algorithms

MDPI

Article

Robust Bilinear Probabilistic Principal Component Analysis

Yaohang Lu and Zhongming Teng *

College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China; yaohlu97@163.com
* Correspondence: zhmteng@fafu.edu.cn

Abstract: Principal component analysis (PCA) is one of the most popular tools in multivariate exploratory data analysis. Its probabilistic version (PPCA) based on the maximum likelihood procedure provides a probabilistic manner to implement dimension reduction. Recently, the bilinear PPCA (BPPCA) model, which assumes that the noise terms follow matrix variate Gaussian distributions, has been introduced to directly deal with two-dimensional (2-D) data for preserving the matrix structure of 2-D data, such as images, and avoiding the curse of dimensionality. However, Gaussian distributions are not always available in real-life applications which may contain outliers within data sets. In order to make BPPCA robust for outliers, in this paper, we propose a robust BPPCA model under the assumption of matrix variate *t* distributions for the noise terms. The alternating expectation conditional maximization (AECM) algorithm is used to estimate the model parameters. Numerical examples on several synthetic and publicly available data sets are presented to demonstrate the superiority of our proposed model in feature extraction, classification and outlier detection.

Keywords: 2-D data; probabilistic principal component analysis; AECM algorithm; matrix variate Gaussian distributions; matrix variate *t* distributions; outliers

check for updates

Citation: Lu, Y.; Teng, Z. Robust Bilinear Probabilistic Principal Component Analysis. *Algorithms* **2021**, *14*, 322. https://doi.org/10.3390/a14110322

Academic Editors: Frank Werner and Bogdan Dumitrescu

Received: 27 September 2021
Accepted: 30 October 2021
Published: 1 November 2021

1. Introduction

High-dimensional data are increasingly collected for a variety of applications in the real world. However, high-dimensional data are not often distributed uniformly in their ambient space, instead of that the interesting structure inside the data often lies in a low-dimensional space [1]. One of the fundamental challenges is how to find the low-dimensional data representation for high-dimensional observed data in pattern recognition, machine learning and statistics [2,3]. Principal component analysis (PCA) [4] is arguably the most well-known dimension reduction method for high-dimensional data analysis, and it aims to find the first few principal eigenvectors corresponding to the first few largest eigenvalues of the covariance matrix, and then projects the high-dimensional data onto the low-dimensional subspace spanned by these principal eigenvectors to achieve the purpose of dimensionality reduction.

The traditional PCA is concerned with vectorial data, i.e., 1-D data. For 2-D image trained sample matrices, it is usual to first convert 2-D image matrices into 1-D image vectors. This transformation leads to higher dimensional image sample vectors and a larger covariance matrix, and thus suffers from the difficulty of accurately evaluating the principal eigenvectors of the large scale covariance matrix. Furthermore, such vectorizing of 2-D data destroys the natural matrix structure, and ignores potentially valuable information about the spatial relationships among 2-D data. Therefore, two-dimensional PCA (2DPCA) type algorithms [5–7] are proposed to compute principal component weight matrices directly based on 2-D image training with sample matrices instead of using vectorization.

These conventional PCA and 2DPCA algorithms are both derived and interpreted in the standard algebraic framework, thus they lack capability in handling issues of statistical inference or missing data. To remedy these drawbacks, a probabilistic PCA model (PPCA) has been proposed by Tipping and Bishop in [8], which is processed by assuming some

Gaussian distributions on observations with introduced extra latent variables, and it has been successfully applied in many machine learning tasks [9]. Following PPCA, a probabilistic second-order PCA, called PSOPCA, is developed in [10] to directly model 2-D image matrices based on the so-called matrix variate Gaussian distributions.

Throughout this paper, $\mathbb{R}^{n\times m}$ is the set of all $n \times m$ real matrices, I_n and $0_{m\times n}$ are the $n \times n$ identity matrix and $m \times n$ zero matrix, respectively. The superscript "$\cdot^{\mathrm{T}}$" means transpose only, $\|\cdot\|_2$ and $\|\cdot\|_{\mathrm{F}}$ denote the ℓ_2-norm and Frobenius norm of a matrix, respectively. Denoted by $\mathcal{N}_{d_c,d_r}(M,\Omega_c,\Omega_r)$ is the matrix variate Gaussian distribution [11] with the mean matrix $M \in \mathbb{R}^{d_c\times d_r}$, column covariance $\Omega_c \in \mathbb{R}^{d_c\times d_c}$ and row covariance $\Omega_r \in \mathbb{R}^{d_r\times d_r}$. A random matrix $X \in \mathbb{R}^{d_c\times d_r}$ is said to follow the matrix variate Gaussian distribution $\mathcal{N}_{d_c,d_r}(M,\Omega_c,\Omega_r)$, i.e.,

$$X \sim \mathcal{N}_{d_c,d_r}(M,\Omega_c,\Omega_r), \text{ if } \operatorname{vec}(X) \sim \mathcal{N}(\operatorname{vec}(M), \Omega_r \otimes \Omega_c), \tag{1}$$

where $\operatorname{vec}(\cdot)$ is the vectorization of a matrix obtained by stacking the columns of the matrix on top of one another. That means that the probability density function (pdf) of X is

$$\begin{aligned} p(X) &= (2\pi)^{-\frac{1}{2}d_cd_r}|\Omega_r\otimes\Omega_c|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}[\operatorname{vec}(X-M)]^{\mathrm{T}}(\Omega_r^{-1}\otimes\Omega_c^{-1})\operatorname{vec}(X-M)\right) \\ &= (2\pi)^{-\frac{1}{2}d_cd_r}|\Omega_r|^{-\frac{1}{2}d_c}|\Omega_c|^{-\frac{1}{2}d_r}\operatorname{etr}\left(-\frac{1}{2}\Omega_c^{-1}(X-M)\Omega_r^{-1}(X-M)^{\mathrm{T}}\right), \end{aligned} \tag{2}$$

where "$\otimes$" is the Kronecker product of two matrices, $\operatorname{etr}(\cdot) = \exp(\operatorname{tr}(\cdot))$ and $\operatorname{tr}(\cdot)$ denotes the trace of a matrix. The last equality of (2) holds because of $|\Omega_r\otimes\Omega_c|^{-\frac{1}{2}} = |\Omega_r|^{-\frac{1}{2}d_c}|\Omega_c|^{-\frac{1}{2}d_r}$ and

$$[\operatorname{vec}(X-M)]^{\mathrm{T}}(\Omega_r^{-1}\otimes\Omega_c^{-1})\operatorname{vec}(X-M) = \operatorname{tr}\left(\Omega_c^{-1}(X-M)\Omega_r^{-1}(X-M)^{\mathrm{T}}\right).$$

See ([11], Theorem 1.2.21) and ([11], Theorem 1.2.22) for more details.

PSOPCA in [10] considers the following two-sided latent matrix variable model

$$\begin{cases} X = CZR^{\mathrm{T}} + W + \Upsilon, \\ Z \sim \mathcal{N}_{q_c,q_r}(0_{q_c\times q_r}, I_{q_c}, I_{q_r}),\ \Upsilon \sim \mathcal{N}_{d_c,d_r}(0_{d_c\times d_r}, \sigma^2 I_{d_c}, \sigma^2 I_{d_r}), \end{cases} \tag{3}$$

where $C \in \mathbb{R}^{d_c\times q_c}$ and $R \in \mathbb{R}^{d_r\times q_r}$ are the column and row factor loading matrices, respectively, $W \in \mathbb{R}^{d_c\times d_r}$ and $\Upsilon \in \mathbb{R}^{d_c\times d_r}$ are the mean and error matrices, respectively, and $Z \in \mathbb{R}^{q_c\times q_r}$ is the latent core variable of X. The PSOPCA model is further extended to the bilinear probabilistic principal component analysis (BPPCA) model in [12] for better establishing the relationship with the 2DPCA algorithm [6], which is defined as

$$\begin{cases} X = CZR^{\mathrm{T}} + W + C\mathcal{E}_r + \mathcal{E}_cR^{\mathrm{T}} + \mathcal{E}, \\ \mathcal{E}_c \sim \mathcal{N}_{d_c,q_r}(0_{d_c\times q_r}, \sigma_c^2 I_{d_c}, I_{q_r}), \quad \mathcal{E}_r \sim \mathcal{N}_{q_c,d_r}(0_{q_c\times d_r}, I_{q_c}, \sigma_r^2 I_{d_r}), \\ \mathcal{E} \sim \mathcal{N}_{d_c,d_r}(0_{d_c\times d_r}, \sigma_c^2 I_{d_c}, \sigma_r^2 I_{d_r}),\ \ Z \sim \mathcal{N}_{q_c,q_r}(0_{q_c\times q_r}, I_{q_c}, I_{q_r}). \end{cases} \tag{4}$$

In contrast to the PSOPCA model (3), the column and row noise matrices $\mathcal{E}_c \in \mathbb{R}^{d_c\times q_r}$ and $\mathcal{E}_r \in \mathbb{R}^{q_c\times d_r}$ with different noise variances σ_c^2 and σ_r^2, respectively, are included in the BPPCA model, and $\mathcal{E} \in \mathbb{R}^{d_c\times d_r}$ is represented as the common noise matrix. The model (4) improves the flexibility in capturing data uncertainty, and makes the marginal distribution $p(X)$ to be the matrix variable Gaussian. In particular, we can see that if $\mathcal{E}_r$ and $\mathcal{E}_c$ are removed and $\sigma_c = \sigma_r$, then (4) reduces to the PSOPCA model.

All of the above mentioned probabilistic models assume that the noise terms follow Gaussian distributions. It is a well-known issue that Gaussian noises will lead to a serious drawback while dealing with anomalous observations. Thus, the probabilistic PCA models based on Gaussian distributions are not robust to outliers. To make probabilistic models which are insensitive to outliers, one prefers heavy-tailed distributions, such as the Student t distribution or centered Laplacian distribution with ℓ_1-norm. Using the t distribution or centered Laplacian distribution instead of the Gaussian distribution in the PPCA model [8] results in tPPCA [13,14] and probabilistic L1-PPCA [15] algorithms, respectively. Similarly, a robust version of PSOPCA, called L1-2DPPCA, is introduced in [16] based on the Laplacian distribution combined with variational EM-type algorithms to learn parameters. However, it is difficult to generalize a robust version of the BPPCA algorithm based on the Laplacian distribution. The reason is that if the error term Υ in the PSOPCA model is a Laplacian distribution, then the condition distribution $X|Z$ is also a Laplacian distribution, but it does not hold in the BPPCA model. Fortunately, the same goal can be achieved by using the t distribution. In fact, the Gaussian distribution is a special t distribution. Compared to the Gaussian distribution, the t distribution has significantly heavier tails and contains one more free parameter. Recently, some robust probabilistic models under the assumption of the t distribution have already been done successfully by a number of researchers in [17–22]. Motivated by these facts, we will continue the effort to develop a robust BPPCA model from matrix variate t distributions to handle 2-D data sets in the presence of outliers.

The remainder of the paper is organized as follows. Section 2 introduces some notations and a matrix variate t distribution which are essential to our later development. The robust BPPCA model and its associated parameters estimation based on the AECM algorithm are given and analyzed in detail in Section 3. Section 4 is dedicated to present some numerical examples for showing the behaviors of our proposed model and to support our analysis. Finally, conclusions are made in Section 5.

