergence speeds of the CM algorithm (Section IV-A) and AECM algorithm (Section IV-B) for BPPCA. We use the same data set (with

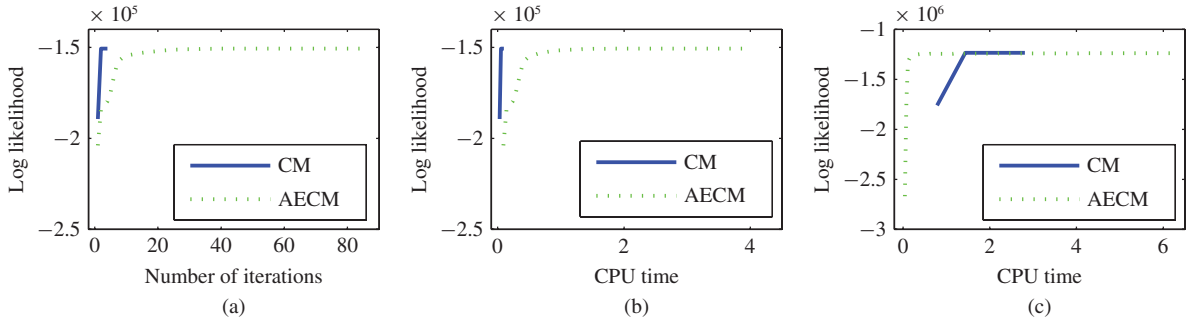

Fig. 3. Changes in log likelihood $\mathcal{L}$ for the CM (solid) and AECM (dotted) algorithms (a) with the number of iterations on the first synthetic data set, (b) with CPU time on the first synthetic data set, and (c) with CPU time on the second high-dimensional synthetic data set.

TABLE II
NEGATIVE LOG-LIKELIHOOD VALUES AND ARC LENGTH DISTANCES OBTAINED FOR TEN DIFFERENT INITIALIZATIONS OF AECM

Trial	Iteration 1	3	50	150	Distance
1	10360469819.2	72888.0	60169.5	60168.0	1.22e–07
2	10617967930.8	77084.8	60169.0	60168.0	1.46e–07
3	10912126971.6	78603.7	60172.2	60168.0	1.44e–07
4	9460828603.7	73512.5	60169.8	60168.0	1.44e–07
5	10091592873.6	74570.0	60168.5	60168.0	1.53e–07
6	11040349303.6	75792.0	60168.0	60168.0	1.49e–07
7	11432559741.1	72319.0	60168.7	60168.0	1.67e–07
8	10429865860.2	75821.6	60168.9	60168.0	1.69e–07
9	10818886067.2	73340.7	60168.3	60168.0	1.37e–07
10	10995998085.1	73546.0	60168.9	60168.0	1.59e–07

sample size $N = 500$) from Section V-A. Again, we fit the data with $q_c = q_r = 3$. For demonstration purpose, we set $tol = 10^{-8}$. Moreover, $\mathbf{C}$ or $\mathbf{R}$ are initialized randomly and σ_c^2, σ_r^2 are set to 0.01.

Fig. 3(a) plots the evolution of the log likelihood value versus the number of iterations. It can be observed that CM converges in a few iterations while AECM requires around 60 iterations to achieve comparable likelihood value. This is consistent with the theoretical result that the inclusion of missing information may yield slower convergence [15].

However, as the per-iteration complexity of AECM is lower than that of CM, it is interesting to investigate whether AECM could actually be more efficient. Fig. 3(b) plots the evolution of their log likelihood values versus CPU time. It can be observed that CM is indeed more efficient than AECM on this data set. In general, this is to be expected when the data dimensionality is not high.

When one/both of the data dimensionalities (d_c and d_r) is high, AECM can become more efficient, as is demonstrated in the following experiment. We sample a data set (with sample size 50) from a 500×20-D matrix-variate normal distribution with latent dimensions $q_c = q_r = 3$. Again, we fit the data with the true latent dimensions. Fig. 3(c) plots the evolution of their log likelihood values versus CPU time. It can be observed that AECM is more efficient than CM on this high-dimensional data set.

