Circuits, Systems, and Signal Processing (2020) 39:4110–4135
https://doi.org/10.1007/s00034-020-01358-1

Bayesian Robust Principal Component Analysis with Adaptive Singular Value Penalty

Kaiyan Cui[1,2] · **Guan Wang**[1] · **Zhanjie Song**[1,3] · **Ningning Han**[1]

Received: 8 May 2019 / Revised: 23 January 2020 / Accepted: 25 January 2020 /
Published online: 6 February 2020

Abstract

Robust principal component analysis (RPCA) has recently seen ubiquitous activity for dimensionality reduction in image processing, visualization and pattern recognition. Conventional RPCA methods model the low-rank component as regularizing each singular value equally. However, in numerous modern applications, each singular value has different physical meaning and should be treated differently. This is one of the main reasons why RPCA techniques cannot work well in dealing with many realistic problems. To solve this problem, a novel hierarchical Bayesian RPCA model with adaptive singular value penalty is proposed. This model enforces the low-rank constraint by introducing an adaptive penalty function on the singular values of the low-rank component. In particular, we impose a hierarchical Exponent-Gamma prior on the singular values of the low-rank component and the Beta-Bernoulli prior on sparsity indicators. The variational Bayesian framework and the Markov chain Monte Carlo-based Bayesian inference are considered for inferring the posteriors of all latent variables involved in low-rank and sparse components. Numerical experiments demonstrate the competitive performance of the proposed model on synthetic and real data.

Keywords Robust principal component analysis · Bayesian modeling · Adaptive singular value penalty · Variational Bayesian inference

✉ Guan Wang
crownwang@tju.edu.cn

Kaiyan Cui
cuikaiyan@tju.edu.cn

Zhanjie Song
zhanjiesong@tju.edu.cn

Ningning Han
ning_ninghan@tju.edu.cn

1 School of Mathematics, Tianjin University, Tianjin 300350, China

2 Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA

3 Visual Pattern Analysis Research Lab, Tianjin University, Tianjin 300072, China

1 Introduction

Robust principal component analysis (RPCA) has been a common tool instead of principal component analysis (PCA) [12,16,23,30,34,35,40] for dimensionality reduction in numerous modern applications such as face modeling [21,39], video surveillance [18], subspace clustering [25]. This is motivated by the theoretical advances for extracting low-dimensional information from the observed high-dimensional data matrix and addressing the fragility of PCA with respect to outlier sparse noise [1,5,11]. Essentially, RPCA incorporates low-rank and sparse constraints on the observed matrix and searches the best low-rank approximation of the observed data.

A typical RPCA assumes that the observed matrix $Y \in \mathbb{R}^{m\times n}$ can be decomposed into a low-rank component $L \in \mathbb{R}^{m\times n}$ (with the rank $r << \min\{m, n\}$) and a sparse component $S \in \mathbb{R}^{m\times n}$ (corresponding to the sparse outliers). The objective is to recover the low-rank component L and the sparse component S. A standard approach is to find L and S from Y by solving

$$\min_{L,S} \operatorname{rank}(L) + \nu\|S\|_0, \quad \text{s.t. } Y = L + S, \tag{1}$$

where ν is a tuning parameter and $\|S\|_0$ is the number of nonzero entries in S. Unfortunately, (1) is a non-deterministic polynomial (NP)-hard problem. To overcome this problem, several methods based on the nuclear norm and ℓ_1-norm minimization have been proposed, which relaxed the rank function into the nuclear norm and replaced the ℓ_0-norm with the ℓ_1-norm. Subsequently, one has the following convex optimization problem

$$\min_{L,S} \|L\|_* + \nu\|S\|_1, \quad \text{s.t. } Y = L + S, \tag{2}$$

where $\|L\|_*$ is the nuclear norm (the sum of singular values) of L, and $\|S\|_1$ is the ℓ_1-norm (the sum of absolute values of the entries) of S. Candès et al. [5] showed that L and S can be exactly recovered from Y with high probability under broad condition by solving the optimization problem (2).

Many effective methods [1,4,11,13–15,24,29,38] have been presented to solve the optimization problem (2). On the one hand, heuristic deterministic approaches including singular value thresholding (SVT) [4], accelerated proximal gradient algorithm (APG) [38] and augmented Lagrange multiplier (ALM) [24] have been developed. On the other hand, there has been a significant interest in Bayesian approaches. For example, Gao [13] and Luttinen et al. [29] proposed the probabilistic RPCA models by introducing the heavy-tail random noise component. Ding et al. [11] modeled the entries of sparse component S and the singular values of low-rank component L with Beta-Bernoulli prior and proposed using Markov chain Monte Carlo (MCMC) method to approximate inference. Babacan et al. [1] presented the sparse Bayesian learning (SBL) principles for solving probabilistic RPCA models, which started from a matrix factorization formulation and enforced the low-rank constraint in estimating the sparse component. In real scenarios, noises commonly appear in the measurements. The deterministic models cannot deal with the noises well since the convergence of the

solution and its stability are influenced by the noises [36,37]. In contrary, the probabilistic models can deal with noises well. For example, in [7,43], complex noises were added in probabilistic RPCA models to improve their recovery performances in real scenarios. By exploiting structural dependencies between the values and locations of the sparse component, Han et al. [19] proposed Bayesian RPCA with the structured sparse component. Overall, these probabilistic RPCA models offer three main advantages over the deterministic methods. Firstly, it is not necessary to know the prior knowledge about the rank of the low-rank component and the way to estimate the unknown rank is similar to the automatic relevance determination strategy. Secondly, algorithmic parameters are insensitive to the initialization of parameters since they are treated as stochastic quantities in the Bayesian framework. Thirdly, the probabilistic RPCA models allow us to exploit and explain complex noise structure in observation matrix Y, which are robust to a broad of noises and provide high recovery performance.

However, the present probabilistic RPCA models obviously impose the low-rank constraint on each singular value of the low-rank component L equally and ignore the physical meaning, i.e., the prior knowledge, on these singular values [1,9–11,19,26,27,42]. In numerous modern applications, the weight distributions of each singular value of the low-rank component L are not truly uniformly distributions, and should be treated differently so that the information contained in the small singular values covered by that of the big singular values can be recovered well. There have been nonconvex low-rank constraints used in some deterministic RPCA models. For example, Hu et al. [22] proposed the truncated nuclear norm regularization (TNNR) by minimizing the sum of the largest few singular values; Lu et al. [28] proposed to minimize the rank using a family of nonconvex surrogates of ℓ_0-norm on the singular values. By now, however, these works do not extend to the probabilistic RPCA models.

Motivated by the trajectory of deterministic methods in addressing the aforementioned problem, in this paper, we present a novel hierarchical Bayesian RPCA model with adaptive singular value penalty for decomposing the observed matrix into low-rank and sparse components. To boost the performance of recovering low-rank component of RPCA model, we enforce the low-rank constraint by introducing an adaptive penalty function on the singular values of the low-rank component. In particular, we impose a hierarchical Exponent-Gamma prior on the singular values of the low-rank component and the Beta-Bernoulli prior on sparsity indicators. The exponential prior for low-rank matrix recovery has been widely used and researched, and the Gamma prior penalizes the singular values (see Sect. 2 for details); thus, an Exponent-Gamma framework is employed on the singular values of the low-rank component, which provides more flexibility-weighted structures for the learning of the low-rank component. The zero elements in singular values of low-rank component are further guaranteed to be exactly zero by introducing the binary latent factor indicators [19]. The variational Bayesian framework and the Markov chain Monte Carlo (MCMC)-based Bayesian inference are considered for inferring the posteriors of all latent variables involved in low-rank and sparse components. Besides, different initial values of the input parameters may affect the outcome. The optimal parameters should be selected carefully [32]. We discussed the possible initial conditions that provided satisfactory results.

The rest of this paper is organized as follows. Section 2 presents the motivations. The proposed Bayesian robust principal component analysis model with the adaptive penalty on the singular values (APSV-BRPCA) is proposed in Sect. 3. A variational Bayesian inference algorithm is employed to estimate the posterior distributions in Sect. 4. We present an analysis of the APSV-BRPCA approach in Sect. 5. Experiments including synthetic and real data are shown in Sect. 6. Finally, Sect. 7 gives the conclusion.