2. Preliminaries

Let $X \in \mathbb{R}^{d_c \times d_r}$, $M \in \mathbb{R}^{d_c \times d_r}$, $\Omega_c \in \mathbb{R}^{d_c \times d_c}$ and $\Omega_r \in \mathbb{R}^{d_r \times d_r}$, and the probability density function of the random variable μ having a Gamma distribution with parameters α and β, i.e., $\mu \sim Ga(\alpha, \beta)$, be

$$p(\mu) = \beta^{\alpha} \mu^{\alpha-1} \exp(-\beta\mu)/\Gamma(\alpha), \tag{5}$$

where $\Gamma(\cdot)$ is the Gamma function, i.e.,

$$\Gamma(\alpha) = \int_0^{+\infty} \tau^{\alpha-1} \exp(-\tau)\, d\tau. \tag{6}$$

Analogously to the process of tPPCA in [14], we derive the matrix variate t distribution in this paper by considering

$$\mathrm{vec}(X) \sim \int_0^{+\infty} \mathcal{N}\left(\mathrm{vec}(M), \frac{\Omega_r \otimes \Omega_c}{\mu}\right) p(\mu)\, d\mu,$$

where $\mu \sim Ga(\nu/2, \nu/2)$. Let $\delta = \mathrm{tr}\left(\Omega_c^{-1}(X-M)\Omega_r^{-1}(X-M)^{\mathrm{T}}\right)$. We have

$$\begin{aligned}
&\int_0^{+\infty} \mathcal{N}\left(\mathrm{vec}(M), \frac{\Omega_r \otimes \Omega_c}{\mu}\right) p(\mu) d\mu \\
=&\int_0^{+\infty} (2\pi)^{-\frac{d_c d_r}{2}} \left|\frac{\Omega_r \otimes \Omega_c}{\mu}\right|^{-\frac{1}{2}} \exp\left(-\frac{\delta\mu}{2}\right) \left(\frac{\nu}{2}\right)^{\frac{\nu}{2}} \mu^{\frac{\nu}{2}-1} \exp\left(-\frac{\nu}{2}\mu\right) \frac{1}{\Gamma\left(\frac{\nu}{2}\right)}\, d\mu \\
=&\frac{(2\pi)^{-\frac{d_c d_r}{2}} |\Omega_c|^{-\frac{d_r}{2}} |\Omega_r|^{-\frac{d_c}{2}} \left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \int_0^{+\infty} \mu^{\frac{\nu+d_c d_r}{2}-1} \exp\left(-\frac{(\delta+\nu)\mu}{2}\right) d\mu.
\end{aligned} \tag{7}$$

Let $\tau = \frac{(\delta+\nu)\mu}{2}$. Then, $\mu = \frac{2\tau}{\nu+\delta}$ and $d\mu = \frac{2}{\nu+\delta}\,d\tau$. Therefore, (7) can be rewritten as

$$\begin{aligned}
&\int_0^{+\infty} \mathcal{N}\left(\mathrm{vec}(M), \frac{\Omega_r \otimes \Omega_c}{\mu}\right) p(\mu) d\mu \\
=&\frac{(2\pi)^{-\frac{d_c d_r}{2}} |\Omega_c|^{-\frac{d_r}{2}} |\Omega_r|^{-\frac{d_c}{2}} \left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \left(\frac{2}{\nu+\delta}\right)^{\frac{\nu+d_c d_r}{2}} \int_0^{+\infty} \tau^{\frac{\nu+d_c d_r}{2}-1} \exp(-\tau)\, d\tau \\
=&\frac{(2\pi)^{-\frac{d_c d_r}{2}} |\Omega_c|^{-\frac{d_r}{2}} |\Omega_r|^{-\frac{d_c}{2}} \left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \left(\frac{2}{\nu+\delta}\right)^{\frac{\nu+d_c d_r}{2}} \Gamma\left(\frac{\nu+d_c d_r}{2}\right) \\
=&\frac{\pi^{-\frac{d_c d_r}{2}} |\Omega_c|^{-\frac{d_r}{2}} |\Omega_r|^{-\frac{d_c}{2}} \nu^{\frac{\nu}{2}} \Gamma\left(\frac{\nu+d_c d_r}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \nu^{-\frac{\nu+d_c d_r}{2}} \left(1+\frac{\delta}{\nu}\right)^{-\frac{\nu+d_c d_r}{2}} \\
=&\frac{|\Omega_c|^{-\frac{d_r}{2}} |\Omega_r|^{-\frac{d_c}{2}} \Gamma\left(\frac{\nu+d_c d_r}{2}\right)}{(\nu\pi)^{\frac{d_c d_r}{2}} \Gamma\left(\frac{\nu}{2}\right)} \left(1+\frac{\delta}{\nu}\right)^{-\frac{\nu+d_c d_r}{2}}.
\end{aligned} \tag{8}$$

In this paper, if the pdf of the random matrix X is

$$p(X) = \frac{|\Omega_c|^{-\frac{d_r}{2}} |\Omega_r|^{-\frac{d_c}{2}} \Gamma\left(\frac{\nu+d_c d_r}{2}\right)}{(\nu\pi)^{\frac{d_c d_r}{2}} \Gamma\left(\frac{\nu}{2}\right)} \left(1+\frac{\delta}{\nu}\right)^{-\frac{\nu+d_c d_r}{2}}, \tag{9}$$

then the random matrix X is said to follow the matrix variate t distribution with degrees of freedom ν, and is denoted by

$$X \sim \mathcal{T}_{d_c,d_r}(\nu, M, \Omega_c, \Omega_r).$$

In particular, if $d_c = 1$ or $d_r = 1$, then the matrix variate t distribution degenerates to the multivariate t distribution. As the classical multivariate t distribution, another favorite perspective on the matrix variate t distribution which is critical to our later developments, is to treat μ as a latent variable, then the conditional distribution of $X|\mu$ is a matrix variate Gaussian distribution by (8), i.e.,

$$X|\mu \sim \mathcal{N}_{d_c,d_r}(M, \mu_c\Omega_c, \mu_r\Omega_r), \tag{10}$$

where $\mu_c\mu_r = \mu^{-1}$. Notice that, despite the non-uniqueness in the factorization $\mu^{-1} = \mu_c\mu_r$, $\mathcal{N}_{d_c,d_r}(M, \mu_c\Omega_c, \mu_r\Omega_r)$ in (10) always returns the same pdf of X.

3. Robust BPPCA

3.1. The Model

In this section, we develop a robust model by replacing the matrix variate Gaussian distribution in BPPCA with the matrix variate t distribution with degrees of freedom ν defined in (9) to deal with 2-D data sets. Specifically, the proposed robust bilinear probabilistic principal analysis model (RBPPCA for short) is defined as

$$\begin{cases}
X = CZR^{\mathrm{T}} + W + C\mathcal{E}_r + \mathcal{E}_c R^{\mathrm{T}} + \mathcal{E}, \\
\mu \sim Ga(\nu/2, \nu/2), \quad \mu_c\mu_r = \mu^{-1}, \\
Z|\mu \sim \mathcal{N}_{q_c,q_r}(0_{q_c\times q_r}, \mu_c I_{q_c}, \mu_r I_{q_r}),\ \mathcal{E}_c|\mu \sim \mathcal{N}_{d_c,q_r}(0_{d_c\times q_r}, \mu_c\sigma_c^2 I_{d_c}, \mu_r I_{q_r}), \\
\mathcal{E}_r|\mu \sim \mathcal{N}_{q_c,d_r}(0_{q_c\times d_r}, \mu_c I_{q_c}, \mu_r\sigma_r^2 I_{d_r}),\ \mathcal{E}|\mu \sim \mathcal{N}_{d_c,d_r}(0_{d_c\times d_r}, \mu_c\sigma_c^2 I_{d_c}, \mu_r\sigma_r^2 I_{d_r}).
\end{cases} \tag{11}$$

As BPPCA [12], in the RBPPCA model (11), $\mathcal{E}_c \in \mathbb{R}^{d_c\times q_r}$, $\mathcal{E}_r \in \mathbb{R}^{q_c\times d_r}$ and $\mathcal{E} \in \mathbb{R}^{d_c\times d_r}$ are the column, row and common noise matrices, respectively, $Z \in \mathbb{R}^{q_c\times q_r}$ is the latent matrix, and these are assumed to be independent of each other, and the mean matrix, and the column and row factor loading matrices are $W \in \mathbb{R}^{d_c\times d_r}$, $C \in \mathbb{R}^{d_c\times q_c}$ and $R \in \mathbb{R}^{d_r\times q_r}$, respectively. Similarly to BPPCA [12], the parameters C, R, σ_c and σ_r can not be uniquely identified, but the interested subspaces spanned by the columns of C and R are unique. The reader is referred to ([12], Appendix B) for details. The difference from BPPCA is that the noise matrices $\mathcal{E}_c$, $\mathcal{E}_r$ and $\mathcal{E}$ and latent matrix variate Z in the RBPPCA model (11) are supposed matrix variate t distributions by (10), i.e.,

$$\mathcal{E}_c \sim \mathcal{T}_{d_c,q_r}(\nu, 0_{d_c\times q_r}, \sigma_c^2 I_{d_c}, I_{q_r}), \quad \mathcal{E}_r \sim \mathcal{T}_{q_c,d_r}(\nu, 0_{q_c\times d_r}, I_{q_c}, \sigma_r^2 I_{d_r}),$$
$$\mathcal{E} \sim \mathcal{T}_{d_c,d_r}(\nu, 0_{d_c\times d_r}, \sigma_c^2 I_{d_c}, \sigma_r^2 I_{d_r}), \quad Z \sim \mathcal{T}_{q_c,q_r}(\nu, 0_{q_c\times q_r}, I_{q_c}, I_{q_r}).$$

It follows by (11) that

$$\begin{aligned} CZR^{\mathrm{T}}|\mu &\sim \mathcal{N}_{d_c,d_r}(0_{d_c\times d_r}, \mu_c CC^{\mathrm{T}}, \mu_r RR^{\mathrm{T}}), \\ C\mathcal{E}_r|\mu &\sim \mathcal{N}_{d_c,d_r}(0_{d_c\times d_r}, \mu_c CC^{\mathrm{T}}, \mu_r\sigma_r^2 I_{d_r}), \\ \mathcal{E}_c R^{\mathrm{T}}|\mu &\sim \mathcal{N}_{d_c,d_r}(0_{d_c\times d_r}, \mu_c\sigma_c^2 I_{d_c}, \mu_r RR^{\mathrm{T}}). \end{aligned}$$

Consequently, $X|\mu \sim \mathcal{N}_{d_c,d_r}(W, \mu_c\Sigma_c, \mu_r\Sigma_r)$ where

$$\Sigma_c = CC^{\mathrm{T}} + \sigma_c^2 I_{d_c} \quad \text{and} \quad \Sigma_r = RR^{\mathrm{T}} + \sigma_r^2 I_{d_r}. \tag{12}$$

That means the random matrix X follows the matrix variate t distribution, i.e., $X \sim \mathcal{T}_{d_c,d_r}(\nu, W, \Sigma_c, \Sigma_r)$. In addition, as shown in [23], the conditional distribution $\mu|X$ which is also required in our later estimation of model parameters is a Gamma distribution, i.e.,

$$\mu|X \sim Ga\left(\frac{\nu + d_c d_r}{2}, \frac{\nu+\rho}{2}\right), \tag{13}$$

where $\rho = \operatorname{tr}\left(\Sigma_c^{-1}(X-W)\Sigma_r^{-1}(X-W)^{\mathrm{T}}\right)$.

By introducing two latent matrix variates $Y^r \in \mathbb{R}^{q_c\times d_r}$ and $Y_\epsilon^r \in \mathbb{R}^{d_c\times d_r}$, the RBPPCA model (11) can be rewritten as

$$\begin{cases} X = CY^r + W + Y_\epsilon^r, \\ Y^r = ZR^{\mathrm{T}} + \mathcal{E}_r, \\ Y_\epsilon^r = \mathcal{E}_c R^{\mathrm{T}} + \mathcal{E}, \end{cases} \tag{14}$$

where Y^r and Y_ϵ^r are the row projected intermediate and residual matrices, respectively. By (11), we have the conditional distributions

$$Y^r|\mu \sim \mathcal{N}_{q_c,d_r}(0_{q_c\times d_r}, \mu_c I_{q_c}, \mu_r\Sigma_r), \tag{15a}$$
$$Y_\epsilon^r|\mu \sim \mathcal{N}_{d_c,d_r}(0_{d_c\times d_r}, \mu_c\sigma_c^2 I_{d_c}, \mu_r\Sigma_r), \tag{15b}$$
$$Y^r|Z,\mu \sim \mathcal{N}_{q_c,d_r}(ZR^{\mathrm{T}}, \mu_c I_{q_c}, \mu_r\sigma_r^2 I_{d_r}), \tag{15c}$$
$$X|Y^r,\mu \sim \mathcal{N}_{d_c,d_r}(CY^r + W, \mu_c\sigma_c^2 I_{d_c}, \mu_r\Sigma_r), \tag{15d}$$

where Σ_r is given by (12). In addition, by using (15c) and the Bayes' rule, the conditional distributions $Z|Y^r,\mu$ and $Y^r|X,\mu$ can be calculated as

$$Z|Y^r,\mu \sim \mathcal{N}_{q_c,q_r}\left(Y^r R\Phi_r^{-1}, \mu_c I_{q_c}, \mu_r\sigma_r^2\Phi_r^{-1}\right), \tag{16a}$$
$$Y^r|X,\mu \sim \mathcal{N}_{q_c,d_r}\left(\Phi_c^{-1}C^{\mathrm{T}}(X-W), \mu_c\sigma_c^2\Phi_c^{-1}, \mu_r\Sigma_r\right), \tag{16b}$$

where

$$\Phi_c = C^{\mathrm{T}}C + \sigma_c^2 I_{q_c} \quad \text{and} \quad \Phi_r = R^{\mathrm{T}}R + \sigma_r^2 I_{q_r}. \tag{17}$$