TABLE III
METHODS USED ON THE FACE DATA SETS

Method	Compressed representation
BPPCA	$\mathbb{E}[\mathbf{Z}\|\mathbf{X}]$ in (42)
PSOPCA [10]	$\mathbb{E}[\mathbf{Z}\|\mathbf{X}]$
PPCA [4]	$\mathbb{E}[\mathbf{z}\|\mathbf{x}]$ in (9)
GLRAM [11]	$\mathbf{U}_c'\mathbf{X}\mathbf{U}_r$
PCA [11]	$\mathbf{U}'\text{vec}(\mathbf{X})$
FPCA [27]	$\mathbf{U}_c'\mathbf{X}\mathbf{U}_r$
2-DPCA [12]	$\mathbf{X}\mathbf{U}_r$

D. Classification Performance on Face Data Sets

In this section, we perform face recognition experiments on two real-world image data sets.

1) XM2VTS[5], which contains images for 295 individuals. Each individual has eight images taken over a period of four months. The image size is 51×55.
2) AR, which contains 126 individuals. Each individual has 26 images. As in [34], we use a subset containing 100 individuals (50 men and 50 women), and each person has 14 nonoccluded images with variations in expression and illumination. The image size is 100×100.

The data is randomly split into training and test sets, such that each class has two, three, or four training samples. The classification error rate, averaged over 20 such repetitions, will be reported. Table III lists the dimension reduction methods to be compared and their corresponding representations in the reduced-dimensional space. After the compressed representations by each method are obtained, the one-nearest-neighbor classifier is then used to obtain the error rates. For all these methods, all possible dimensionalities of the compressed representation are tried and with the best results reported.

Table IV shows the error rates obtained by the various methods. The following can be seen.

1) BPPCA and PPCA substantially outperforms GLRAM, PCA and FPCA.
2) BPPCA is better than PPCA, and this can be attributed to the use of the underlying 2-D data structure.
3) BPPCA is significantly better than PSOPCA. This indicates that the features obtained by BPPCA are significantly superior than those by PSOPCA.

[5]Available from http://www.face-rec.org/databases/.

TABLE IV
AVERAGED ERROR RATES (MEAN±STD %) OBTAINED BY THE VARIOUS METHODS ON THE FACE DATA SETS. THE METHOD THAT IS STATISTICALLY SIGNIFICANTLY BETTER (WITH A P-VALUE OF 0.05 USING THE TWO-SAMPLE ONE-TAILED t-TEST) THAN THE OTHER METHODS IS MARKED ⋆

Data set	Method	Number of training images per individual 2	3	4
XM2VTS	BPPCA	⋆**19.8**±2.7	⋆**15.3**±2.2	⋆**11.3**±1.8
	PSOPCA	31.8±2.8	25.1±2.2	19.9±2.1
	PPCA	25.5±3.2	19.0±2.1	14.5±2.1
	FPCA	26.2±3.0	20.9±2.2	16.3±2.1
	GLRAM	26.5±3.1	21.1±2.1	16.5±2.1
	PCA	26.6±3.1	21.2±2.0	16.6±2.0
	2DPCA	26.7±3.1	21.1±2.0	16.6±2.0
AR	BPPCA	⋆**36.1**±5.8	⋆**25.3**±6.0	**24.1**±6.2
	PSOPCA	44.0±7.5	29.2±7.3	27.4±6.8
	PPCA	44.6±10.1	27.3±5.4	24.5±5.9
	FPCA	58.8±11.6	42.0±4.5	41.8±10.3
	GLRAM	58.8±11.6	42.0±4.5	41.8±10.3
	PCA	58.9±11.6	42.2±4.4	41.9±10.2
	2DPCA	58.9±11.6	42.1±4.5	41.8±10.3

TABLE V
AVERAGED ERROR RATES (MEAN±STD %) OBTAINED BY BPPCA AND PPCA ON THE IRIS DATA SET. THE METHOD THAT IS STATISTICALLY SIGNIFICANTLY BETTER (WITH A p-VALUE OF 0.05 USING THE TWO-SAMPLE ONE-TAILED t-TEST) THAN THE OTHER METHOD IS MARKED ⋆