2 Motivations

The Bayesian model presented in this paper is closely related to some deterministic models. In this section, we provide insights into understanding the low-rank Bayesian model with the adaptive penalty on singular values.

In sparse Bayesian learning, the penalty function (i.e., the log-likelihood of the singular value vector λ) is assumed to be an exponential prior

$$f(\lambda) = -\ln p(\lambda) = \text{constant} + \gamma \|X\|_*, \tag{3}$$

where $p(\lambda) = \prod_{i=1}^{r} \gamma \exp(-\gamma \lambda_i)$, r is the rank of X and $f(\lambda)$ is penalty function. This penalty function is equal to the nuclear norm of matrix x

$$\min_{x \in \mathbb{R}^n} \|x\|_*, \quad \text{s.t.} \quad y = \Psi x, \tag{4}$$

where $\|x\|_* = |\lambda|_1 = \sum_{i=1}^{r} \lambda_i$, $\lambda = [\lambda_1, \ldots, \lambda_r]$ is the singular value vector of x.

To improve the ℓ_1-minimization problem for compressive sensing, the reweighted ℓ_1 minimization [6] is given by

$$\min_{x \in \mathbb{R}^n} \sum_i w_i |x_i|, \quad \text{s.t.} \quad y = \Psi x, \tag{5}$$

where w_i is positive weight.

Meanwhile, the prior model (3) corresponding to the traditional nuclear norm can be further elaborated by introducing the hierarchical Exponent-Gamma prior

$$\begin{aligned} p(\lambda|\gamma) &= \prod_{i=1}^{r} \gamma_i \exp(-\gamma_i \lambda_i), \\ p(\gamma_i) &= \text{Gamma}(\gamma_i|a, b), \end{aligned} \tag{6}$$

where λ_i represents the ith singular value and $\gamma = [\gamma_1, \ldots, \gamma_r]$ is a weighted parameter vector of the singular values. Here, the penalty function as calculated in model (3) is given by

$$f(\lambda) = -\ln p(\lambda|\gamma) = \text{constant} + \sum_{i=1}^{r} \gamma_i \lambda_i. \tag{7}$$

Consequently, model (7) is actually related to the reweighted ℓ_1-norm (5),

which means the new hierarchical prior (6) adaptively imposes weights on different singular values so that it potentially improves the flexibility of sparse Bayesian learning.

3 APSV-BRPCA Model

The Bayesian model treats its parameters as stochastic quantities and uses the hierarchical Bayesian framework for the factorized form of measurements to connect stochastic parameters. Generally, the observed data matrix $Y \in \mathbb{R}^{m\times n}$ is assumed to be the superposition of three parts: the low-rank component $L \in \mathbb{R}^{m\times n}$, the sparse component $S \in \mathbb{R}^{m\times n}$ and the noise term $N \in \mathbb{R}^{m\times n}$, i.e., $Y = L + S + N$. Utilizing singular value decomposition, any matrix L of rank r can be decomposed as $L = U\Lambda V^T$, where U and V are $m \times r$ and $n \times r$ matrices, respectively, and diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_r) \in \mathbb{R}^{r\times r}$ is consisted of nonzero singular values of L. Then the proposed factorized form of measurements for the Bayesian frame is as follows:

$$Y = U(D\Lambda)V^T + B \circ E + N, \tag{8}$$

where $\circ$ denotes the Hadamard product. The diagonal matrix D and the sparse indicator matrix B are binary matrices which are used for enforcing sparsity.

3.1 Hierarchical Low-Rank Model

The low-rank component proposed in (8) is given by $L = U(D\Lambda)V^T$. Our main goal is to add the adaptive penalty on the singular values in Bayesian RPCA model. To track this issue, as mentioned in Sect. 2, the hierarchical prior of Λ is drawn from an Exponent-Gamma distribution

$$\begin{aligned} \lambda_k &\sim \mathrm{Exp}(\lambda_k|\gamma_k), \\ \gamma_k &\sim \mathrm{Gamma}(a_0, b_0), \end{aligned} \quad k = 1, \ldots, r,$$

where $\mathrm{Exp}(\lambda_k|\gamma_k) = \gamma_k \exp(-\gamma_k\lambda_k)$.

The diagonal matrix $D = \mathrm{diag}(d_1, \ldots, d_r)$ has binary entries along the diagonal, i.e., $d_k \in \{0, 1\}, k = 1, \ldots, r$. Then $D\Lambda$ is still a diagonal matrix. In the Bayesian model, diagonal matrix D decouples the rank learning and the singular value learning [11]. We can infer the magnitudes of the singular values using Λ and the rank of L by $r = ||D||_0$. An appropriate choice for modeling D is based on a product of Bernoulli-Beta prior distribution

$$\begin{aligned} d_k &\sim \mathrm{Bernoulli}(d_k|\pi_k), \\ \pi_k &\sim \mathrm{Beta}(\pi_k|\theta_0, \eta_0), \end{aligned} \quad k = 1, \ldots, r. \tag{9}$$

Note that the sparse of diagonal elements of D is controlled by the hyperparameters $\theta_0 > 0$ and $\eta_0 > 0$. Specifically, the expectation of π_k is $\frac{\theta_0}{\theta_0+\eta_0}$, thus one can set $\frac{\theta_0}{\theta_0+\eta_0} \ll 1$ (e.g., $\theta_0 = \min\{\frac{1}{r}, \frac{1}{150}\}$, $\eta_0 = 1 - \theta_0$) to encourage the sparse of diagonal elements of D.

The columns of matrices $U = [\mathbf{u}_{\cdot i}]$ and rows of $V = [\mathbf{v}_{j\cdot}]$ are assumed to be drawn from Gaussian distributions

$$\begin{aligned} u_{\cdot i} &\sim \mathcal{N}(u_{\cdot i}|0, \frac{1}{m} I_m), \quad i = 1, \ldots, r, \\ v_{j\cdot} &\sim \mathcal{N}(v_{j\cdot}|0, I_r), \quad j = 1, \ldots, n. \end{aligned} \tag{10}$$

3.2 Sparse Model

The sparse component proposed in (8) is given by $S = B \circ E$, where B is a binary matrix. Again, the binary matrix is used for enhancing the recovery of the sparse component, such that the Bayesian model can keep the sparse nature of E. Each column of the binary matrix $B = [\mathbf{b}_{\cdot j}]$ is modeled as follows:

$$\begin{aligned} \mathbf{b}_{\cdot j} &\sim \prod_{i=1}^{m} \text{Bernoilli}(b_{ij}|\omega_i), \quad j = 1, \ldots, n, \\ \omega_i &\sim \text{Beta}(\omega_i|\alpha_0, \beta_0), \quad i = 1, \ldots, m, \end{aligned} \tag{11}$$

where $\alpha_0 = \min\{\frac{1}{n}, \frac{1}{150}\}$, $\beta_0 = 1 - \alpha_0$.

The element e_{ij} in E follows a Gaussian-Gamma distribution

$$\begin{aligned} \mathbf{e}_{i\cdot} &\sim \mathcal{N}(\mathbf{e}_{i\cdot}|\mathbf{0}, \boldsymbol{\phi}^{-1} I_n), \quad i = 1, \ldots, m, \\ \boldsymbol{\phi} &\sim \text{Gamma}(\tau|c_0, d_0). \end{aligned} \tag{12}$$

If necessary, one can learn different noise precisions for different parts of E. But it will result in overfitting because of limited observations and too many parameters. To avoid the overfitting, we use one precision τ to model the element e_{ij}.