In (14), the bilinear projection in the RBPPCA model is split into two stages by first projecting the latent matrix Z in the row direction to obtain Y^r, then Y^r being projected in the column direction to finally generate X. Similarly, we can also consider the decomposition of the bilinear projection by first projecting column and then row directions to rewrite (11) as

$$\begin{cases} X = Y^c R^{\mathrm{T}} + W + Y^c_\epsilon, \\ Y^c = CZ + \mathcal{E}_c, \\ Y^c_\epsilon = C\mathcal{E}_r + \mathcal{E}, \end{cases} \tag{18}$$

where $Y^c \in \mathbb{R}^{d_c \times q_r}$ and $Y^c_\epsilon \in \mathbb{R}^{d_c \times d_r}$ with $Y^c|\mu \sim \mathcal{N}_{d_c,q_r}(0_{d_c \times q_r}, \mu_c\Sigma_c, \mu_r I_{q_r})$ and $Y^c_\epsilon|\mu \sim \mathcal{N}_{d_c,d_r}(0_{d_c \times d_r}, \mu_c\Sigma_c, \mu_r\sigma_r^2 I_{d_r})$. Furthermore,

$$Y^c|Z,\mu \sim \mathcal{N}_{d_c,q_r}(CZ, \mu_c\sigma_c^2 I_{d_c}, \mu_r I_{q_r}), \tag{19a}$$

$$X|Y^c,\mu \sim \mathcal{N}_{d_c,d_r}(Y^c R^{\mathrm{T}} + W, \mu_c\Sigma_c, \mu_r\sigma_r^2 I_{d_r}), \tag{19b}$$

$$Z|Y^c,\mu \sim \mathcal{N}_{q_c,q_r}\left(\Phi_c^{-1}C^{\mathrm{T}}Y^c, \mu_c\sigma_c^2\Phi_c^{-1}, \mu_r I_{q_r}\right), \tag{19c}$$

$$Y^c|X,\mu \sim \mathcal{N}_{d_c,q_r}\left((X-W)R\Phi_r^{-1}, \mu_c\Sigma_c, \mu_r\sigma_r^2\Phi_r^{-1}\right). \tag{19d}$$

3.2. Estimation of the Parameters

In the model (11), the parameter set to be estimated is $\theta = \{\nu, \sigma_c, \sigma_r, W, C, R\}$. We will introduce how to calculate the parameters by using the alternating expectation conditional maximization (AECM) algorithm in this subsection. The AECM algorithm [12,24,25] is a two stage iterative optimization technique for finding maximum likelihood solutions. To apply it to the AECM algorithm, we divide the parameter set θ into two subsets $\theta_c = \{\nu, \sigma_c, W, C\}$ and $\theta_r = \{\nu, \sigma_r, W, R\}$.

In the first stage, we consider the AECM algorithm for the model (14) to compute θ_c. Let $\{X_n\}_{n=1}^N$ with $X_n \in \mathbb{R}^{d_c \times d_r}$ be a set of 2-D sample observations. The latent variables' data $\{Y^r_n, \mu_n\}_{n=1}^N$ are treated as "missing data", and the "complete" data log-likelihood is

$$\begin{aligned} L_{com,c}(\theta_c) &= \sum_{n=1}^N \ln\left(p(X_n, Y^r_n, \mu_n)\right) \\ &= \sum_{n=1}^N \ln\left(p(X_n|Y^r_n,\mu_n)p(Y^r_n|\mu_n)p(\mu_n)\right). \end{aligned}$$

It follows by (5), (15a) and (15d) that

$$\begin{aligned} L_{com,c}(\theta_c) = \sum_{n=1}^N \ln\Bigg(&(2\pi)^{-\frac{d_c d_r}{2}} |\mu_{c,n}\sigma_c^2 I_{d_c}|^{-\frac{d_r}{2}} |\mu_{r,n}\Sigma_r|^{-\frac{d_c}{2}} \operatorname{etr}\Big(-\frac{1}{2}(\mu_{r,n}\Sigma_r)^{-1}[X_n - (CY^r_n \\ &+ W)]^{\mathrm{T}}(\mu_{c,n}\sigma_c^2 I)^{-1}[X_n - (CY^r_n + W)]\Big)(2\pi)^{-\frac{q_c d_r}{2}} |\mu_{c,n} I_{q_c}|^{-\frac{d_r}{2}} |\mu_{r,n}\Sigma_r|^{-\frac{q_c}{2}} \\ &\operatorname{etr}\Big(-\frac{1}{2}(\mu_{r,n}\Sigma_r)^{-1}(Y^r_n)^{\mathrm{T}}\mu_{c,n}^{-1}Y^r_n\Big)\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}} \mu_n^{\frac{\nu}{2}-1} \exp\left(-\frac{\nu}{2}\mu_n\right)\frac{1}{\Gamma(\frac{\nu}{2})}\Bigg) \\ = -\sum_{n=1}^N \Bigg\{ &\frac{(d_c+q_c)d_r}{2}\ln(2\pi) + \ln\Gamma\left(\frac{\nu}{2}\right) + \frac{d_c+q_c}{2}\ln|\Sigma_r| - \frac{\nu}{2}\ln\left(\frac{\nu}{2}\right) + \frac{\nu\mu_n}{2} \\ &- \frac{(d_c+q_c)d_r + \nu - 2}{2}\ln\mu_n + \frac{d_c d_r}{2}\ln\sigma_c^2 + \operatorname{tr}\Big(\frac{\mu_n}{2\sigma_c^2}\Sigma_r^{-1}[X_n - (CY^r_n + W)]^{\mathrm{T}} \\ &[X_n - (CY^r_n + W)] + \frac{\mu_n}{2}\Sigma_r^{-1}(Y^r_n)^{\mathrm{T}}Y^r_n\Big)\Bigg\}. \end{aligned}$$

In E-step, given the parameter set $\theta^{(i)} = \{\nu^{(i)}, \sigma_c^{(i)}, \sigma_r^{(i)}, W^{(i)}, C^{(i)}, R^{(i)}\}$ which is obtained from the i-th iteration, we compute the expectation of $L_{com,c}(\theta_c)$ with respect to the condition distribution $Y^r_n, \mu_n|X_n$, i.e.,

$$\begin{aligned}
Q_c(\theta_c) &= \int_{Y_n^r}\int_{\mu_n} L_{com,c}(\theta_c)p(Y_n^r,\mu_n|X_n)d\mu_n dY_n^r \\
&= \int_{Y_n^r}\int_{\mu_n} L_{com,c}(\theta_c)p(Y_n^r|\mu_n,X_n)p(\mu_n|X_n)d\mu_n dY_n^r \\
&= -\sum_{n=1}^{N}\Big\{\ln\Gamma\Big(\frac{\nu}{2}\Big)-\frac{\nu}{2}\ln\Big(\frac{\nu}{2}\Big)+\frac{\nu}{2}\mathbb{E}(\mu_n|X_n)-\frac{(d_c+q_c)d_r+\nu-2}{2}\mathbb{E}(\ln\mu_n|X_n) \\
&\quad +\frac{d_cd_r}{2}\ln\sigma_c^2+\frac{1}{2\sigma_c^2}\mathbb{E}\Big(\operatorname{tr}\big(\mu_n\Sigma_r^{-1}[X_n-(CY_n^r+W)]^{\mathrm{T}}[X_n-(CY_n^r+W)]\big)\big|X_n\Big)\Big\} \\
&\quad + \text{constant},
\end{aligned} \tag{20}$$

where the constant contains those terms without referring the parameters in the set θ_c. We denote $E_{\mu_n}^{(i)}=\mathbb{E}(\mu_n|X_n)$ and $E_{Y_n^r}^{(i)}=\mathbb{E}(Y_n^r|X_n,\mu_n)$ for convenience. It is noted by (13) and (16b) that given the parameter set $\theta^{(i)}=\{\nu^{(i)},\sigma_c^{(i)},\sigma_r^{(i)},W^{(i)},C^{(i)},R^{(i)}\}$, the conditional distributions of $\mu_n|X_n$ and $Y_n^r|X_n,\mu_n$ are known. That is, for $1\le n\le N$,

$$\begin{aligned}
\mu_n|X_n &\sim Ga\Big(\frac{\nu^{(i)}+d_cd_r}{2},\frac{\nu^{(i)}+\rho_n^{(i)}}{2}\Big), \\
Y_n^r|X_n,\mu_n &\sim \mathcal{N}_{q_c,d_r}\Big((\Phi_c^{(i)})^{-1}(C^{(i)})^{\mathrm{T}}(X_n-W^{(i)}),\mu_{c,n}(\sigma_c^{(i)})^2(\Phi_c^{(i)})^{-1},\mu_{r,n}\Sigma_r^{(i)}\Big),
\end{aligned}$$

where $\mu_{r,n}\mu_{c,n}=\mu_n$, $\Phi_c^{(i)}=(C^{(i)})^{\mathrm{T}}C^{(i)}+(\sigma_c^{(i)})^2I_{q_c}$, $\Sigma_r^{(i)}=R^{(i)}(R^{(i)})^{\mathrm{T}}+(\sigma_r^{(i)})^2I_{d_r}$, and $\rho_n^{(i)}=\operatorname{tr}\big((\Sigma_c^{(i)})^{-1}(X_n-W^{(i)})(\Sigma_r^{(i)})^{-1}(X_n-W^{(i)})^{\mathrm{T}}\big)$ with $\Sigma_c^{(i)}=C^{(i)}(C^{(i)})^{\mathrm{T}}+(\sigma_c^{(i)})^2I_{d_c}$. Then, it is easy to obtain that

$$E_{\mu_n}^{(i)}=\frac{\nu^{(i)}+d_cd_r}{\nu^{(i)}+\rho_n^{(i)}} \quad\text{and}\quad E_{Y_n^r}^{(i)}=(\Phi_c^{(i)})^{-1}(C^{(i)})^{\mathrm{T}}(X_n-W^{(i)}). \tag{21}$$

In addition, based on the conditional distributions of $\mu_n|X_n$ and $Y_n^r|X_n,\mu_n$, in (20), the condition expectations $\mathbb{E}(\ln\mu_n|X_n)=\psi\Big(\frac{\nu^{(i)}+d_cd_r}{2}\Big)-\ln\Big(\frac{\nu^{(i)}+\rho_n^{(i)}}{2}\Big)$ by [13] where $\psi(\cdot)$ is the digamma function, and

$$\begin{aligned}
&\mathbb{E}\Big(\operatorname{tr}\big(\mu_n\Sigma_r^{-1}[X_n-(CY_n^r+W)]^{\mathrm{T}}[X_n-(CY_n^r+W)]\big)\big|X_n\Big) \\
=&E_{\mu_n}^{(i)}\operatorname{tr}\Big(\Sigma_r^{-1}(X_n-W)^{\mathrm{T}}(X_n-W)\Big)-2E_{\mu_n}^{(i)}\operatorname{tr}\Big(\Sigma_r^{-1}(X_n-W)^{\mathrm{T}}CE_{Y_n^r}^{(i)}\Big) \\
&+\operatorname{tr}(\Sigma_r^{-1}\Sigma_r^{(i)})\operatorname{tr}\Big(C^{\mathrm{T}}C(\sigma_c^{(i)})^2(\Phi_c^{(i)})^{-1}\Big)+E_{\mu_n}^{(i)}\operatorname{tr}\Big(\Sigma_r^{-1}(E_{Y_n^r}^{(i)})^{\mathrm{T}}C^{\mathrm{T}}CE_{Y_n^r}^{(i)}\Big),
\end{aligned} \tag{22}$$

which is detailed in Appendix A.