Method	Number of training samples per class 5	15	25	35
BPPCA	⋆**5.2**±2.2	⋆**3.5**±1.1	⋆**3.2**±1.2	**3.2**±1.8
PPCA	9.4±3.0	7.1±2.1	5.6±1.8	4.3±2.9

E. Performance on Data With Nonseparable Covariance

Recall that BPPCA relies on the assumption of separable covariance (Section III-B). For low-dimensional data, Lu and Zimmerman [23] proposed a likelihood ratio test for separability. However, for high-dimensional data sets such as those used in Section V-D, this test is impractical as the data covariance matrix becomes singular [35].

To study how nonseparability affects BPPCA, we will examine its performance on the classical iris data set, which is known to have a nonseparable covariance structure [23]. The iris data set has four variables: sepal length, sepal width, petal length, and petal width. In [23], they considered the two crossing factors: "plant part" (sepal or petal) and "physical dimension" (length or width). Using the likelihood ratio test in [23], it is shown that these two factors are not separable.

Table V compares the error rates for BPPCA and PPCA, using the one-nearest-neighbor classifier as in Section V-D. As can be seen, even though BPPCA relies on the separable covariance assumption, it is still significantly better than PPCA (which uses a nonrestrictive covariance). The difference is especially prominent on small training sets. This thus supports the observation in Section III-B that separability can trade bias for lower variance, leading to better generalization even on data sets with nonseparable covariance structure.

VI. CONCLUSION

In this paper, we proposed a bilinear probabilistic model called BPPCA for probabilistic dimension reduction on 2-D data. This signals a breakthrough from the classical 1-D latent variable model to the 2-D case. We developed two maximum likelihood estimation algorithms for BPPCA, one is based on CM while the other is based on AECM. The CM algorithm has faster convergence but higher per-iteration complexity, while the AECM algorithm has slower convergence but scales better on high-dimensional data. Similar to PPCA, we showed that the MLE of the BPPCA parameters ($\mathbf{C}$ and $\mathbf{R}$) are principal subspaces of the column and row covariance matrices (up to scaling and rotation). In contrast, PSOPCA fails to provide such an important connection. Moreover, empirical results on synthetic data and real-world data sets demonstrate the usefulness of BPPCA over existing methods.

Nowadays, many real-world data sets are in the form of 3-D or even higher-order tensor [36]. For example, color images and grayscale video sequences can be regarded as 3-D data, while color video sequence can be regarded as 4-D. Recently, GLRAM has been extended to MPCA for the handling of tensor data [37]. In the future, we will also consider extending BPPCA, and the accompanying CM and AECM algorithms, along this direction.

APPENDIX A
MATRIX-VARIATE NORMAL DISTRIBUTION

The matrix-variate normal distribution is a normal distribution with separable covariance matrix (17) [38]. It is a generalization for the multivariate normal distribution in 1-D. Formally, it is defined as follows.

Definition 1: A random matrix $\mathbf{X} \in \mathbb{R}^{d_c \times d_r}$ is said to follow matrix-variate normal, denoted $\mathcal{N}_{d_c,d_r}(\mathbf{W}, \Sigma_c, \Sigma_r)$, with mean matrix $\mathbf{W}$, column covariance matrix $\Sigma_c \in \mathbb{R}^{d_c \times d_c}$ and row covariance matrix $\Sigma_r \in \mathbb{R}^{d_r \times d_r}$, if $\mathrm{vec}(\mathbf{X}) \sim \mathcal{N}_{d_c \times d_r}(\mathrm{vec}(\mathbf{W}), \Sigma_r \otimes \Sigma_c)$. The pdf of $\mathbf{X}$ is given by

$$p(\mathbf{X}) = (2\pi)^{-\frac{1}{2}d_r d_c} |\Sigma_c|^{-\frac{1}{2}d_r} |\Sigma_r|^{-\frac{1}{2}d_c} \mathrm{etr}\left\{-\frac{1}{2}\Sigma_c^{-1}(\mathbf{X}-\mathbf{W})\Sigma_r^{-1}(\mathbf{X}-\mathbf{W})'\right\} \quad (43)$$

where $\mathrm{etr}(\cdot) = \exp(\mathrm{tr}(\cdot))$.