3.3 Noisy Observation Model

One advantage of probability RPCA is the robustness for the additive interference. The elements n_{ij} in N are assumed drawn from a Gaussian distribution as follows:

$$\begin{aligned} n_{ij} &\sim \mathcal{N}(n_{ij}|0, \beta^{-1}), \quad i = 1, \ldots, m, \quad j = 1, \ldots, n, \\ \beta &\sim \text{Gamma}(\beta|e_0, f_0). \end{aligned} \tag{13}$$

The noise precision is assumed unknown and learnt within the model inference. As discussed above, to avoid the overfitting and improve the adaptivity, the same precision

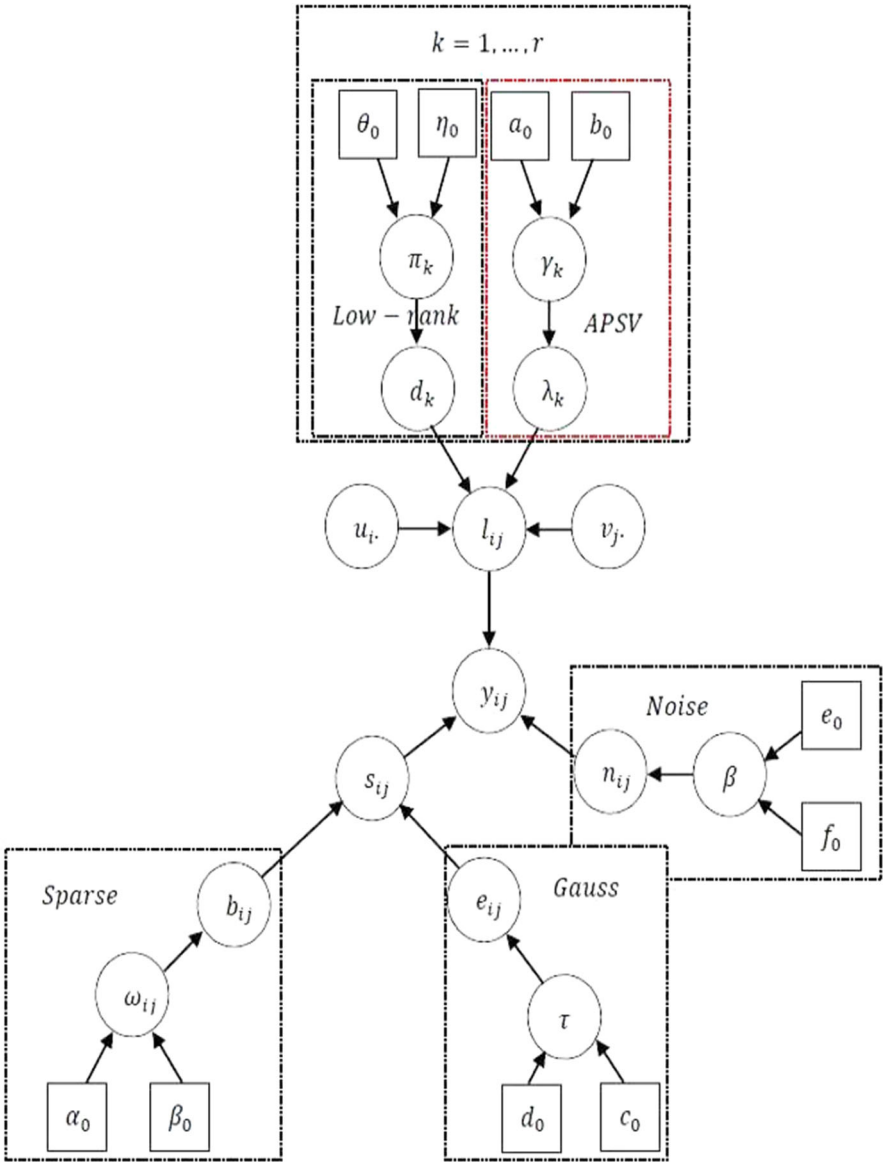

Fig. 1 APSV-BRPCA model

parameter β is imposed on the noise n_{ij}. The common choice in sparse Bayesian learning such as [8,19,41] has shown that it can reduce model complexity and increase robustness.

Given the prior defined as above mentioned, the conditional distribution for the observation model is as follows:

$$p(Y|U, D, \Lambda, V, B, E) = \mathcal{N}(Y|U(D\Lambda)V^T + B \circ E, \beta^{-1} I_{mn}). \tag{14}$$

To facilitate the illustration of the proposed Bayesian model, Fig. 1 shows the complete graphical model.

4 Variational Bayesian Inference

Bayesian inference is evaluating the posterior distributions of unknowns given the observation. However, the posterior distribution is computationally intractable since the marginal distribution $p(Y)$ is not calculated analytically. In this paper, to deal with the tractable joint posterior distribution problem, the approximate posterior distribution factorized with respect to the partition $q(\Theta) = \prod_k q_k(\Theta_k) = q_k$ is considered, and the variational Bayesian method [3] is used to obtain the Bayesian inference of $q(\Theta)$. In particular, the variational Bayesian method estimates the posterior distribution of each unknown parameter by holding the other parameters fixed [3].

The objective of the variational Bayesian method is to select the parameters to minimize the Kullback–Leibler (KL) divergence between $q(\Theta)$ and the true posterior distribution $p(\Theta|Y)$, i.e.,

$$\min_{q(\Theta)} KL(q(\Theta)\|p(\Theta)) = \int q(\Theta) \ln \frac{q(\Theta)}{p(\Theta|Y)} \mathrm{d}\Theta,$$

where Θ=$\{U, D, \Lambda, V, \pi, \lambda, \gamma, B, E, \omega, \tau, \beta\}$ denotes the set contained all parameter variables. It is equivalent to the following problem

$$\max_{q(\Theta)} -KL(q(\Theta)\|p(\Theta)) = \int q(\Theta) \ln \frac{p(\Theta, Y)}{q(\Theta)p(Y)} \mathrm{d}\Theta.$$

Since $KL(q(\Theta)\|p(\Theta)) \geq 0$ and $\int q(\Theta)\mathrm{d}\Theta = 1$, the key procedure is to estimate the marginal likelihood $p(Y)$ with a maximal lower bound, i.e.,

$$\ln p(Y) \geq \mathcal{L}(\theta) = \int q(\Theta) \ln \frac{p(\Theta, Y)}{q(\Theta)} \mathrm{d}\Theta.$$

The problem is developed to maximize the lower bound $\mathscr{L}(\theta)$.

$$\begin{aligned}
\mathscr{L}(\theta) &= \int \prod_k q_k[\ln p(\Theta, Y) - \sum_k \ln q_k]\mathrm{d}\Theta \\
&= \int \prod_k q_k \ln p(\Theta, Y) \prod_k \mathrm{d}\Theta_k - \sum_k \int \prod_j q_j \ln q_k \prod_j \mathrm{d}\Theta_j \\
&= \int q_k[\ln p(\Theta, Y) \prod_{j\neq k}(q_j \mathrm{d}\Theta_j)]\mathrm{d}\Theta_k - \sum_k \int q_k \ln q_k \mathrm{d}\Theta_k \\
&= \int q_k[\ln p(\Theta, Y) \prod_{j\neq k}(q_j \mathrm{d}\Theta_j)]\mathrm{d}\Theta_k - \int q_k \ln q_k \mathrm{d}\Theta_k - \sum_{j\neq k} \int q_j \ln q_j \mathrm{d}\Theta_j \\
&= \int q_k \ln \overline{p}(\Theta_k, Y)\mathrm{d}\Theta_k - \int q_k \ln q_k \mathrm{d}\Theta_k - \sum_{j\neq k} \int q_j \ln q_j \mathrm{d}\Theta_j \\
&= -KL(q_k \| \overline{p}) - \sum_{j\neq k} \int q_j \ln q_j \mathrm{d}\Theta_j,
\end{aligned} \tag{15}$$

where $\ln \overline{p}(\Theta_k, Y) = E_{\Theta\setminus\Theta_k}[\ln p(\Theta, Y)] = \int \ln p(\Theta, Y) \prod_{j\neq k}(q_j \mathrm{d}\Theta_j)$ and $\int q_j \mathrm{d}\Theta_j = 1$ $(j = 1, \cdots)$. The expectation $E_{\Theta\setminus\Theta_k}$ is taken about the set Θ with Θ_k removed. Clearly, the bound in (15) is maximized when $q_k(\Theta_k) = \overline{p}(\Theta_k, y)$. In this case, the KL divergence is equal to zero. Consequently, the optimal posterior estimation $q_k(\Theta_k)$ with other variables fixed is as follows:

$$\ln q_k(\Theta_k) = E_{\Theta\setminus\Theta_k}[\ln p(Y, \Theta)] + \mathrm{C}, \tag{16}$$

where C is a constant. In the next subsection, we calculate each parameter in its turn holding other parameters fixed with respect to their most recent distributions and separately show the update rules for each parameter. For notational simplicity, the expectation of the approximate posterior $q(\cdot)$ is denoted by $\langle\cdot\rangle$.