In the subsequent conditional maximization (CM) step of the first stage, given the condition $\theta_r^{(i)}=\{\nu^{(i)},\sigma_r^{(i)},W^{(i)},R^{(i)}\}$, we maximize $Q_c(\theta_c)$ with respect to $\theta_c=\{\nu,\sigma_c,W,C\}$. It follows by (20) that

$$\begin{aligned}
\frac{\partial Q_c(\theta_c)}{\partial W} &= -\frac{1}{2\sigma_c^2}\sum_{n=1}^{N}E_{\mu_n}^{(i)}\Big((-2X_n+2W)(\Sigma_r^{(i)})^{-1}+2CE_{Y_n^r}^{(i)}(\Sigma_r^{(i)})^{-1}\Big), \\
\frac{\partial Q_c(\theta_c)}{\partial C} &= -\frac{1}{2\sigma_c^2}\sum_{n=1}^{N}\Big\{-2E_{\mu_n}^{(i)}(X_n-W)(\Sigma_r^{(i)})^{-1}(E_{Y_n^r}^{(i)})^{\mathrm{T}} \\
&\quad +2d_rC(\sigma_c^{(i)})^2(\Phi_c^{(i)})^{-1}+2E_{\mu_n}^{(i)}CE_{Y_n^r}^{(i)}(\Sigma_r^{(i)})^{-1}(E_{Y_n^r}^{(i)})^{\mathrm{T}}\Big\}, \\
\frac{\partial Q_c(\theta_c)}{\partial\sigma_c^2} &= -\frac{Nd_cd_r}{2\sigma_c^2}+\frac{1}{2\sigma_c^4}\sum_{n=1}^{N}\mathbb{E}\Big(\operatorname{tr}\big(\mu_n(\Sigma_r^{(i)})^{-1}[X_n-(CY_n^r+W)]^{\mathrm{T}}[X_n-(CY_n^r+W)]\big)\big|X_n\Big).
\end{aligned}$$

Therefore, by successively solving the equations $\frac{\partial Q_c(\theta_c)}{\partial W} = 0$, $\frac{\partial Q_c(\theta_c)}{\partial C} = 0$ and $\frac{\partial Q_c(\theta_c)}{\partial \sigma_c^2} = 0$, we can iteratively update the parameters W, C and σ_c. Specifically, by $\frac{\partial Q_c(\theta_c)}{\partial W} = 0$, we have

$$\sum_{n=1}^{N} E_{\mu_n}^{(i)}(-2X_n + 2W + 2CE_{Y_n^r}^{(i)})(\Sigma_r^{(i)})^{-1} = 0.$$

That means

$$\sum_{n=1}^{N} E_{\mu_n}^{(i)}(-X_n + CE_{Y_n^r}^{(i)}) + \sum_{n=1}^{N} E_{\mu_n}^{(i)} W = 0.$$

Thus, an iterative updating of W can be obtained by

$$\widetilde{W}^{(i)} = \frac{\sum_{n=1}^{N} E_{\mu_n}^{(i)}(X_n - C^{(i)}E_{Y_n^r}^{(i)})}{\sum_{n=1}^{N} E_{\mu_n}^{(i)}}. \tag{23}$$

Similarly, based on $\frac{\partial Q_c(\theta_c)}{\partial C} = 0$ and $\frac{\partial Q_c(\theta_c)}{\partial \sigma_c^2} = 0$, we have

$$\begin{aligned} C^{(i+1)} &= \sum_{n=1}^{N} E_{\mu_n}^{(i)}(X_n - \widetilde{W}^{(i)})(\Sigma_r^{(i)})^{-1}(E_{Y_n^r}^{(i)})^{\mathrm{T}} \\ &\quad \times \left[\sum_{n=1}^{N}\Big(d_r(\sigma_c^{(i)})^2(\Phi_c^{(i)})^{-1} + E_{\mu_n}^{(i)}E_{Y_n^r}^{(i)}(\Sigma_r^{(i)})^{-1}(E_{Y_n^r}^{(i)})^{\mathrm{T}}\Big)\right]^{-1}, \end{aligned} \tag{24a}$$

$$\begin{aligned} (\sigma_c^{(i+1)})^2 &= \frac{1}{Nd_cd_r}\sum_{n=1}^{N} \mathbb{E}\Big(\operatorname{tr}\big(\mu_n(\Sigma_r^{(i)})^{-1}(X_n - C^{(i+1)}Y_n^r - \widetilde{W}^{(i)})^{\mathrm{T}}(X_n - C^{(i+1)}Y_n^r - \widetilde{W}^{(i)})\big)\,|X_n\Big) \\ &= \frac{1}{Nd_cd_r}\sum_{n=1}^{N} E_{\mu_n}^{(i)} \operatorname{tr}\Big((\Sigma_r^{(i)})^{-1}(X_n - \widetilde{W}^{(i)})^{\mathrm{T}}(X_n - \widetilde{W}^{(i)} - C^{(i+1)}E_{Y_n^r}^{(i)})\Big). \end{aligned} \tag{24b}$$

The last equality (24b) holds because of (22) and (24a). Finally, we update $\nu^{(i)}$ by maximizing the scalar nonlinear function $Q_c(\theta_c)$ defined in (20) on v, which can be solved numerically by most scientific computation software packages [26,27], to obtain $\tilde{\nu}^{(i)}$.

In the second stage, the AECM algorithm is used for the model (18) to update θ_r. In such a case, we consider the latent variables' data $\{Y_n^c, \mu_n\}_{n=1}^N$ as "missing data". Then, by (19b), the "complete" data log-likelihood is

$$\begin{aligned} L_{com,r}(\theta_r) &= \sum_{n=1}^{N} \ln\Big(p(X_n, Y_n^c, \mu_n)\Big) = \sum_{n=1}^{N} \ln\Big(p(X_n|Y_n^c, \mu_n)p(Y_n^c|\mu_n)p(\mu_n)\Big) \\ &= \sum_{n=1}^{N} \ln\Bigg((2\pi)^{-\frac{d_cd_r}{2}}|\mu_{c,n}\Sigma_c|^{-\frac{d_r}{2}}|\mu_{r,n}\sigma_r^2 I_{d_r}|^{-\frac{d_c}{2}} \operatorname{etr}\Big(-\frac{1}{2}(\mu_{r,n}\sigma_r^2 I)^{-1}[X_n - (Y_n^c R^{\mathrm{T}} \\ &\quad + W)]^{\mathrm{T}}(\mu_{c,n}\Sigma_c)^{-1}[X_n - (Y_n^c R^{\mathrm{T}} + W)]\Big)(2\pi)^{-\frac{d_cq_r}{2}}|\mu_{c,n}\Sigma_c|^{-\frac{q_r}{2}}|\mu_{r,n}I_{q_r}|^{-\frac{d_c}{2}} \\ &\quad \times \operatorname{etr}\Big(-\frac{1}{2}(\mu_{r,n}I_{q_r})^{-1}(Y_n^c)^{\mathrm{T}}(\mu_{c,n}\Sigma_c)^{-1}Y_n^c\Big)\Big(\frac{\nu}{2}\Big)^{\frac{\nu}{2}}\mu_n^{\frac{\nu}{2}-1}\exp\Big(-\frac{\nu}{2}\mu_n\Big)\frac{1}{\Gamma(\frac{\nu}{2})}\Bigg) \\ &= -\sum_{n=1}^{N}\Bigg\{\frac{(d_r+q_r)d_c}{2}\ln(2\pi) + \ln\Gamma\Big(\frac{\nu}{2}\Big) + \frac{d_r+q_r}{2}\ln|\Sigma_c| - \frac{\nu}{2}\ln\Big(\frac{\nu}{2}\Big) + \frac{\nu\mu_n}{2} \\ &\quad - \frac{(d_r+q_r)d_c+\nu-2}{2}\ln\mu_n + \frac{d_cd_r}{2}\ln\sigma_r^2 + \operatorname{tr}\Big(\frac{\mu_n}{2\sigma_r^2}[X_n - (Y_n^cR^{\mathrm{T}}+W)]^{\mathrm{T}}\Sigma_c^{-1} \\ &\quad \times [X_n - (Y_n^cR^{\mathrm{T}}+W)] + \frac{\mu_n}{2}(Y_n^c)^{\mathrm{T}}\Sigma_c^{-1}Y_n^c\Big)\Bigg\}. \end{aligned}$$

Similarly, in E-step of the second stage, given the updated parameter set

$$\tilde{\theta}^{(i)} = \{\tilde{\nu}^{(i)}, \sigma_c^{(i+1)}, \sigma_r^{(i)}, \widetilde{W}^{(i)}, C^{(i+1)}, R^{(i)}\},$$

where $\tilde{\nu}^{(i)}$, $\sigma_c^{(i+1)}$, $\widetilde{W}^{(i)}$ and $C^{(i+1)}$ are calculated from the first stage, we compute the expectation of $L_{com,r}(\theta_r)$ with respect to the condition distribution $Y_n^c, \mu_n | X_n$, denoted by $Q_r(\theta_r)$. Based on (19) and the current parameter set $\tilde{\theta}^{(i)}$, we define

$$\Sigma_c^{(i+1)} = C^{(i+1)}(C^{(i+1)})^{\mathrm{T}} + (\sigma_c^{(i+1)})^2 I_{d_c}, \tag{25a}$$

$$\tilde{\rho}_n^{(i)} = \mathrm{tr}\left((\Sigma_c^{(i+1)})^{-1}(X_n - \widetilde{W}^{(i)})(\Sigma_r^{(i)})^{-1}(X_n - \widetilde{W}^{(i)})^{\mathrm{T}}\right), \tag{25b}$$

$$\widetilde{E}_{\mu_n}^{(i)} = \mathbb{E}(\mu_n | X_n) = \frac{\tilde{\nu}^{(i)} + d_c d_r}{\tilde{\nu}^{(i)} + \tilde{\rho}_n^{(i)}}, \tag{25c}$$

$$\Phi_r^{(i)} = (R^{(i)})^{\mathrm{T}} R^{(i)} + (\sigma_r^{(i)})^2 I_{q_r}, \tag{25d}$$

$$E_{Y_n^c}^{(i)} = \mathbb{E}(Y_n^c | X_n, \mu_n) = (X_n - \widetilde{W}^{(i)}) R^{(i)} (\Phi_r^{(i)})^{-1}. \tag{25e}$$

We have, up to a constant,

$$Q_r(\theta_r) = -\sum_{n=1}^{N} \Bigg\{ \ln\Gamma\left(\frac{\nu}{2}\right) - \frac{\nu}{2}\ln\left(\frac{\nu}{2}\right) + \frac{\nu}{2}\widetilde{E}_{\mu_n}^{(i)} - \frac{(d_r + q_r)d_c + \nu - 2}{2}\mathbb{E}(\ln\mu_n | X_n) + \frac{d_c d_r}{2}$$
$$\ln\sigma_r^2 + \frac{1}{2\sigma_r^2}\mathbb{E}\Big(\mathrm{tr}\left(\mu_n [X_n - (Y_n^c R^{\mathrm{T}} + W)]^{\mathrm{T}} \Sigma_c^{-1} [X_n - (Y_n^c R^{\mathrm{T}} + W)]\right) \big| X_n\Big) \Bigg\}, \tag{26}$$

where $\mathbb{E}(\ln\mu_n | X_n) = \psi\left(\frac{\tilde{\nu}^{(i)} + d_c d_r}{2}\right) - \ln\left(\frac{\tilde{\nu}^{(i)} + \tilde{\rho}_n^{(i)}}{2}\right)$ and

$$\mathbb{E}\Big(\mathrm{tr}\left(\mu_n [X_n - (Y_n^c R^{\mathrm{T}} + W)]^{\mathrm{T}} \Sigma_c^{-1} [X_n - (Y_n^c R^{\mathrm{T}} + W)]\right) \big| X_n\Big)$$
$$= \widetilde{E}_{\mu_n}^{(i)} \,\mathrm{tr}\Big((X_n - W)^{\mathrm{T}} \Sigma_c^{-1} (X_n - W)\Big) - 2\widetilde{E}_{\mu_n}^{(i)} \,\mathrm{tr}\left((X_n - W)^{\mathrm{T}} \Sigma_c^{-1} E_{Y_n^c}^{(i)} R^{\mathrm{T}}\right)$$
$$+ \mathrm{tr}(\Sigma_c^{-1} \Sigma_c^{(i+1)})\,\mathrm{tr}\Big(R^{\mathrm{T}} R (\sigma_r^{(i)})^2 (\Phi_r^{(i)})^{-1}\Big) + \widetilde{E}_{\mu_n}^{(i)} \,\mathrm{tr}\Big(R^{\mathrm{T}} R (E_{Y_n^c}^{(i)})^{\mathrm{T}} \Sigma_c^{-1} E_{Y_n^c}^{(i)}\Big). \tag{27}$$

See Appendix A for the derivation of (27). At last, in CM-step of the second stage, based on $\tilde{\theta}_c^{(i)} = \{\tilde{\nu}^{(i)}, \sigma_c^{(i+1)}, \widetilde{W}^{(i)}, C^{(i+1)}\}$, similarly to (23) and (24), we maximize $Q_r(\theta_r)$ with respect to θ_r to update

$$W^{(i+1)} = \frac{\sum_{n=1}^{N} \widetilde{E}_{\mu_n}^{(i)} \left(X_n - E_{Y_n^c}^{(i)} (R^{(i)})^{\mathrm{T}}\right)}{\sum_{n=1}^{N} \widetilde{E}_{\mu_n}^{(i)}}, \tag{28a}$$

$$R^{(i+1)} = \sum_{n=1}^{N} \widetilde{E}_{\mu_n}^{(i)} (X_n - W^{(i+1)})^{\mathrm{T}} (\Sigma_c^{(i+1)})^{-1} E_{Y_n^c}^{(i)}$$
$$\times \left[\sum_{n=1}^{N} \left(d_c (\sigma_r^{(i)})^2 (\Phi_r^{(i)})^{-1} + \widetilde{E}_{\mu_n}^{(i)} (E_{Y_n^c}^{(i)})^{\mathrm{T}} (\Sigma_c^{(i+1)})^{-1} E_{Y_n^c}^{(i)}\right)\right]^{-1}, \tag{28b}$$

$$(\sigma_r^{(i+1)})^2 = \frac{1}{N d_c d_r} \sum_{n=1}^{N} \widetilde{E}_{u_n}^{(i)} \,\mathrm{tr}\Big((X_n - W^{(i+1)})^{\mathrm{T}} (\Sigma_c^{(i+1)})^{-1} \Big(X_n - W^{(i+1)} - E_{Y_n^c}^{(i)} (R^{(i+1)})^{\mathrm{T}}\Big)\Big), \tag{28c}$$

and then solve the scalar nonlinear maximization problem (26) on ν to get $\nu^{(i+1)}$.