The pdf (43) of the matrix-variate normal is obtained by rewriting the pdf of $\mathrm{vec}(\mathbf{X})$ in vector form into the equivalent matrix form. If $d_r = 1$ or $d_c = 1$, the matrix-variate normal degenerates to multivariate normal.

APPENDIX B
ROTATION AND SCALING INDETERMINACIES OF BPPCA

The BPPCA model is unique up to the following transformations.

1) Orthogonal rotations of the factor loading matrices, latent matrix, column and row noise matrices: For any orthogonal matrices $\mathbf{V}_c \in \mathbb{R}^{q_c \times q_c}$ and $\mathbf{V}_r \in \mathbb{R}^{q_r \times q_r}$, it is

easy to see that

$$\begin{aligned}&\mathbf{CZR}' + \mathbf{W} + \mathbf{C}\boldsymbol{\epsilon}_r + \boldsymbol{\epsilon}_c\mathbf{R}' + \boldsymbol{\epsilon}\\ &= (\mathbf{CV}_c')(\mathbf{V}_c\mathbf{ZV}_r')(\mathbf{V}_r\mathbf{R}') + \mathbf{W} + (\mathbf{CV}_c')(\mathbf{V}_c\boldsymbol{\epsilon}_r)\\ &\quad + (\boldsymbol{\epsilon}_c\mathbf{V}_r')(\mathbf{V}_r\mathbf{R}') + \boldsymbol{\epsilon}.\end{aligned}$$

As a subspace learning method, we are interested in the subspaces spanned by the columns of $\mathbf{C}$ and $\mathbf{R}$ and hence this rotation indeterminacy is not a matter of concern.

2) Scaling of the column and row factor loading matrices. By multiplying $(\mathbf{C}, \sigma_c)$ by a positive constant a and $(\mathbf{R}, \sigma_r)$ by a^{-1} simultaneously, it is easy to see that

$$\begin{aligned}&\mathbf{CZR}' + \mathbf{W} + \mathbf{C}\boldsymbol{\epsilon}_r + \boldsymbol{\epsilon}_c\mathbf{R}' + \boldsymbol{\epsilon}\\ &= a\mathbf{CZR}'a^{-1} + \mathbf{W} + a\mathbf{C}\boldsymbol{\epsilon}_r a^{-1} + a\boldsymbol{\epsilon}_c\mathbf{R}'a^{-1} + a\boldsymbol{\epsilon}a^{-1}.\end{aligned}$$

This is not a problem as we are usually interested in: 1) the Kronecker product of the column and row parameters (instead of either one of them); and 2) the column and row principal subspaces, and the variance ratios contained in these subspaces. For 1), clearly, the effect of scaling can be eliminated. For 2), the scaling effect is also eliminated as follows. It can be seen from (30) that the change $\mathbf{C} \to a\mathbf{C}$ and $\sigma_c^2 \to a\sigma_c^2$ leads to $\mathbf{S}_r \to a^{-1}\mathbf{S}_r$ and hence its eigenvalues $\lambda_{ri} \to a^{-1}\lambda_{ri}, i = 1, \ldots, d_r$. Consequently, $\mathbf{U}_r$ remains unchanged, $\Lambda_r \to a^{-1}\Lambda_r, \mathbf{R} \to a^{-1}\mathbf{R}$ in (31) and $\sigma_r^2 \to a^{-1}\sigma_r^2$ in (32). Thus, the row principal subspace spanned by $\mathbf{U}_r$ is unchanged and the variance ratio $\sum_{i=1}^{q_r} a^{-1}\lambda_{ri} / \sum_{i=1}^{d_r} a^{-1}\lambda_{ri}$ is unchanged as well. A similar conclusion can be drawn for the column principal subspace and its variance ratio. Thus, the scaling effect is eliminated.