4.1 Estimation of Low-Rank Component

The parameters involved in the low-rank component are U, D, Λ, V, d, π, λ, γ. Invoking the prior model (10), the observed model (14), one can obtain the posterior distribution of the row $u_{i\cdot}$ of U,

$$q(u_{i\cdot}) = \mathscr{N}(u_{i\cdot} \mid \mu_{u_{i\cdot}}, \Sigma^U), \tag{17}$$

with mean and covariance

$$\begin{aligned}
\Sigma^U &= (\langle\beta\rangle\langle D\Lambda V^T V \Lambda^T D^T\rangle + mI_r)^{-1}, \\
\mu_{u_{i\cdot}}^T &= \langle\beta\rangle \Sigma^U \langle V\Lambda^T D^T\rangle^T (y_{i\cdot} - \langle b_{i\cdot}\rangle \circ \langle e_{i\cdot}\rangle)^T.
\end{aligned}$$

Similarly, for each row of V (i.e., $v_{j.}$), we get

$$q(v_{j.}) = \mathcal{N}(v_{j.} \mid \mu_{v_{i.}}, \Sigma^V), \tag{18}$$

where

$$\begin{aligned}
\Sigma^V &= (\langle\beta\rangle\langle D^T \Lambda^T U^T U D \Lambda\rangle + I_r)^{-1}, \\
\mu_{v_{j.}}^T &= \langle\beta\rangle \Sigma^V \langle U \Lambda D\rangle^T (y_{.j} - \langle b_{.j}\rangle \circ \langle e_{.j}\rangle).
\end{aligned}$$

The required expectations in (17) and (18) can be calculated as follows:

$$\begin{aligned}
&\langle D \Lambda V^T V \Lambda^T D^T\rangle = \langle d^T d\rangle \circ \langle \lambda^T \lambda\rangle \circ \langle V^T V\rangle, \\
&\langle D^T \Lambda^T U^T U D \Lambda\rangle = \langle d^T d\rangle \circ \langle \lambda^T \lambda\rangle \circ \langle U^T U\rangle, \\
&\langle U^T U\rangle = m \Sigma^U + \langle U^T\rangle\langle U\rangle, \\
&\langle V^T V\rangle = n \Sigma^V + \langle V^T\rangle\langle V\rangle, \\
&\langle d^T d\rangle = \Sigma_d + \langle d^T\rangle\langle d\rangle, \\
&\langle \lambda^T \lambda\rangle = \Sigma_\lambda + \langle \lambda^T\rangle\langle \lambda\rangle,
\end{aligned}$$

where d =diag(D) and λ= diag(Λ) are row vectors, Σ_d= diag(Var(d_k)) and Σ_λ =diag(Var(λ_k)) are diagonal matrices.

The posterior approximation of D is a Bernoulli distribution

$$q(d_k) = \text{Bernoulli}\left(\frac{\xi^{d_k}}{\xi^{d_k} + \zeta^{d_k}}\right), \tag{19}$$

where

$$\begin{aligned}
\xi^{d_k} &= \exp\{\langle \ln \pi_k\rangle - \frac{\langle\beta\rangle}{2} \sum_{ij} [\langle u_{ik}^2 v_{jk}^2 \lambda_k^2\rangle - 2(y_{ij} - \langle b_{ij}\rangle\langle e_{ij}\rangle)\langle \lambda_k u_{ik} v_{jk}\rangle \\
&\quad + 2\langle u_{i.}^{-k} D^{-k} \Lambda^{-k} (v_{j.}^{-k})^T \lambda_k u_{ik} v_{jk}\rangle]\}, \\
\zeta^{d_k} &= \exp\{\langle \ln(1 - \pi_k)\rangle\},
\end{aligned}$$

a^{-k} means that the kth coordinate of a is deleted,

$$\begin{aligned}
&\langle d_k\rangle = \frac{\xi^{d_k}}{\xi^{d_k} + \zeta^{d_k}}, \\
&\langle u_{ik}^2 v_{jk}^2 \lambda_k^2\rangle = \langle u_{ik}^2\rangle\langle v_{jk}^2\rangle\langle \lambda_k^2\rangle, \\
&\langle u_{i.}^{-k} D^{-k} \Lambda^{-k} (v_{j.}^{-k})^T \lambda_k u_{ik} v_{jk}\rangle = \{\langle u_{i.}^T u_{i.}\rangle \circ \langle \lambda^T \lambda\rangle \circ \langle v_{j.}^T v_{j.}\rangle\}_{.k}^T \langle d^k\rangle, \\
&\langle d^k\rangle = (\langle d_1\rangle, \ldots, 0, \ldots, \langle d_r\rangle), \\
&\langle u_{i.}^T u_{i.}\rangle = \Sigma^U + \langle u_{i.}^T\rangle\langle u_{i.}\rangle, \\
&\langle v_{j.}^T v_{j.}\rangle = \Sigma^V + \langle v_{j.}^T\rangle\langle v_{j.}\rangle.
\end{aligned}$$

According to the prior model (9), one can find $q(\pi_k)$ follows a Beta distribution,

$$q(\pi_k) = \text{Beta}(\pi_k \mid \langle d_k \rangle + \theta_0, 1 - \langle d_k \rangle + \eta_0). \tag{20}$$

Then we have

$$\begin{aligned}\langle \ln \pi_k \rangle &= \psi(\langle d_k \rangle + \theta_0) - \psi(\theta_0 + \eta_0 + 1),\\ \langle \ln(1 - \pi_k) \rangle &= \psi(1 - \langle d_k \rangle + \eta_0) - \psi(\theta_0 + \eta_0 + 1),\end{aligned}$$

where $\psi(x) = \frac{\mathrm{d} \ln \Gamma(x)}{\mathrm{d}x}$ is a digamma function.

The variational posterior of λ_k is a truncated Gaussian distribution

$$q(\lambda_k) = \frac{\mathcal{N}(\lambda_k \mid \mu_{\lambda_k}, \sigma^2_{\lambda_k})}{1 - \Phi(\epsilon)} I_{\{\lambda_k > 0\}}, \epsilon = -\frac{\mu_{\lambda_k}}{\sigma_{\lambda_k}}, \tag{21}$$

where

$$\begin{aligned}\sigma^2_{\lambda_k} &= (\langle \beta \rangle \sum_{ij} \langle u^2_{ik} v^2_{jk} d^2_k \rangle)^{-1},\\ \mu_{\lambda_k} &= \sigma^2_{\lambda_k} \Bigg\{ \sum_{ij} [(y_{ij} - \langle b_{ij} \rangle \langle e_{ij} \rangle) \langle u_{ik} d_k v_{jk} \rangle \\ &\qquad - \langle \lambda^{-k} U_i^{-k} D^{-k} (v_{j\cdot}^{-k})^T u_{ik} d_k v_{jk} \rangle] \langle \beta \rangle - \langle \gamma_k \rangle \Bigg\},\end{aligned}$$

$U_i = \text{diag}(u_{i\cdot})$ is a diagonal matrix, $\phi(x) = N(x \mid 0, 1)$, $\Phi(x)$ is a standard Gaussian distribution function. Then we can get

$$\begin{aligned}\langle \lambda_k \rangle &= \mu_{\lambda_k} + \sigma_{\lambda_k} \frac{\phi(\epsilon)}{1 - \Phi(\epsilon)},\\ \text{Var}(\lambda_k) &= \sigma^2_{\lambda_k} [1 + \epsilon \frac{\phi(\epsilon)}{1 - \Phi(\epsilon)} - (\frac{\phi(\epsilon)}{1 - \Phi(\epsilon)})^2],\\ \langle \lambda^{-k} U_i^{-k} D^{-k} (v_{j\cdot}^{-k})^T d_k u_{ik} v_{jk} \rangle &= \{\langle u_{i\cdot}^T u_{i\cdot} \rangle \circ \langle d^T d \rangle \circ \langle v_{j\cdot}^T v_{j\cdot} \rangle\}_{\cdot k}^T \langle \lambda^k \rangle,\\ \langle \lambda^k \rangle &= (\langle \lambda_1 \rangle, \ldots, 0, \ldots, \langle \lambda_r \rangle).\end{aligned}$$

Similarly, the posterior approximation of γ can be derived as follows:

$$q(\gamma_k) = \text{Gamma}(\gamma_k \mid a_0 + 1, b_0 + \langle \lambda_k \rangle). \tag{22}$$