We summarize what we do in this subsection in Algorithm 1. A few remarks regarding Algorithm 1 are in order:

(1) In Algorithm 1, it is not necessarily to explicitly compute $\Sigma_c^{(i)}$ and $\Sigma_r^{(i)}$. The reason is that the calculation of $(\Sigma_c^{(i)})^{-1}$ and $(\Sigma_r^{(i)})^{-1}$ can be more efficiently performed by using

$$(\Sigma_c^{(i)})^{-1} = \frac{I_{d_c} - C^{(i)}(\Phi_c^{(i)})^{-1}(C^{(i)})^{\mathrm{T}}}{(\sigma_c^{(i)})^2}, \quad (\Sigma_r^{(i)})^{-1} = \frac{I_{d_r} - R^{(i)}(\Phi_r^{(i)})^{-1}(R^{(i)})^{\mathrm{T}}}{(\sigma_r^{(i)})^2}.$$

In its per-iteration of Algorithm 1, the most expensive computational cost is $\mathcal{O}(Nd_cd_r(d_c+d_r))$ appearing on the formation of $\rho_n^{(i)}$ with $1 \le n \le N$. Owing to introducing the new latent variable μ_n, RBPPCA is a little more time complex than BPPCA having a calculation cost of $\mathcal{O}(Nd_cd_r(q_c+q_r))$. However, it will be shown in our numerical example that the RBPPCA algorithm presents less sensitivity to outliers.

(2) Compared with the AECM algorithm of BPPCA in [12] which uses the centered data and estimates $\{\sigma_c, C\}$ and $\{\sigma_r, R\}$ based on the model $X = CY^r + Y_\epsilon^r$ and $X = Y^cR^{\mathrm{T}} + Y_\epsilon^c$, respectively, two more parameters ν and W are needed to be computed in the AECM iteration of RBPPCA. Notice that both the models (14) and (18) contain the parameters ν and W. Thus, we split the parameter set θ into $\theta_c = \{\nu, \sigma_c, W, C\}$ and $\theta_r = \{\nu, \sigma_r, W, R\}$ which naturally leads to the parameters ν and W being calculated twice in each loop of Algorithm 1. Though other partitions of the set θ, such as $\{\nu, \sigma_c, C\}$ and $\{\sigma_r, W, R\}$, are also available for the estimation of parameters, we prefer θ_c and θ_r, because updating ν and W one more time in each iteration can be obtained by adding a little more computational cost.

(3) As stated in Section 3.3 of [24], any AECM sequence increases $L(\theta) = \sum_{n=1}^N \ln(p(X_n|\theta))$ at each iteration, and converges to a stationary point of $L(\theta)$. Notice that the convergence results of the AECM algorithm proved in Section 3.3 of [24] do not depend on the distributions of the data sets. Therefore, up to set a limit on the maximum number of steps $iter_{max}$, we use the following relative change of log-likelihood as the stopping criterion, i.e.,

$$\zeta = \left|1 - \frac{L(\theta^{(i+1)})}{L(\theta^{(i)})}\right| = \left|1 - \frac{\sum_{n=1}^N \ln(p(X_n|\theta^{(i+1)}))}{\sum_{n=1}^N \ln(p(X_n|\theta^{(i)}))}\right| \le \varepsilon \tag{29}$$

where ε is a specified tolerance used, which by default is set to 10^{-5} in our numerical examples.

(4) Based on the computed results of Algorithm 1, and similar to PPCA [8] and BPPCA [12], it is known that

$$\mathbb{E}(Z|X) = \mathbb{E}\big(\mathbb{E}(Z|Y^r)\big|X\big) = \Phi_c^{-1}C^{\mathrm{T}}(X-W)R\Phi_r^{-1}$$

can be considered as the compressed representation of X. Hence, we can reconstruct X as

$$\widehat{X} = C\mathbb{E}(Z|X)R^{\mathrm{T}} + W = C\Phi_c^{-1}C^{\mathrm{T}}(X-W)R\Phi_r^{-1}R^{\mathrm{T}} + W. \tag{30}$$

Algorithm 1 Robust bilinear probabilistic PCA algorithm (RBPPCA).

```
Input: Initialization $\nu^{(0)}, \sigma_c^{(0)}, \sigma_r^{(0)}, W^{(0)}, C^{(0)}, R^{(0)}$, and sample matrices $\{X_n\}_{n=1}^N$. Compute $\Phi_c^{(0)}$.
Output: the converged $\{\nu, \sigma_c, \sigma_r, W, C, R\}$
  for $i = 0, 1, 2, \ldots$, until $\zeta$ defined in (29) is smaller than a threshold do
    % Stage 1
    Compute $\Phi_r^{(i)}, \rho_n^{(i)}, E_{\mu_n}^{(i)}$ and $E_{Y_n^r}^{(i)}$ via (21).
    Compute $\widetilde{W}^{(i)}, C^{(i+1)}$ and $\sigma_c^{(i+1)}$ by (23) and (24).
    Solve the scalar nonlinear maximization problem (20) to obtain $\tilde{\nu}^{(i)}$.
    % Stage 2
    Compute $\Phi_c^{(i+1)}, \tilde{\rho}_n^{(i)}, \widetilde{E}_{\mu_n}^{(i)}$ and $E_{Y_n^c}^{(i)}$ via (25).
    Compute $W^{(i+1)}, R^{(i+1)}$ and $\sigma_r^{(i+1)}$ by (28).
    Solve the scalar nonlinear maximization problem (26) to obtain $\nu^{(i+1)}$.
  end for
```

4. Numerical Examples

In this section, we conduct several numerical examples based on synthetic problems and three real-world data sets to demonstrate the effectiveness of our proposed RBPPCA algorithm. All experiments were run by using MATLAB (2016a) with machine epsilon 2.22×10^{-16} on a Windows 10 (64 bit) Laptop with an Intel Core i7-8750H CPU (2.20GHz) and 8GB memory. Each random experiment was repeated 20 times independently, then the average numerical results were reported.

Example 1 (Experiments on the synthetic data). *In this example, we only compare ours with the BPPCA algorithm [12] to illustrate the significant improvement of the RBPPCA algorithm. We take N data matrices X_n for $n = 1, \ldots, N$ with $N = 200$ and $d_c = d_r = 64$, N_g of which are generated by*

$$X_n = CZ_nR^{\mathrm{T}} + W + C\mathcal{E}_{r_n} + \mathcal{E}_{c_n}R^{\mathrm{T}} + \mathcal{E}_n, \quad n = 1, \ldots, N_g,$$

where C, R and W are simply synthesized by MATLAB as

$$C = \mathrm{eye}(d_c, q_c),\ R = \mathrm{eye}(d_r, q_r),\ W = \mathrm{rand}(d_c, d_r) \text{ with } q_c = q_r = 8,$$

Z_n, $\mathcal{E}_{r_n}$, $\mathcal{E}_{c_n}$ and $\mathcal{E}_n$ are sampled from matrix variate normal distributions $\mathcal{N}_{q_c,q_r}(0_{q_c\times q_r}, I_{q_c}, I_{q_r})$, $\mathcal{N}_{q_c,d_r}(0_{q_c\times d_r}, I_{q_c}, \sigma_r^2 I_{d_r})$, $\mathcal{N}_{d_c,q_r}(0_{d_c\times q_r}, \sigma_c^2 I_{d_c}, I_{q_r})$, and $\mathcal{N}_{d_c,d_r}(0_{d_c\times d_r}, \sigma_c^2 I_{d_c}, \sigma_r^2 I_{d_r})$ with $\sigma_c = \sigma_r = 1$, respectively. The other N_o data matrices, i.e., X_n for $n = N_g + 1, \ldots, N$, are regarded as outliers of which each entry is sampled from the uniform distribution over the range of 0 to 10. In order to demonstrate the quality of computed approximations $C^{(i)}$ and $R^{(i)}$, we calculate the arc length distance between the two subspaces $\mathrm{span}(R \otimes C)$ and $\mathrm{span}(R^{(i)} \otimes C^{(i)})$, which is used in [12] and defined as

$$\arcsin\left(\left\|(I_{d_c d_r} - QQ^{\mathrm{T}})Q^{(i)}\right\|_2\right), \tag{31}$$

to monitor the numerical performance of the RBPPCA and BPPCA method, where $\mathrm{span}(R \otimes C)$ and $\mathrm{span}(R^{(i)} \otimes C^{(i)})$ are the column space of $R \otimes C$ and $R^{(i)} \otimes C^{(i)}$, respectively, and Q and $Q^{(i)}$ are the orthogonal base matrices of $\mathrm{span}(R \otimes C)$ and $\mathrm{span}(R^{(i)} \otimes C^{(i)})$, respectively. In fact, by ([28], Definition 4.2.1), the computational result of (31) is the largest canonical angle between the estimated subspace $\mathrm{span}(R^{(i)} \otimes C^{(i)})$ and the true $\mathrm{span}(R \otimes C)$.

In this test, we start with $C^{(0)} = \mathrm{rand}(d_c, q_c)$, $R^{(0)} = \mathrm{rand}(d_r, q_r)$, $\sigma_c^{(0)} = 1$, $\sigma_r^{(0)} = 1$ and $\nu^{(0)} = 1$, then consider the effect of the ratio of outliers, i.e., N_o/N, which is varied from 0 to 30% with a stride length of 10% in this example. In these cases, the estimated values of ν are all $\nu = 1$. If we use other initial values $\nu^{(0)}$ here, the computed ν also converges to one as iterations increase. The corresponding numerical results of arc length distances are plotted in Figure 1. Figure 1 shows that the RBPPCA and BPPCA methods almost perform with the same convergence behavior when the data matrices are without outliers.

As the ratio of outliers goes to 30%, it is reasonable that more iterations are required for the RBPPCA method to achieve a satisfactory accuracy. Unlike the BPPCA method, the presented RBPPCA method is more robust to outliers because the arc length distances of the BPPCA method are always held to approximately 1.5 when the data includes outliers.

In this example, we also test the impact of the initializations on $C^{(0)}$ and $R^{(0)}$, and the sample size N, respectively, based on the synthetic data having 10% outliers. Three different types of initializations of $C^{(0)}$ and $R^{(0)}$ are set as follows:

(1) initialization 1: $C^{(0)} = \mathrm{rand}(d_c, q_c)$ and $R^{(0)} = \mathrm{rand}(d_r, q_r)$;
(2) initialization 2: $C^{(0)} = \mathrm{randn}(d_c, q_c)$ and $R^{(0)} = \mathrm{randn}(d_r, q_r)$;

(3) initialization 3:

$$C^{(0)} = R^{(0)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{9}{d_c} & \frac{9}{d_c}+9 & \sin(9) & \cos(9) & \tan(9) & \cot(9) & \sec(9) & \csc(9) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{d_c}{d_c} & \frac{d_c}{d_c}+d_c & \sin(d_c) & \cos(d_c) & \tan(d_c) & \cot(d_c) & \sec(d_c) & \csc(d_c) \end{bmatrix},$$

and other parameters are fixed to be the same. The results associated with the different initializations are shown in Figure 2. Inspection of the plot illustrates that our RBPPCA method appears to be insensitive to different initializations, since the convergence behaviors of the RBPPCA method based on different initializations do not have a significant difference. Figure 3 presents the required CPU time in seconds and the quantities of arc length distances of the BPPCA and RBPPCA methods with the number of iterations being 25 with respect to the number of samples, where the sample size N varies from 200 to 5000 with a stride length of 200. Such graphs covey the fact that with the increase of the sample size N, the BPPCA and RBPPCA methods both required more CPU time for 25 iterations, and the BPPCA method needs less time complexity than RBPPCA as we stated in the remarks of Algorithm 1. However, the bigger N does not lead to the improved arc length distances of the BPPCA method.