APPENDIX C
DERIVATIONS FOR THE PROBABILITY DISTRIBUTIONS IN SECTION III-D

From (18) and (43), the probability density of $\mathbf{Y}^r$ given $\mathbf{Z}$ can be obtained as

$$\begin{aligned}p(\mathbf{Y}^r|\mathbf{Z}) &= (2\pi\sigma_r^2)^{-\frac{1}{2}d_r q_c}\\ &\quad \text{etr}\left\{-\frac{1}{2}(\mathbf{Y}^r - \mathbf{ZR}')\sigma_r^{-2}(\mathbf{Y}^r - \mathbf{ZR}')'\right\} \qquad (44)\end{aligned}$$

and the prior density of the latent matrix $\mathbf{Z}$ is

$$p(\mathbf{Z}) = (2\pi)^{-\frac{1}{2}q_r q_c}\text{etr}\left\{-\frac{1}{2}\mathbf{ZZ}'\right\}. \qquad (45)$$

From (44) and (45), we have the marginal density of $\mathbf{Y}^r$

$$\begin{aligned}p(\mathbf{Y}^r) &= \int p(\mathbf{Y}^r|\mathbf{Z})p(\mathbf{Z})d\mathbf{Z}\\ &= (2\pi)^{-\frac{1}{2}d_r q_c}\text{etr}\left\{-\frac{1}{2}\mathbf{Y}^r\Sigma_r^{-1}\mathbf{Y}^{r\prime}\right\} \qquad (46)\end{aligned}$$

where Σ_r is given by (15). Using the Bayes' rule, the conditional density of $\mathbf{Z}$ given $\mathbf{Y}^r$ is

$$\begin{aligned}&p(\mathbf{Z}|\mathbf{Y}^r) = (2\pi)^{-\frac{1}{2}q_r q_c}\\ &\text{etr}\left\{-\frac{1}{2}(\mathbf{Z} - \mathbf{Y}^r\mathbf{RM}_r^{-1})\sigma_r^{-2}\mathbf{M}_r(\mathbf{Z} - \mathbf{Y}^r\mathbf{RM}_r^{-1})'\right\}\end{aligned}$$

where $\mathbf{M}_r$ is given by (24). Similarly, from (18) and (43), the conditional density of $\mathbf{Y}_\epsilon^r$ given $\boldsymbol{\epsilon}_c$ is

$$\begin{aligned}p(\mathbf{Y}_\epsilon^r|\boldsymbol{\epsilon}_c) &= (2\pi\sigma_c^2\sigma_r^2)^{-\frac{1}{2}d_r d_c}\\ &\quad \text{etr}\left\{-\frac{1}{2}\sigma_c^{-2}(\mathbf{Y}_\epsilon^r - \boldsymbol{\epsilon}_c\mathbf{R}')\sigma_r^{-2}(\mathbf{Y}_\epsilon^r - \boldsymbol{\epsilon}_c\mathbf{R}')'\right\} \qquad (47)\end{aligned}$$

and the prior distribution of the noise matrix $\boldsymbol{\epsilon}_c$ is

$$p(\boldsymbol{\epsilon}_c) = (2\pi\sigma_c^2)^{-\frac{1}{2}q_r d_c}\text{etr}\left\{-\frac{1}{2}\sigma_c^{-2}\boldsymbol{\epsilon}_c\boldsymbol{\epsilon}_c'\right\}. \qquad (48)$$

Using (47) and (48), we obtain

$$\begin{aligned}p(\mathbf{Y}_\epsilon^r) &= \int p(\mathbf{Y}_\epsilon^r|\boldsymbol{\epsilon}_c)p(\boldsymbol{\epsilon}_c)d\boldsymbol{\epsilon}_c\\ &= (2\pi\sigma_c^2)^{-\frac{1}{2}d_r d_c}|\Sigma_r|^{-\frac{1}{2}d_c}\text{etr}\left\{-\frac{1}{2}\sigma_c^{-2}\mathbf{Y}_\epsilon^r\Sigma_r^{-1}\mathbf{Y}_\epsilon^{r\prime}\right\}. \qquad (49)\end{aligned}$$