Then we have

$$\langle \gamma_k \rangle = \frac{a_0 + 1}{b_0 + \langle \lambda_k \rangle}.$$

4.2 Estimation of Sparse Component

The parameters involved in the sparse component are E, B, ω, τ. With the Bernoulli (11) and the Gaussian observation likelihood (14), b_{ij} follows a Bernoulli distribution,

$$q(b_{ij}) = \text{Bernoulli}\left(\frac{\xi^{b_{ij}}}{\xi^{b_{ij}} + \zeta^{b_{ij}}}\right), \tag{23}$$

where

$$\begin{aligned}\xi^{b_{ij}} &= \exp\{\langle \ln \omega_i \rangle - \frac{\langle \beta \rangle}{2}(\langle e_{ij}^2 \rangle - 2\langle e_{ij} \rangle (y_{ij} - \langle u_{i\cdot} D \Lambda v_{j\cdot}^T \rangle))\},\\ \zeta^{b_{ij}} &= \exp\{\langle \ln(1-\omega_i) \rangle\}.\end{aligned}$$

Similarly, the approximate posterior of each column $e_{\cdot j}$ is as follows:

$$q(e_{\cdot j}) = \mathcal{N}(e_{\cdot j} \mid \mu_{e_{\cdot j}}, \Sigma_{e_{\cdot j}}), \tag{24}$$

where

$$\begin{aligned}\Sigma_{e_{\cdot j}} &= (\langle \tau \rangle I_m + \langle \beta \rangle \langle B_j \rangle)^{-1},\\ \mu_{e_{\cdot j}} &= \langle \beta \rangle \Sigma_{e_{\cdot j}} \langle B_j \rangle^T (y_{\cdot j} - \langle U D \Lambda \mathbf{v}_{\cdot j}^T \rangle),\end{aligned}$$

where $B_i = \text{diag}(\langle b_{\cdot j} \rangle)$ is a diagonal matrix.

The posterior approximation of ω_i is a Beta distribution

$$\begin{aligned}q(\omega_i) &= \text{Beta}(\omega_i \mid \alpha_0 + \sum_{j=1}^{n} \langle b_{ij} \rangle, \beta_0 + n - \sum_{j=1}^{n} \langle b_{ij} \rangle), i = 1, 2, \ldots, m,\\ \langle \ln \omega_i \rangle &= \psi(\alpha_0 + \sum_{j=1}^{n} \langle b_{ij} \rangle) - \psi(\alpha_0 + \beta_0 + n),\\ \langle \ln(1-\omega_i) \rangle &= \psi(\beta_0 + n - \sum_{j=1}^{n} \langle b_{ij} \rangle) - \psi(\alpha_0 + \beta_0 + n).\end{aligned} \tag{25}$$

The variational posterior of τ is as follows:

$$\begin{aligned}q(\tau) &= \text{Gamma}(\tau \mid c_0 + 0.5mn, d_0 + 0.5 \sum_{j=1}^{n} \langle e_{\cdot j}^T e_{\cdot j} \rangle),\\ \langle \tau \rangle &= \frac{c_0 + 0.5mn}{d_0 + 0.5 \sum_{j=1}^{n} \langle e_{\cdot j}^T e_{\cdot j} \rangle}, \langle e_{\cdot j}^T e_{\cdot j} \rangle = tr(\Sigma_{e_{\cdot j}}) + \langle e_{\cdot j} \rangle^T \langle e_{\cdot j} \rangle.\end{aligned} \tag{26}$$

4.3 Estimation of Noise Precision

Finally, the variational distribution of β is a Gamma distribution,

$$q(\beta) = \text{Gamma}(\beta \mid e_0 + 0.5mn,\ f_0 + 0.5\langle \|Y - UD\Lambda V^T - B \circ E\|_F^2 \rangle),$$
$$\langle \beta \rangle = \frac{e_0 + 0.5mn}{f_0 + 0.5\langle \|Y - UD\Lambda V^T - B \circ E\|_F^2 \rangle}, \tag{27}$$

where

$$\begin{aligned}\langle \|Y - UD\Lambda V^T - B \circ E\|_F^2 \rangle =& \|Y - \langle U\rangle\langle D\rangle\langle \Lambda\rangle\langle V^T\rangle - \langle B\rangle \circ \langle E\rangle\|_F^2 \\ &+ tr(\langle U^T U\rangle \langle D\Lambda V^T V \Lambda^T D^T\rangle) \\ &- tr(\langle U^T\rangle\langle U\rangle\langle D\Lambda V^T\rangle\langle V\Lambda^T D^T\rangle) \\ &+ |\langle B \circ B\rangle \circ \langle E \circ E\rangle|_1 \\ &- tr((\langle B\rangle \circ \langle E\rangle)^T(\langle B\rangle \circ \langle E\rangle)),\end{aligned}$$

$\langle B \circ B\rangle = \Sigma^B + \langle B\rangle \circ \langle B\rangle$, $\Sigma^B = (\text{Var}(b_{ij}))$, $\langle E \circ E\rangle = \Sigma^E + \langle E\rangle \circ \langle E\rangle$, and $\Sigma^E = (\text{diag}(\Sigma_{e_{\cdot j}}))$.

In summary, the VB procedure approximates the posterior distributions of the unknowns iteratively and the whole algorithm is outlined in Algorithm 1.

Algorithm 1 VB for APSV-BRPCA

Input: The measurement Y, parameters a_0, b_0, c_0, d_0, e_0, f_0, θ_0, η_0, α_0, β_0, initial matrices U, D, Λ, V, B, E
Output: The low-rank component L and the sparse component S

while not converged **do**.
Update U using (17).
Update V using (18).
Update D using (19).
Update Λ using (21).
Update B using (23).
Update E using (24).
Update N using (27).
end while.
Set $L = U(D\Lambda)V^T$, $S = B \circ E$.

5 Discussion and MCMC-Based Bayesian Inference

5.1 The Connection Between APSV-BRPCA and Deterministic Approach

There exists a connection between the proposed APSV-BRPCA and WNNM-RPCA [6,11,17]. To see the connection clearly, we show the negative logarithm of the full

posterior density function of the proposed APSV-BRPCA model as follows:

$$
\begin{aligned}
&-\log p(\Theta|Y,\mathscr{H}) \\
&=\| \Lambda \|_{\omega,*} - \log[g_{BB}(D;\mathscr{H})] + \frac{m}{2}\sum_{j=1}^{R}\|\mathbf{u}_{\cdot,j}\|_2^2 + \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{v}_{i,\cdot}\|_2^2 + \frac{\tau}{2}\|E\|_F^2 \\
&\quad - \log[g_{BB}(B;\mathscr{H})] + \frac{\beta}{2}\|Y-L-S\|_F^2 \\
&\quad - \log[\mathrm{Gamma}(\gamma|\mathscr{H})\mathrm{Gamma}(\tau|\mathscr{H})\mathrm{Gamma}(\beta|\mathscr{H})] + \mathrm{C},
\end{aligned} \tag{28}
$$

where C denotes a constant, Θ is the set of all model parameters, $g_{BB}(\cdot|\mathscr{H})$ is the Beta-Bernoulli prior in (9) or (11), and $\mathscr{H}$ represents the set of all model hyperparameters.

For the low-rank component, rather than employing a constraint on Frobenius matrix norm to impose the sparseness of the singular values by using a Gaussian prior, we employ an Exponent-Gamma prior to obtain a constraint on weight nuclear norm, together with a Beta-Bernoulli distribution to encourage the sparseness of the singular values (like [11]). Note that for the low-rank component, an exponent prior is imposed on the singular values so that a big weight for a small nonzero singular value is desired. The main difference between the proposed APSV-BRPCA model and the WNNM-RPCA model is that we use numerical methods to estimate the distribution of unknown parameters, while one effectively seeks a single solution to minimize a function in WNNM-RPCA.

5.2 Convergence Analysis

The convergence of the proposed algorithm can be transformed into the convergence of the variational Bayesian method because the proposed algorithm is derived based on the variational Bayesian method. The expression (16) of the optimal solution $q(\Theta_k)$ depends on the expectation of other factors $q(\Theta_j)$ for $j \neq k$. To obtain the maximum value of the lower bound (15), all the factors need to be circulated. One needs to initialize all the factors $q(\Theta_j)$ for conducting the variational inference. Then, using the current solution of all other factors, we can estimate each factor in turn with the updated value obtained in (16). Since the bound is convex for each of the factors $q(\Theta_j)$, convergence is guaranteed [2,20].