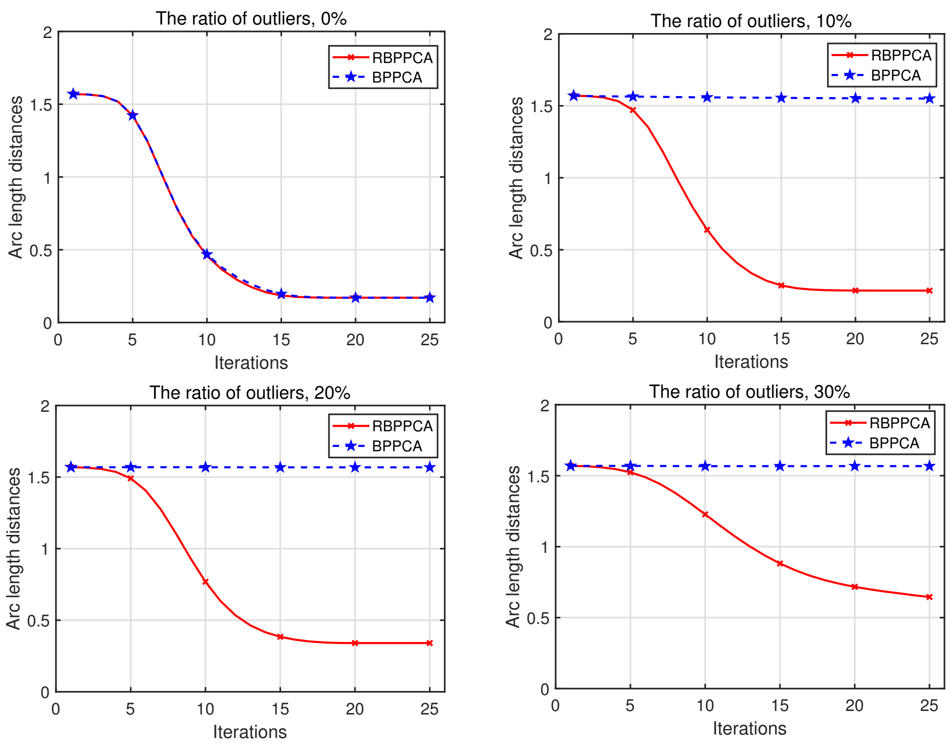

Figure 1. Convergence behaviors of RBPPCA and BPPCA with the ratio of outliers being 0%, 10%, 20% and 30%, respectively.

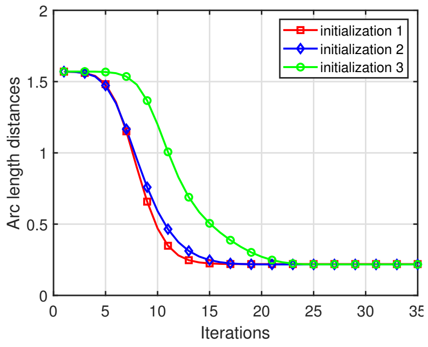

Figure 2. Convergence behaviors of RBPPCA for three different types of initialization.

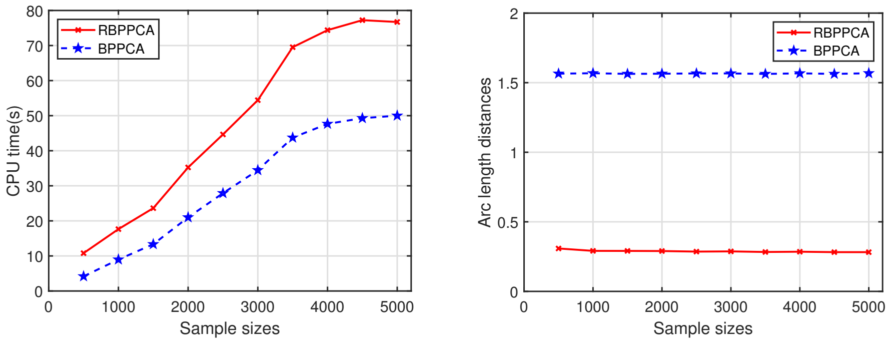

Figure 3. CPU time in seconds (**left**) and arc length distances (**right**) of BPPCA and RBPPCA with the number of iterations being 25 with the sample size N from 200 to 5000.

Example 2 (**Experiments on face image databases**)**.** *In this example, all of the experiments are based on two publicly available face image databases: the Yale face database (Available from http://vision.ucsd.edu/content/yale-face-database) (accessed on 22 September 2021), and the Yale B face database (Available from http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html) (accessed on 22 September 2021).The Yale face database includes different face orientations, facial expression, and whether there exist glasses, and the Yale B face database is collected under different illumination conditions. We select 265 face images of 15 individuals for each database. Each person has 11 images, and these images are cropped to* 64×64 *pixels. For each individual, eight images are randomly selected as the training set, and the rest are the test set. Then, we randomly select two and four images from the eight images in the training set to be corrupted as outliers, respectively. Half of the corrupted images are generated by replacing part of the original image with a* 30×30 *rectangle of noise, and the other half of corrupted images use a* 50×50 *rectangle of noise to replace it. Within each rectangle, the pixel value comes from a uniform distribution on the interval* $[0, 255]$*. Then, we rescale the images from the range of [0, 255] to the range [0, 1]. Some original and corrupted images of the Yale and Yale B databases are shown in Figures 4–7, respectively. The iteration of BPPCA and RBPPCA is stopped when their corresponding relative changes of the log-likelihood defined in* (29) *are smaller than* 10^{-5}*.*

We run the BPPCA and RBBPCA algorithms for the original data set and the corrupted data set, respectively, with the order of Z in (11) being $q_c = q_r = 8$, and consider the reconstructed images $\widehat{X}_n$ defined in (30) based on the computed results. A comparison of the reconstructed images based on the original data set and corrupted data set with two corrupted images for each individual is shown in Figure 8. In Figure 8, the first, second and third columns are the original images, the reconstructed images of RBPPCA, and the reconstructed images of BPPCA for the Yale (left) and Yale B (right) databases, respectively, and the images of the first, second and third rows are shown based on the original, corrupted data sets with 30×30 rectangles of noise, and corrupted data sets with 50×50 rectangles of noise, respectively. The BPPCA and RBPPCA almost perform the same as the reconstructed images on the original data set. However, for corrupted data sets, the reconstructed performance of RBPPCA presents better images than BPPCA because it tries to explain noise information.

We also compare the average reconstruction errors $\eta = \sum_{n=1}^{N} \|X_n - \widehat{X}_n\|_{\mathrm{F}}/N$ and the recognition accuracy rates for the RBPPCA and BPPCA algorithms in the cases where each person has two corrupted images and four corrupted images, respectively, where recognition accuracy rates are calculated based on the nearest neighbor classifier (1-NN) which is employed for the classification. The average reconstruction errors and recognition accuracy rates versus the order of Z are plotted in Figures 9 and 10, respectively. As expected, the average reconstruction errors decrease, while the recognition accuracy rates rise as the order of Z increases. In these cases, our proposed RBPPCA algorithm outperforms the BPPCA algorithm in reducing average reconstruction errors and enhancing the recognition accuracy. In addition, another advantage of robust probabilistic algorithms based on t distributions is outlier detection. By [14], we can compute $\rho_n = [\mathrm{vec}(X_n - M)]^{\mathrm{T}}(\Sigma_r \otimes \Sigma_c)^{-1}\,\mathrm{vec}(X_n - M)$ as the standard of outlier detection. Figure 11 is the scatter chart for ρ_n of all the images in the training set of the Yale and Yale B databases with two corrupted images for one person, respectively. It is exhibited that the quantity of ρ_n can be divided into three parts. Notice that these three parts correspond to the images with no noise, a 30×30 rectangle of noise, and a 50×50 rectangle of noise, respectively. Hence, the comparison of ρ_n provides a method for judging the outliers.

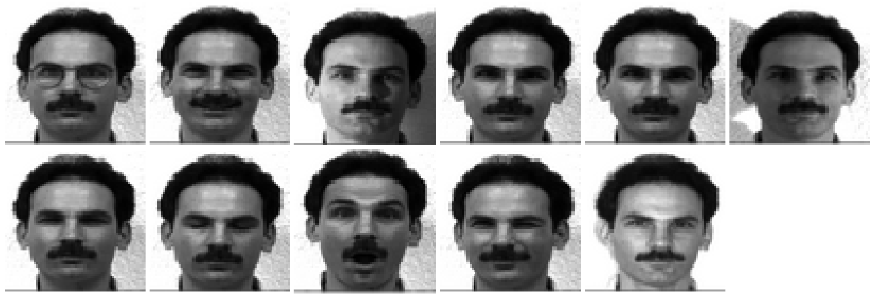
Figure 4. One set of processed samples from the Yale face database.

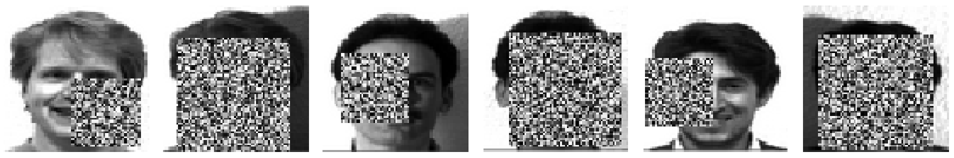
Figure 5. Some generated outlier face images in the training set from the Yale face database.

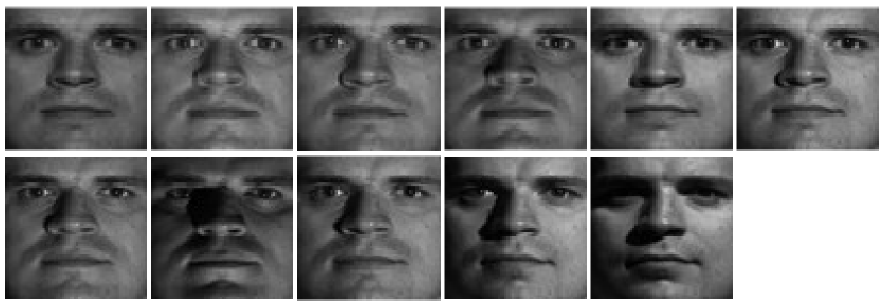
Figure 6. Eleven images of an individual from the Yale B face database.

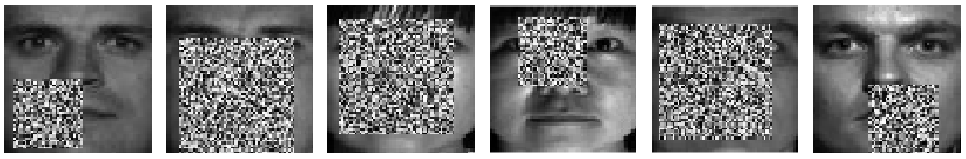

Figure 7. Some corrupted face images in the training set of the Yale B face database.

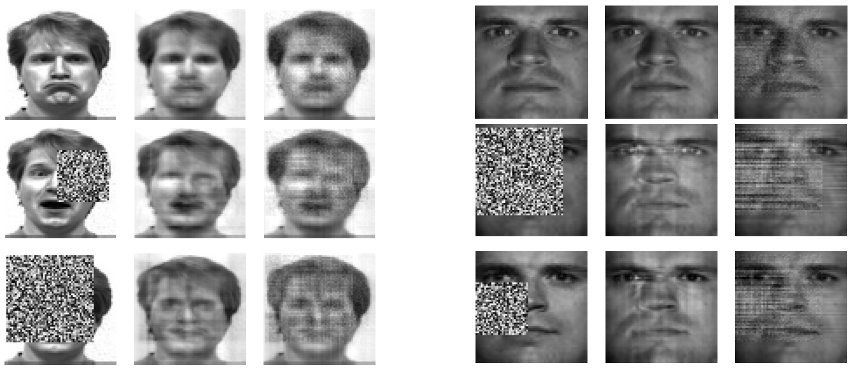

Figure 8. Original images (the first column), reconstructed images by RBPPCA (the second column), and reconstructed images by BPPCA (the third column) of the Yale (**left**) and Yale B (**right**) databases, respectively.