Substituting $\mathbf{Y}_\epsilon^r = \mathbf{X} - \mathbf{CY}^r - \mathbf{W}$ into (49) and using (46), we have

$$\begin{aligned}p(\mathbf{X}) &= \int p(\mathbf{X}|\mathbf{Y}^r)p(\mathbf{Y}^r)d\mathbf{Y}^r\\ &= (2\pi)^{-\frac{1}{2}d_r d_c}|\Sigma_c|^{-\frac{1}{2}d_r}|\Sigma_r|^{-\frac{1}{2}d_c}\\ &\quad \text{etr}\left\{-\frac{1}{2}\Sigma_c^{-1}(\mathbf{X} - \mathbf{W})\Sigma_r^{-1}(\mathbf{X} - \mathbf{W})'\right\}\end{aligned}$$

where Σ_r is given by (15), and the conditional density of $\mathbf{Y}^r$ given $\mathbf{X}$ is

$$\begin{aligned}&p(\mathbf{Y}^r|\mathbf{X}) = (2\pi)^{-\frac{1}{2}d_r q_c}\\ &\text{etr}\left\{-\frac{1}{2}\sigma_c^{-2}\mathbf{M}_c(\mathbf{Y}^r - \mathbf{M}_c^{-1}\mathbf{C}'\mathbf{X})\Sigma_r^{-1}(\mathbf{Y}^r - \mathbf{M}_c^{-1}\mathbf{C}'\mathbf{X})'\right\}\end{aligned}$$

where $\mathbf{M}_c$ is given by (24).

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 2006.
[2] K. Zhang and J. T. Kwok, "Clustered Nyström method for large scale manifold learning and dimension reduction," *IEEE Trans. Neural Netw.*, vol. 21, no. 10, pp. 1576–1587, Oct. 2010.
[3] S. Yu, "Advanced probabilistic models for clustering and projection," Ph.D. dissertation, Faculty Math. Comput. Sci. Statist., Univ. Munich, Munich, Germany, 2006.
[4] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analysers," *Neural Comput.*, vol. 11, no. 2, pp. 443–482, 1999.
[5] D. J. Bartholomew, *Latent Variable Models and Factor Analysis*, 1st ed. New York: Oxford Univ. Press, 1987.
[6] J. H. Zhao and Q. Jiang, "Probabilistic PCA for t distributions," *Neurocomputing*, vol. 69, nos. 16–18, pp. 2217–2226, 2006.
[7] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York: Springer-Verlag, 2002.
[8] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 696–710, Jul. 1997.
[9] J. Ye, R. Janardan, and Q. Li, "GPCA: An efficient dimension reduction scheme for image compression and retrieval," in *Proc. 10th ACM Int. Conf. Knowl. Discov. Data Min.*, Seattle, WA, Aug. 2004, pp. 354–363.
[10] X. Xie, S. Yan, J. Kwok, and T. Huang, "Matrix-variate factor analysis and its applications," *IEEE Trans. Neural Netw.*, vol. 19, no. 10, pp. 1821–1826, Oct. 2008.
[11] J. Ye, "Generalized low rank approximations of matrices," *Mach. Learn.*, vol. 61, nos. 1–3, pp. 167–191, 2005.
[12] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "2-D PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2004.