5.3 MCMC-Based Bayesian Inference

The posterior density function can also be approximated using Gibbs sampler. In the Gibbs sampler, the Markov chain Monte Carlo (MCMC) analysis is implemented and the posterior distributions of model parameters are approximated by samples drawing from the corresponding conditional posterior distributions, respectively. The MCMC-based Bayesian inference method is summarized in Algorithm 2, where $\mathbf{y}_s$ represents the sth column in Y and $(a|-)$ represents random variable a given all the other ran-

dom variables in the model, $\mathcal{N}(\mu, \Sigma)_+$ represents a truncation normal distribution from 0.

Algorithm 2 MCMC-based Bayesian Inference for APSV-BRPCA

Input: The measurement Y, randomly initialize parameters $\Theta = \{a_0, b_0, c_0, d_0, e_0, f_0, \theta_0, \eta_0, \alpha_0, \beta_0, U, D, \Lambda, V, B, E\}$
Output: $\{\Theta^{(i)}\}_{i=N_{burn-in}+1:N_{burn-in}+N_{collect}}\}$
for $iter = 1$ to $N_{burn-in} + N_{collect}$ **do**
% low-rank component
for $s = 1$ to R **do**
$\tilde{\mathbf{y}}_j^{-s} \leftarrow \mathbf{y}_j - U(D\Lambda)\mathbf{v}_{j\cdot}^T + d_{ss}\lambda_{ss}v_{js}\mathbf{u}_{\cdot s} - \mathbf{b}_{\cdot j} \circ \mathbf{e}_{\cdot j}$ for $j = 1, \ldots, n$
$(\mathbf{u}_{\cdot s}|-) \sim \mathcal{N}(\mu, \Sigma)$, $\Sigma = (\beta \sum_{j=1}^n \lambda_{ss}^2 d_{ss}^2 v_{js}^2 + mI_m)^{-1}$, $\mu = \beta\Sigma \sum_{j=1}^n \lambda_{ss} d_{ss} v_{js} \tilde{\mathbf{y}}_j^{-s}$
$(d_{ss}|-) \sim Bernoulli(\frac{q_1}{q_0+q_1})$, $q_0 = 1 - p_s$, $q_1 = p_s \exp(-\frac{\beta}{2}\sum_{j=1}^n(\lambda_{ss}^2 v_{js}^2 \mathbf{u}_{\cdot s}^T\mathbf{u}_{\cdot s} - 2\lambda_{ss}v_{js}\mathbf{u}_{\cdot s}^T\tilde{\mathbf{y}}_j^{-s}))$
$(\lambda_{ss}|-) \sim \mathcal{N}(\mu, \Sigma)_+$, $\Sigma = (\beta \sum_{j=1}^n d_{ss}^2 v_{js}^2 \mathbf{u}_{\cdot s}^T\mathbf{u}_{\cdot s})^{-1}$, $\mu = \Sigma(\beta \sum_{j=1}^n d_{ss} v_{js} \mathbf{u}_{\cdot s}^T\tilde{\mathbf{y}}_j^{-s} - \gamma_s)$
$(v_{js}|-) \sim \mathcal{N}(\mu, \Sigma)$, $\Sigma = (\beta\lambda_{ss}^2 d_{ss}^2 \mathbf{u}_{\cdot s}^T\mathbf{u}_{\cdot s} + 1)^{-1}$, $\mu = \beta\Sigma\lambda_{ss}d_{ss}\mathbf{u}_{\cdot s}^T\tilde{\mathbf{y}}_j^{-s}$ for $j = 1, \ldots, n$
$(p_s|-) \sim Beta(\theta_0 + d_{kk}, \eta_0 + 1 - d_{kk})$
$(\gamma_s|-) \sim Gamma(a_0 + 1, b_0 + \lambda_{kk})$
end for
% sparse component
$\tilde{\mathbf{y}}_j \leftarrow \mathbf{y}_j - U(D\Lambda)\mathbf{v}_{j\cdot}^T$ for $j = 1, \ldots, n$
$(b_{ij}|-) \sim Berboulli(\frac{q_1}{q_1+q_0}), q_1 = \omega_i \exp(-\frac{\beta}{2}(e_{ij}^2 - 2e_{ij}\tilde{y}_{ij})), q_0 = 1-\omega_i$ for $i = 1, \ldots, m$, $j = 1, \ldots, n$
$(e_{ij}|-) \sim \mathcal{N}(\mu, \Sigma)$, $\Sigma = (\tau + \beta b_{ij}^2)^{-1}$, $\mu = \beta\Sigma b_{ij}\tilde{y}_{ij}$ for $i = 1, \ldots, m$, $j = 1, \ldots, n$
$(\omega_i|-) \sim Beta(\alpha_0 + \sum_{j=1}^n b_{ij}, \beta_0 + n - \sum_{j=1}^n b_{ij})$ for $i = 1, \ldots, m$
$(\tau|-) \sim Gamma(c_0 + 0.5mn, d_0 + 0.5\sum_{j=1}^n \mathbf{e}_{\cdot j}^T\mathbf{e}_{\cdot j})$
% noise component
$(\beta|-) \sim Gamma(e_0 + 0.5mn, f_0 + 0.5\|Y - UD\Lambda V^T - B \circ E\|_F^2)$
% collecting samples
if $iter => N_{burn-in}$ **then**
$\Theta^{iter-N_{burn-in}} \leftarrow \Theta$
end if
end for

5.4 Computational Complexity

In each iteration of the VB inference, the matrix inversion is required and leads to much computational cost. More specifically, to obtain the approximate posterior of each column $e_{\cdot j}$, one needs to calculate the inverses of n m-order matrices. The computational complexity is $n\mathcal{O}(m^3)$ in each iteration. In addition to the above computational cost, the rest posterior parameters can be rapidly computed. For the MCMC-based Bayesian inference, the computational complexity of each sampling is approximately the same as the VB method.

6 Experiments

In this section, we illustrate the recovery accuracy and efficiency of the proposed APSV-BRPCA algorithm compared with four state-of-the-art methods: augmented Lagrange multiplier method (ALM) [24], weight nuclear norm minimization for RPCA (WNNM-RPCA) [17], Bayesian robust principal component analysis (BRPCA) [11], and sparse Bayesian methods for low-rank matrix estimation (VBRPCA) [1]. All experiments are implemented in Matlab R2014b on a PC with 4.0 GHz CPU and 31.4 GB RAM.

6.1 Parameter Setting

In our experiments, very small values (e.g., 10^{-6}) are assigned to the parameters $c_0, d_0, e_0, f_0, \theta_0, \eta_0, \alpha_0, \beta_0$, respectively. Such strategies can lead to broad hyperpriors. Next, we illustrate the choice of tuning parameters a_0 and b_0. In [17], an effective reweighted strategy is proposed to assign weights on different singular values by the following formula

$$\omega_k^l = \frac{C}{\sigma_k(L_l) + \varepsilon}, \tag{29}$$

where $\sigma_k(L_l)$ is the kth singular value in the lth iteration and ω_k^l is the corresponding regularization parameter in the lth iteration, C is chosen as the square root of matrix size, i.e., $C = \sqrt{mn}$ and ε is a small positive number to make the inequality $\varepsilon < \min(\sqrt{C}, \frac{C}{\sigma_1(Y)})$ hold. Comparing (29) with the expectation of the posterior of weight γ_k in (22), we can see that the tuning parameters a_0 and b_0 correspond to the parameters C and ε, respectively. Hence, we let $a_0 \in (0, \sqrt{mn}]$ and let $b_0 \in (0, \min(\sqrt{C}, \frac{C}{\sigma_1(Y)})]$. We randomly generate a synthetic matrix of size 300×300 with rank $\mathbf{rank}(L)$. The low-rank component of the synthetic matrix consists of a product of two matrices generated from a standard Gaussian distribution. The sparse component consists of nonzero entries drawn uniformly in the range $[-100, 100]$ and located uniformly at random. We use notations $\rho_r = \mathbf{rank}(L)/N$ and $\rho_s = \|S\|_0/N^2$, where $N = 300 and \rho_r, \rho_s \in \{0.05, 0.15, 0.25, 0.3\}$ represent the rank and the sparsity, respectively. The noiseless case and the noisy case with $\sigma^2 = 10^{-3}$ are considered.