Example 3 (Experiments on the MNIST dataset). *In Example 2, it is shown that RBPPCA is superior to BPPCA when the data sets contain outliers. In this example, we compare the RBPPCA algorithm to the tPPCA [14] and L1-PPCA [15] algorithms based on handwritten digit images from the MNIST (Available from http://yann.lecun.com/exdb/mnist) (accessed on 22 September 2021) database in which each image has* 28×28 *pixels. We choose 59 images of the digit 4 as the training data set, and randomly select nine of them to be corrupted as outliers. The way of corrupting the images is to add noise from a uniform distribution on the interval* $[0, 510]$*, and then normalize all images to the range* $[0, 1]$*. The normalized corrupted images of the digit 4 are shown in Figure 12. The RBPPCA, tPPCA [14] and L1-PPCA [15] algorithms are implemented with 100 iterations for the original data set of the digit 4 and the corrupted data set, respectively.*

Figure 13 presents the average reconstruction errors of the RBBPCA, tPPCA and L1-PPCA algorithms where the feature number is the order of Z for the RBBPCA algorithm and is the dimension of the low-dimensional representation for the tPPCA and L1-PPCA algorithms. It is observed that the performance of our RBPPCA algorithm is superior to other algorithms. The reconstructed behaviors of different algorithms based on the original data set of the digit 4 and the corrupted data set with $q_c = q_r = 1$ are shown in Figure 14. In Figure 14, the first column is the original images, and the second, third and fourth columns are the reconstructed images by the RBBPCA, tPPCA and L1-PPCA algorithms, respectively. As shown in Figure 14, compared to the tPPCA and L1-PPCA algorithms, the RBPPCA algorithm performs better reconstruction outcomes in such a case.

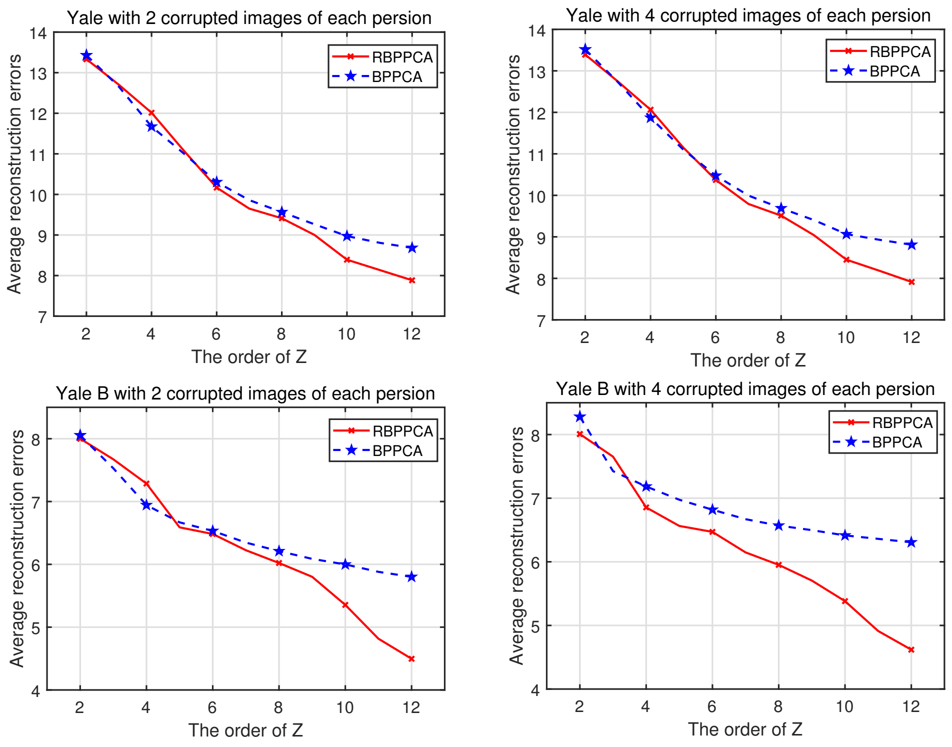

Figure 9. Average reconstruction errors of the BPPCA and RBPPCA algorithms vs. the order of Z for the Yale and Yale B databases with 2 and 4 corrupted images of each individual, respectively.

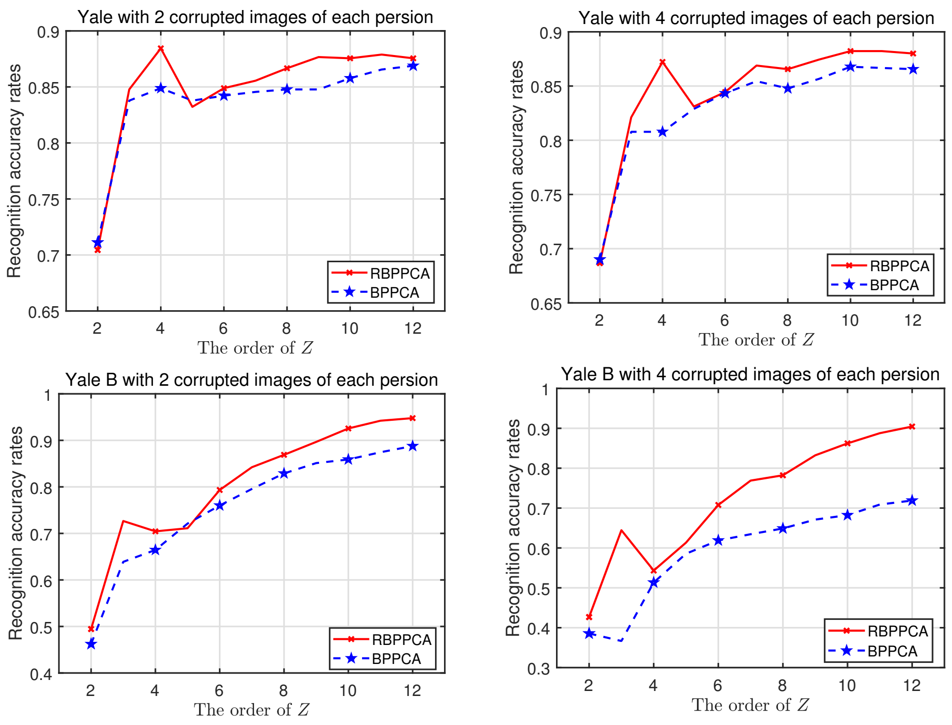

Figure 10. Recognition accuracy rates of the BPPCA and RBPPCA algorithms vs. the order of Z for the Yale and Yale B databases with 2 and 4 corrupted images of each individual, respectively.

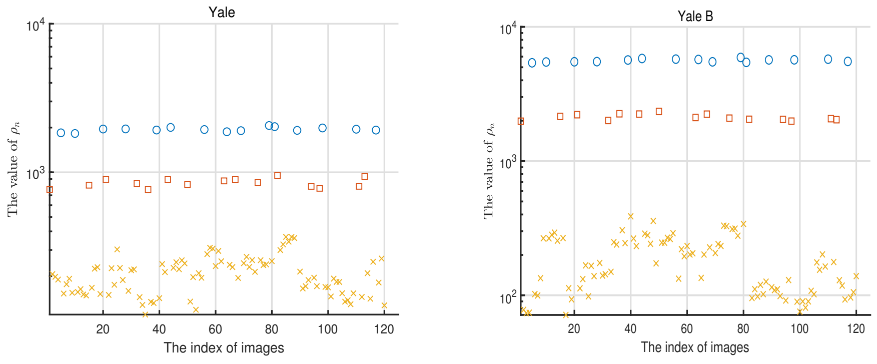

Figure 11. ρ_n of each image in the Yale (**left**) and Yale B (**right**) databases, respectively, with 2 corrupted images of each individual.

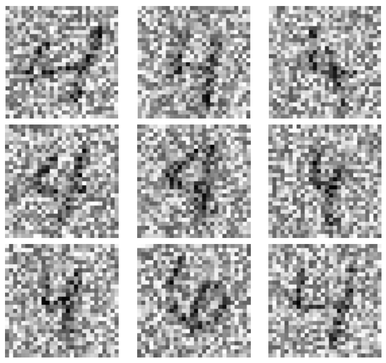

Figure 12. The normalized corrupted images of the digit 4.

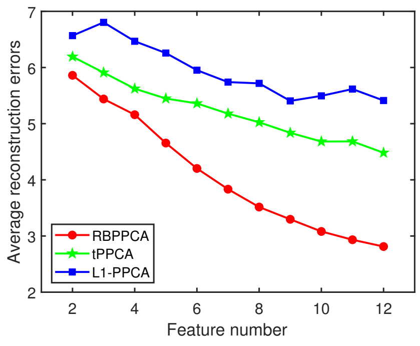

Figure 13. Average reconstruction errors of the RBPPCA, tPPCA and L1-PPCA algorithms vs. feature number on the MINST database.

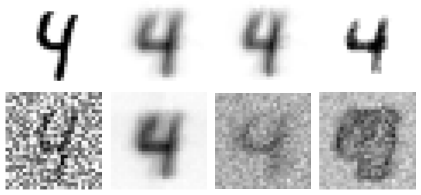

Figure 14. Original images of the digit 4 (**the first column**), images reconstructed by RBPPCA (**the second column**), images reconstructed by tPPCA (**the third column**), and images reconstructed by L1-PPCA (**the fourth column**). The images shown in the first and second rows are based on the original and corrupted image data sets, respectively.

5. Conclusions

To remedy the problem that data are assumed to follow a matrix variate Gaussian distribution which is sensitive to outliers, in this paper, we proposed a robust BPPCA algorithm (RBPPCA), i.e., Algorithm 1, by replacing the matrix variate Gaussian distribution with the matrix variate *t* distribution for noise. Compared to BPPCA, owing to the matrix variate *t* distribution having a significantly heavy tail property, our proposed RBPPCA method combined with AECM for estimating parameters can deal with 2-D data sets in the presence of outliers. The numerical examples based on a synthetic and two publicly available real data sets, Yale and Yale B, are presented to state that Algorithm 1 is far superior to the BPPCA algorithm in computational accuracy, reconstruction performance, average reconstruction errors, recognition accuracy rates, and outlier detection. It is also shown by numerical examples based on the MNIST database that our RBPPCA method outperforms the tPPCA and L1-PPCA algorithms.

Author Contributions: Writing—original draft, Y.L.; Writing—review and editing, Z.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the research fund for distinguished young scholars of Fujian Agriculture and Forestry University No. xjq201727, and the science and technology innovation special fund project of Fujian Agriculture and Forestry University No. CXZX2020105A.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Derivations for (22) and (27) in Section 3

We first consider the conditional expectation of $\mu_n Y_n^r$ with respect to the condition distribution $Y_n^r, \mu_n | X_n$, which will be applied in our derivations of (22). That is

$$\begin{aligned}\mathbb{E}(\mu_n Y_n^r | X_n) &= \int_{Y_n^r} \int_{\mu_n} \mu_n Y_n^r p(Y_n^r, \mu_n | X_n) d\mu_n dY_n^r \\ &= \int_{Y_n^r} \int_{\mu_n} \mu_n Y_n^r p(Y_n^r | X_n, \mu_n) p(\mu_n | X_n) d\mu_n dY_n^r \\ &= \int_{\mu_n} \mu_n p(\mu_n | X_n) \int_{Y_n^r} Y_n^r p(Y_n^r | X_n, \mu_n) dY_n^r d\mu_n \\ &= \int_{\mu_n} \mu_n p(\mu_n | X_n) \mathbb{E}(Y_n^r | X_n, \mu_n) d\mu_n \\ &= \mathbb{E}(\mu_n | X_n) \mathbb{E}(Y_n^r | X_n, \mu_n) = E_{\mu_n}^{(i)} E_{Y_n^r}^{(i)}, \end{aligned} \tag{A1}$$

where $E^{(i)}_{\mu_n}$ and $E^{(i)}_{Y^r_n}$ are given by (21). In addition, for symmetric matrices $A \in \mathbb{R}^{d_r \times d_r}$ and $B \in \mathbb{R}^{q_c \times q_c}$, we have