[13] S. Yu, J. Bi, and J. Ye, "Probabilistic interpretations and extensions for a family of 2D PCA-style algorithms," in *Proc. KDD Workshop Data Min. Using Matri. Tensors*, Las Vegas, NV, Aug. 2008, pp. 1–7.
[14] S. Yu, J. Bi, and J. Ye, "Matrix-variate and higher-order probabilistic projections," *Data Min. Knowl. Discov.*, vol. 22, no. 3, pp. 372–392, 2011.
[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc. Series B Stat. Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.
[16] C. M. Bishop, "Bayesian PCA," *Advances in Neural Information Proceeding System*. Cambridge, MA: MIT Press, 1999.
[17] V. Šmídl and A. Quinn, "On Bayesian principal component analysis," *Comput. Stat. Data Anal.*, vol. 51, no. 9, pp. 4101–4123, 2007.
[18] J. H. Friedman, "Regularized discriminant analysis," *J. Am. Stat. Assoc.*, vol. 84, no. 405, pp. 165–175, 1989.
[19] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proc. Int. Conf. Comput. Vision Pattern Recognit.*, Minneapolis, MN, Jun. 2007, pp. 1–7.
[20] K. Mardia and C. Goodall, "Spatial-temporal analysis of multivariate environmental monitoring data," in *Proc. Multivariate Environ. Stat.*, 1993, pp. 347–385.
[21] K. Werner and M. Jansson, "Estimating MIMO channel covariances from training data under the Kronecker model," *Signal Process.*, vol. 89, no. 1, pp. 1–13, 2009.
[22] J. C. de Munck, H. M. Huizenga, L. J. Waldorp, and R. M. Heethaar, "Estimating stationary dipoles from MEG/EEG data contaminated with spatially and temporally correlated background noise," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1565–1572, Jul. 2002.
[23] N. Lu and D. L. Zimmerman, "The likelihood ratio test for a separable covariance matrix," *Stat. Probabil. Lett.*, vol. 73, no. 4, pp. 449–457, 2005.
[24] K. Werner, M. Jansson, and P. Stoica, "On estimation of covariance matrices with Kronecker product structure," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 478–491, Feb. 2008.
[25] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York: Springer-Verlag, 2009.
[26] S. Dudoit, J. Fridlyand, and T. P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data," *J. Am. Stat. Assoc.*, vol. 97, no. 457, pp. 77–87, 2002.
[27] I. L. Dryden, L. Bai, C. J. Brignell, and L. Shen, "Factored principal components analysis, with applications to face recognition," *Stat. Comput.*, vol. 19, no. 3, pp. 229–238, 2009.
[28] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1996.
[29] W. J. Zangwill, *Nonlinear Programming: A Unified Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1969.
[30] X. L. Meng and D. B. Rubin, "Maximum likelihood estimation via the ECM algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.
[31] X. L. Meng and D. A. van Dyk, "The EM algorithm: An old folk-song sung to a fast new tune," *J. R. Stat. Soc. Series B Stat. Methodol.*, vol. 59, no. 3, pp. 511–567, 1997.
[32] J. H. Zhao and P. L. H. Yu, "Fast ML estimation for the mixture of factor analyzers via an ECM algorithm," *IEEE Trans. Neural Netw.*, vol. 19, no. 11, pp. 1956–1961, Nov. 2008.
[33] A. Edelman, T. Arias, and S. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. App.*, vol. 20, no. 2, pp. 303–353, 1998.
[34] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.
[35] S. Chen, L. Zhang, and P. Zhong, "Tests for high-dimensional covariance matrices," *J. Am. Stat. Assoc.*, vol. 105, no. 490, pp. 810–819, 2010.
[36] P. Comon and C. Jutten, *Handbook of Blind Source Separation, Independent Component Analysis and Applications*. New York: Academic, 2010.
[37] H. P. Lu, N. P. Konstantinos, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.
[38] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*. London, U.K.: Chapman & Hall, 1999.

Jianhua Zhao received the Ph.D. degree in statistics from the University of Hong Kong, Hong Kong, in 2009.

He is currently an Associate Professor with the School of Statistics and Mathematics, Yunnan University of Finance and Economics, Kunming, China. His current research interests include statistical machine learning and pattern recognition.

Philip L. H. Yu is an Associate Professor with the Department of Statistics and Actuarial Science, University of Hong Kong, Hong Kong. His current research interests include computational statistics including data mining, preference learning, and financial risk management.

He currently serves as an Associate Editor of a number of journals, including *Computational Statistics and Data Analysis* and *Computational Statistics*.

James T. Kwok received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 1996.

He was with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, as an Assistant Professor. He is currently a Professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His current research interests include kernel methods, machine learning, pattern recognition, and artificial neural networks.

Dr. Kwok received the IEEE Outstanding Paper Award in 2006 and the Second-Class Prize of the National Natural Science Award 2008, China, in 2009. He is currently an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS and the *Neurocomputing Journal*.