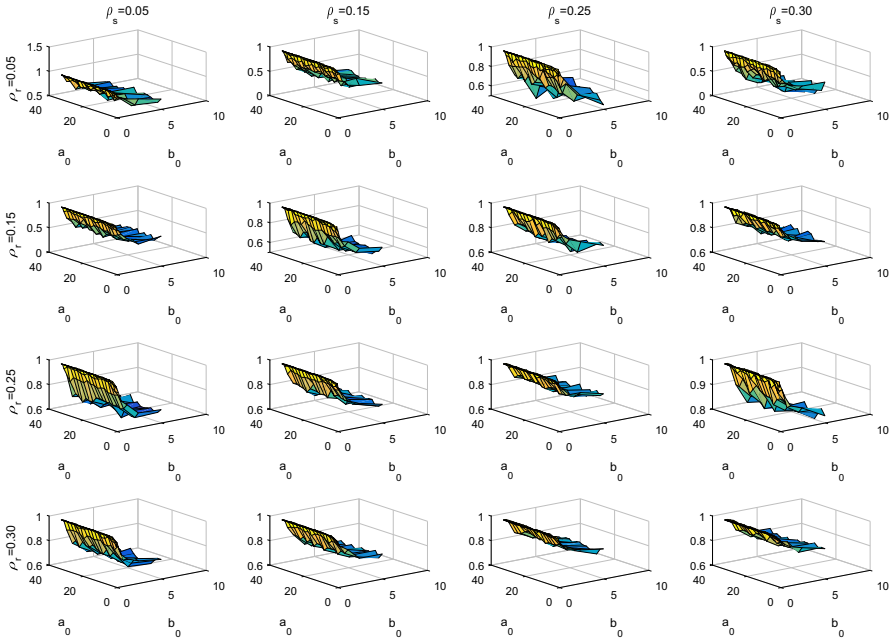

Fig. 2 The noiseless case: reconstruction errors of low-rank $\|L-\hat{L}\|_F/\|L\|_F$ for varying ranks and sparsity computed by APSV-BRPCA

The experimental results with $a_0 \in (0, 40]$ and $b_0 \in (0, 7]$ are shown in Figs. 2 and 3. It can be seen that the preferable parameter b_0 can be obtained in the range [4, 7], and the parameter a_0 can be chosen in the range [1, 40]. The experiments of synthetic matrices of sizes 100×100, 500×500 and 1000×1000 show the same results. Meanwhile, the experiment results demonstrate that the tuning parameters a_0 and b_0 are robust for different ranks, sparsity and noise levels. Without loss of generality, we set the tuning parameters $a_0 = 10$ and $b_0 = 4$ in all experiments.

6.2 Synthetic Experiments

In synthetic experiments, we randomly generate two low-rank square matrices with sizes 200×200, 500×500 and ranks 5, 20, respectively. Each of these matrices is generated by the product of two matrices which obey a standard Gaussian distribution. The magnitudes of nonzero entries in S are drawn uniformly in the range $[-100, 100]$. The sparsity levels ρ of different sparse matrices with sizes 200×200, 500×500 are 0.05 and 0.08, respectively. We introduce the following metrics to evaluate the recovery performance: the reconstruction errors of low-rank and sparse components, $\mathbf{rank}(\hat{X})$, $||\hat{S}||_0$ (the number of nonzero elements). For synthetic matrices with sizes 200×200, the noiseless case and the noisy case with $\sigma^2 = 10^{-2}$ are considered. The noise with variance $\sigma^2 = 10^{-3}$ is added to synthetic matrices with size 500×500.

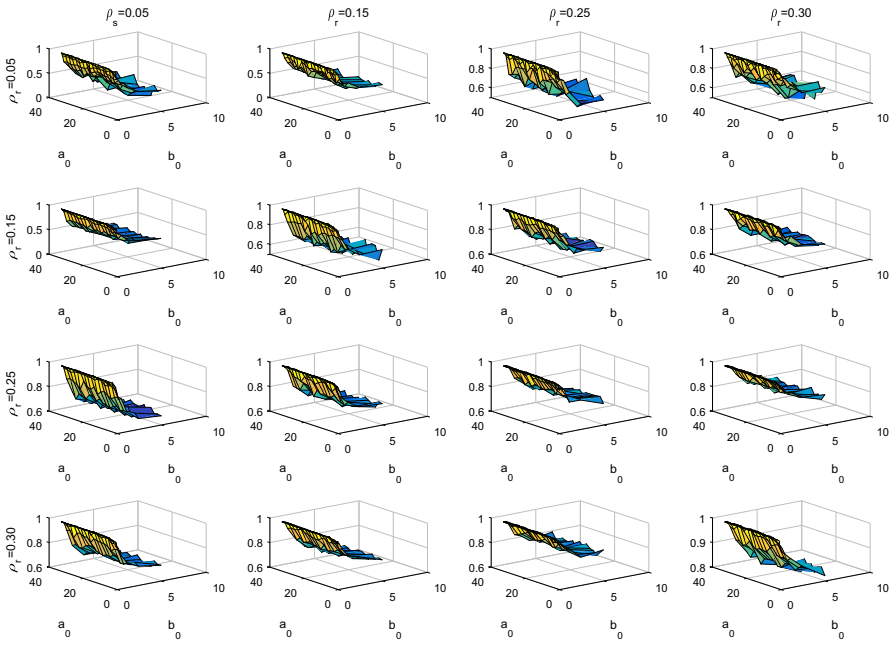

Fig. 3 The noise case: reconstruction errors of low-rank $\|L - \hat{L}\|_F / \|L\|_F$ for varying ranks and sparsity computed by APSV-BRPCA

Experimental results are shown in Table 1. To avoid the random disturbance of Gaussian noise, the experiment results are calculated by averaging the results of 100 independent trials. From Table 1, it can be seen that WNNM-RPCA achieves minimal error among all algorithms in the noiseless case. However, its performance is not robust for noise and corrupted as the noise level increases. The Bayesian algorithms achieve greater robustness than deterministic approaches. The reason may be that the Bayesian approaches regard the noise as a random variable. In addition, the proposed APSV-BRPCA algorithm has the best performance compared with other algorithms in the noisy case.

We demonstrate the ability of APSV-BRPCA by varying ranks, sparsity and noise levels. In this experiment, synthetic matrices with size 100×100 are generated. For each triple tuple $(\rho_r, \rho_s, \sigma^2)$, $\sigma^2 \in \{0, 10^{-4}, 10^{-3}, 10^{-2}\}$, ρ_r, and $\rho_s \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4\}$ represent the noise variance, the rank and the sparsity, respectively. We conduct ten random experiments. The following criterion proposed in [11] is employed to measure the successful reconstruction of L, i.e., $(\|L - \hat{L}\|_F / \|L\|_F) \leq Th$, where Th is a threshold depending on the noise level. The noise standard deviations of different thresholds Th with magnitudes $10^{-4}, 5 \times 10^{-3}, 10^{-2}, 10^{-1}$ are $\sigma^2 = 0, 10^{-4}, 10^{-3}$ and 10^{-2}, respectively.