$$\begin{aligned}
&\mathbb{E}\Big(\mathrm{tr}(\mu_n A(Y^r_n)^{\mathrm{T}} B Y^r_n)|X_n\Big)\\
=&\int_{Y^r_n}\int_{\mu_n} \mathrm{tr}\Big(\mu_n A(Y^r_n)^{\mathrm{T}} B Y^r_n\Big) p(Y^r_n|X_n,\mu_n)p(\mu_n|X_n)d\mu_n dY^r_n\\
=&\int_{\mu_n} \mu_n p(\mu_n|X_n)\int_{Y^r_n} \mathrm{tr}\Big(A(Y^r_n)^{\mathrm{T}} B Y^r_n\Big) p(Y^r_n|X_n,\mu_n)dY^r_n d\mu_n\\
=&\int_{\mu_n} \mu_n p(\mu_n|X_n)\int_{Y^r_n} [\mathrm{vec}(Y^r_n)]^{\mathrm{T}}(A\otimes B)\,\mathrm{vec}(Y^r_n) p(Y^r_n|X_n,\mu_n)dY^r_n d\mu_n\\
=&\int_{\mu_n} \mu_n p(\mu_n|X_n)\int_{Y^r_n} \mathrm{tr}\Big((A\otimes B)\,\mathrm{vec}(Y^r_n)[\mathrm{vec}(Y^r_n)]^{\mathrm{T}}\Big) p(Y^r_n|X_n,\mu_n)dY^r_n d\mu_n\\
=&\int_{\mu_n} \mu_n p(\mu_n|X_n)\,\mathrm{tr}\Bigg((A\otimes B)\int_{\mathrm{vec}(Y^r_n)} \mathrm{vec}(Y^r_n)[\mathrm{vec}(Y^r_n)]^{\mathrm{T}} p(\mathrm{vec}(Y^r_n)|X_n,\mu_n)d\,\mathrm{vec}(Y^r_n)\Bigg)d\mu_n\\
=&\int_{\mu_n} \mu_n p(\mu_n|X_n)\,\mathrm{tr}\Big((A\otimes B)\Big\{\Big[\mu_n^{-1}\Sigma^{(i)}_r\otimes(\sigma^{(i)}_c)^2(\Phi^{(i)}_c)^{-1}\Big]+\mathrm{vec}(E^{(i)}_{Y^r_n})[\mathrm{vec}(E^{(i)}_{Y^r_n})]^{\mathrm{T}}\Big\}\Big)d\mu_n\\
=&\int_{\mu_n} \mu_n p(\mu_n|X_n)\,\mathrm{tr}(\mu_n^{-1}A\Sigma^{(i)}_r\otimes B(\sigma^{(i)}_c)^2(\Phi^{(i)}_c)^{-1})d\mu_n+\int_{\mu_n} \mu_n p(\mu_n|X_n)\,\mathrm{tr}\Big(A(E^{(i)}_{Y^r_n})^{\mathrm{T}} B E^{(i)}_{Y^r_n}\Big)d\mu_n\\
=&\,\mathrm{tr}(A\Sigma^{(i)}_r)\,\mathrm{tr}\Big(B(\sigma^{(i)}_c)^2(\Phi^{(i)}_c)^{-1}\Big)+E^{(i)}_{\mu_n}\,\mathrm{tr}\Big(A(E^{(i)}_{Y^r_n})^{\mathrm{T}} B E^{(i)}_{Y^r_n}\Big).
\end{aligned} \tag{A2}$$

Notice that (22) can be rewritten as

$$\begin{aligned}
&\mathbb{E}\Big(\mathrm{tr}\,(\mu_n\Sigma_r^{-1}[X_n-(CY^r_n+W)]^{\mathrm{T}}[X_n-(CY^r_n+W)])\,\big|X_n\Big)\\
=&\mathbb{E}\Big(\mu_n\,\mathrm{tr}\Big(\Sigma_r^{-1}(X_n-W)^{\mathrm{T}}(X_n-W)\Big)\big|X_n\Big)-2\mathbb{E}\Big(\mu_n\,\mathrm{tr}\Big(\Sigma_r^{-1}(X_n-W)^{\mathrm{T}}CY^r_n\Big)\big|X_n\Big)\\
&+\mathbb{E}\Big(\mathrm{tr}\,(\mu_n\Sigma_r^{-1}(Y^r_n)^{\mathrm{T}}(C^{\mathrm{T}}C)Y^r_n)\,\big|X_n\Big).
\end{aligned} \tag{A3}$$

Then, we give the calculation results of the terms on the right hand side of (A3) from below. It is clear that

$$\begin{aligned}
\mathbb{E}\Big(\mu_n\,\mathrm{tr}\Big(\Sigma_r^{-1}(X_n-W)^{\mathrm{T}}(X_n-W)\Big)\big|X_n\Big)&=\mathbb{E}(\mu_n|X_n)\,\mathrm{tr}\Big(\Sigma_r^{-1}(X_n-W)^{\mathrm{T}}(X_n-W)\Big)\\
&=E^{(i)}_{\mu_n}\,\mathrm{tr}\Big(\Sigma_r^{-1}(X_n-W)^{\mathrm{T}}(X_n-W)\Big).
\end{aligned} \tag{A4}$$

By (A1) and taking $A = \Sigma_r^{-1}$ and $B = C^{\mathrm{T}}C$ into (A2), we have

$$\mathbb{E}\Big(\mu_n\,\mathrm{tr}\Big(\Sigma_r^{-1}(X_n-W)^{\mathrm{T}}CY^r_n\Big)\big|X_n\Big)=E^{(i)}_{\mu_n}\,\mathrm{tr}\Big(\Sigma_r^{-1}(X_n-W)^{\mathrm{T}}CE^{(i)}_{Y^r_n}\Big), \tag{A5}$$

$$\begin{aligned}
\mathbb{E}\Big(\mathrm{tr}(\mu_n\Sigma_r^{-1}(Y^r_n)^{\mathrm{T}}(C^{\mathrm{T}}C)Y^r_n)|X_n\Big)=&\,\mathrm{tr}(\Sigma_r^{-1}\Sigma^{(i)}_r)\,\mathrm{tr}\Big(C^{\mathrm{T}}C(\sigma^{(i)}_c)^2(\Phi^{(i)}_c)^{-1}\Big)\\
&+E^{(i)}_{\mu_n}\,\mathrm{tr}\Big(\Sigma_r^{-1}(E^{(i)}_{Y^r_n})^{\mathrm{T}}C^{\mathrm{T}}CE^{(i)}_{Y^r_n}\Big).
\end{aligned} \tag{A6}$$

Equality (22) is a consequence of (A3)–(A6).
Similarly, for (27), it follows that

$$\begin{aligned}
&\mathbb{E}\Big(\mathrm{tr}\,(\mu_n[X_n-(Y^c_nR^{\mathrm{T}}+W)]^{\mathrm{T}}\Sigma_c^{-1}[X_n-(Y^c_nR^{\mathrm{T}}+W)])\,\big|X_n\Big)\\
=&\mathbb{E}\Big(\mu_n\,\mathrm{tr}\Big((X_n-W)^{\mathrm{T}}\Sigma_c^{-1}(X_n-W)\Big)\big|X_n\Big)-2\mathbb{E}\Big(\mu_n\,\mathrm{tr}\Big((X_n-W)^{\mathrm{T}}\Sigma_c^{-1}Y^c_nR^{\mathrm{T}}\Big)\big|X_n\Big)\\
&+\mathbb{E}\Big(\mu_n\,\mathrm{tr}\,(R^{\mathrm{T}}R(Y^c_n)^{\mathrm{T}}\Sigma_c^{-1}Y^c_n)\,\big|X_n\Big)\\
=&\widetilde{E}^{(i)}_{\mu_n}\,\mathrm{tr}\Big((X_n-W)^{\mathrm{T}}\Sigma_c^{-1}(X_n-W)\Big)-2\widetilde{E}^{(i)}_{\mu_n}\,\mathrm{tr}\,\Big((X_n-W)^{\mathrm{T}}\Sigma_c^{-1}E^{(i)}_{Y^c_n}R^{\mathrm{T}}\Big)\\
&+\mathrm{tr}(\Sigma_c^{-1}\Sigma^{(i+1)}_c)\,\mathrm{tr}\Big(R^{\mathrm{T}}R(\sigma^{(i)}_r)^2(\Phi^{(i)}_r)^{-1}\Big)+\widetilde{E}^{(i)}_{\mu_n}\,\mathrm{tr}\Big(R^{\mathrm{T}}R(E^{(i)}_{Y^c_n})^{\mathrm{T}}\Sigma_c^{-1}E^{(i)}_{Y^c_n}\Big),
\end{aligned} \tag{A7}$$

where $\widetilde{E}^{(i)}_{\mu_n}$ and $E^{(i)}_{\Upsilon^c_n}$ are defined in (25). The last equality (A7) holds due to the same reason as (A3).

References

1. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
2. Burges, C.J.C. *Dimension Reduction: A Guided Tour*; Now Publishers Inc.: Noord-Brabant, The Netherlands, 2010.
3. Ma, Y.; Zhu, L. A review on dimension reduction. *Int. Stat. Rev.* **2013**, *81*, 134–150.
4. Jolliffe, I.T. *Principal Component Analysis*; Springer: Berlin/Heidelberg, Germany, 2002.
5. Yang, J.; Zhang, D.; Frangi, A.F.; Yang, J. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 131–137.
6. Ye, J. Generalized low rank approximations of matrices. *Mach. Learn.* **2005**, *61*, 167–191.
7. Zhang, D.; Zhou, Z.H. (2D) 2PCA: Two-directional two-dimensional PCA for efficient face representation and recognition. *Neurocomputing* **2005**, *69*, 224–231.
8. Tipping, M.E.; Bishop, C.M. Probabilistic principal component analysis. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1999**, *61*, 611–622.
9. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; MIT Press: Cambridge, MA, USA, 2012.
10. Yu, S.; Bi, J.; Ye, J. Matrix-variate and higher-order probabilistic projections. *Data Min. Knowl. Discov.* **2011**, *22*, 372–392.
11. Gupta, A.K.; Nagar, D.K. *Matrix Variate Distributions*; CRC Press: Boca Raton, FL, USA, 2018; Volume 104.
12. Zhao, J.; Philip, L.H.; Kwok, J.T. Bilinear probabilistic principal component analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 492–503.
13. Zhao, J.; Jiang, Q. Probabilistic PCA for t distributions. *Neurocomputing* **2006**, *69*, 2217–2226.
14. Chen, T.; Martin, E.; Montague, G. Robust probabilistic PCA with missing data and contribution analysis for outlier detection. *Comput. Stat. Data Anal.* **2009**, *53*, 3706–3716.
15. Gao, J. Robust L1 principal component analysis and its Bayesian variational inference. *Neural Comput.* **2008**, *20*, 555–572. [CrossRef] [PubMed]
16. Ju, F.; Sun, Y.; Gao, J.; Hu, Y.; Yin, B. Image outlier detection and feature extraction via L1-norm-based 2D probabilistic PCA. *IEEE Trans. Image Process.* **2015**, *24*, 4834–4846. [CrossRef] [PubMed]
17. Galimberti, G.; Soffritti, G. A multivariate linear regression analysis using finite mixtures of t distributions. *Comput. Stat. Data Anal.* **2014**, *71*, 138–150. [CrossRef]
18. Morris, K.; McNicholas, P.D.; Scrucca, L. Dimension reduction for model-based clustering via mixtures of multivariate t-distributions. *Adv. Data Anal. Classif.* **2013**, *7*, 321–338. [CrossRef]
19. Pesevski, A.; Franczak, B.C.; McNicholas, P.D. Subspace clustering with the multivariate-t distribution. *Pattern Recognit. Lett.* **2018**, *112*, 297–302. [CrossRef]
20. Teklehaymanot, F.K.; Muma, M.; Zoubir, A.M. Robust Bayesian cluster enumeration based on the t distribution. *Signal Process.* **2021**, *182*, 107870. [CrossRef]
21. Wei, X.; Yang, Z. The infinite Student's t-factor mixture analyzer for robust clustering and classification. *Pattern Recognit.* **2012**, *45*, 4346–4357. [CrossRef]
22. Zhou, X.; Tan, C. Maximum likelihood estimation of Tobit factor analysis for multivariate t-distribution. *Commun. Stat. Simul. Comput.* **2009**, *39*, 1–16. [CrossRef]
23. Lange, K.L.; Little, R.J.A.; Taylor, J.M.G. Robust statistical modeling using the t distribution. *J. Am. Stat. Assoc.* **1989**, *84*, 881–896. [CrossRef]
24. Meng, X.L.; Van Dyk, D. The EM algorithm—An old folk-song sung to a fast new tune. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1997**, *59*, 511–567. [CrossRef]
25. Zhou, Y.; Lu, H.; Cheung, Y. Bilinear probabilistic canonical correlation analysis via hybrid concatenations. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 5–9 October 2017; Volume 31.
26. Coleman, T.; Branch, M.A.; Grace, A. Optimization toolbox. In *For Use with MATLAB. User's Guide for MATLAB 5, Version 2, Relaese II*; 1999. Available online: https://www.dpipe.tsukuba.ac.jp/~naito/optim_tb.pdf (accessed on 20 September 2021).
27. Schrage, L. *LINGO User's Guide*; LINDO System Inc.: Chicago, IL, USA, 2006.
28. Stewart, G.W. *Matrix Algorithms: Volume II: Eigensystems*; SIAM: Philadelphia, PA, USA, 2001.