Figure 4 depicts the fraction of successful recovery of different algorithms for each triple tuple. The experimental results show that the proposed APSV-BRPCA algorithm has more comprehensive ability than other methods in terms of recovery accuracy and

Table 1 Reconstruction errors, estimated rank and the number of nonzero elements

Method	σ^2	Size	Rank(L)	$\|S\|_0$	Rank($\hat{L}$)	$\|\hat{S}\|_0$	$\frac{\|\hat{L}-L\|_F}{\|L\|_F}$	$\frac{\|\hat{S}-S\|_F}{\|S\|_F}$
ALM	0	200×200	5	2000	**5**	**2000**	1.117×10^{-8}	1.995×10^{-8}
WNNM-RPCA	0	200×200	5	2000	**5**	**2000**	$\mathbf{9.226 \times 10^{-9}}$	$\mathbf{1.913 \times 10^{-8}}$
VBRPCA	0	200×200	5	2000	**5**	**2000**	2.499×10^{-7}	6.514×10^{-8}
BRPCA	0	200×200	5	2000	**5**	**2000**	1.709×10^{-6}	6.216×10^{-8}
APSV-BRPCA	0	200×200	5	2000	**5**	**2000**	1.6881×10^{-6}	6.143×10^{-8}
ALM	10^{-2}	200×200	5	2000	117	25499	2.467×10^{-2}	2.209×10^{-2}
WNNM-RPCA	10^{-2}	200×200	5	2000	**5**	38837	1.246×10^{-2}	3.073×10^{-2}
VBRPCA	10^{-2}	200×200	5	2000	**5**	35212	1.718×10^{-2}	3.236×10^{-2}
BRPCA	10^{-2}	200×200	5	2000	**5**	2002	1.010×10^{-2}	3.504×10^{-4}
APSV-BRPCA	10^{-2}	200×200	5	2000	**5**	**2001**	$\mathbf{9.798 \times 10^{-3}}$	$\mathbf{3.489 \times 10^{-4}}$
ALM	10^{-3}	500×500	20	20000	285	1690701	4.443×10^{-3}	6.252×10^{-3}
WNNM-RPCA	10^{-3}	500×500	20	20000	22	235272	7.537×10^{-2}	8.293×10^{-2}
VBRPCA	10^{-3}	500×500	20	20000	**20**	205883	3.507×10^{-3}	8.049×10^{-3}
BRPCA	10^{-3}	500×500	20	20000	**20**	**20001**	2.037×10^{-3}	1.151×10^{-4}
APSV-BRPCA	10^{-3}	500×500	20	20000	**20**	20003	$\mathbf{1.247 \times 10^{-3}}$	$\mathbf{1.031 \times 10^{-4}}$

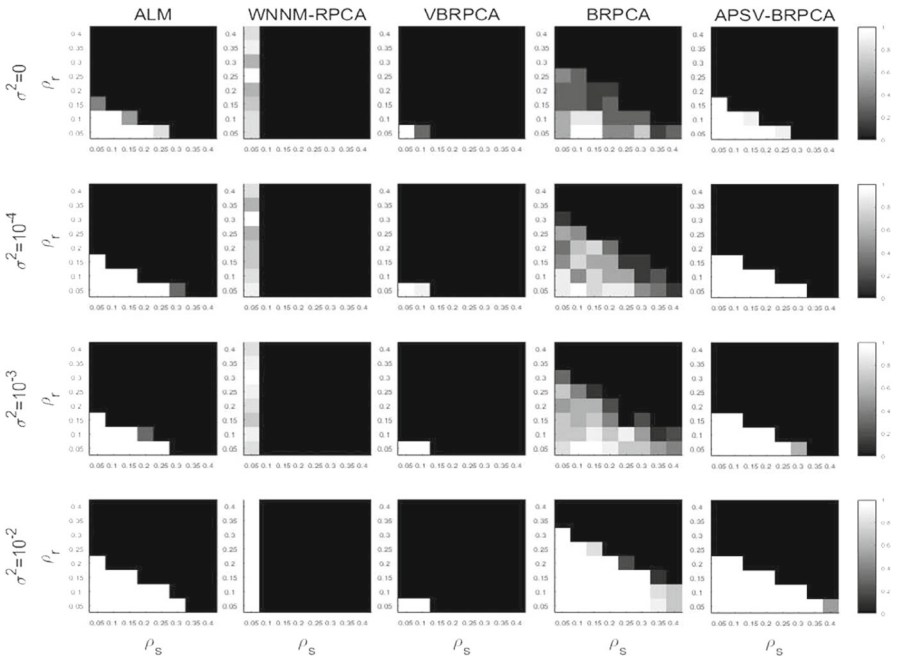

Fig. 4 Fraction of successful recovery for varying ranks, sparsity, and noise levels, computed by ALM, WNNM-RPCA, VBRPCA, BRPCA and APSV-BRPCA. Given a tuple $(\rho_r, \rho_s, \sigma^2)$, white represents that all the 10 trials are successfully recovered, and black means no trials is successfully recovered

robustness. As shown in Fig. 5, the MMSE demonstrates the effectiveness of the proposed algorithm.

6.3 Video Example

In this section, we investigate the application of background subtraction with different methods. The objective of background subtraction is to reconstruct the static background and the moving foreground from the video sequence [11]. Background subtraction can be considered as a binary segmentation of video sequences, in which the video sequences consist of the invariant background and the time-varying foreground. A video stream can be modeled as the combination of low-rank and sparse components, since static backgrounds of different frames are the same, while moving foregrounds are relatively sparse. If we stack the video frames as columns of a matrix, the low-rank component represents the static background and the sparse component corresponds to the moving foreground. Under this assumption, we can use RPCA to reconstruct low-rank and sparse components from the video stream. A video surveillance with slow-changing foreground is considered for examining the performance of different algorithms.

The video sequence is sampled from a shopping center [11], which consists of 158 frames with resolution 144×192. We stack the video sequence as an observed matrix

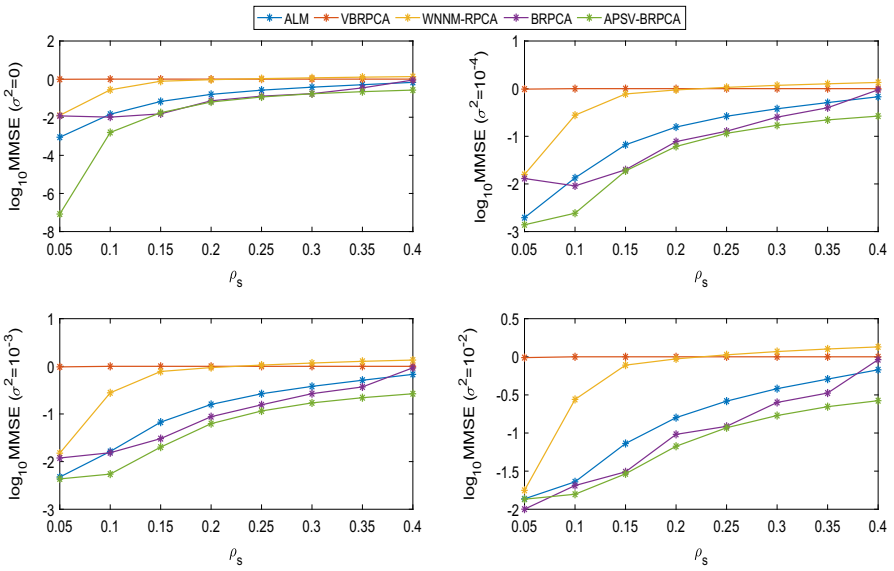

Fig. 5 MMSE for rank equals 0.05 where MMSE $=$ mean$(\frac{||\hat{Y}-Y||_F}{||Y||_F})$ and varying sparsity and noise levels, computed by ALM, WNNM-RPCA, VBRPCA, BRPCA and APSV-BRPCA. Given a tuple $(0.2, \rho_s, \sigma^2)$, color point represents that mean of the 10 trials

of size 27648×158. The noiseless case is shown in Fig. 6. It can be observed that the proposed algorithm and BRPCA can effectively distinguish the slow foreground and the static background and achieve better results than ALM and WNNM-RPCA. Furthermore, the Gaussian noise with mean 0 and variance 25 is added to the observed data and the result is shown in Fig. 7. It suggests that our approach gets satisfied results in discovering the low-rank background and sparse foreground from the observed matrix and shows that ALM algorithm, WNNM-RPCA algorithm and VBRPCA algorithm are sensitive to noise.

To evaluate the reconstruction performance of different algorithms, we plot the background standard deviation and error bars for the noiseless case and the noisy case with $\sigma^2 = 25$ in Figs. 8 and 9, respectively. For each pixel, the background standard deviation is defined as the standard deviation of the extracted background of the video sequence, and the error bars are calculated as the standard deviation of the original noiseless video sequence minus the summation of extracted background and foreground. From Fig. 8, we observe that the error bars of the proposed algorithm and BRPCA are similar; however, the extracted background via the proposed algorithm is more robust. In Fig. 9, it can be seen that, for most inferred pixels, the proposed algorithm presents more accurate entries of background standard deviation than BRPCA. At the same time, we can also observe that the proposed algorithm achieves the lowest error bars. It demonstrates the ability of our method in dealing with the real data.

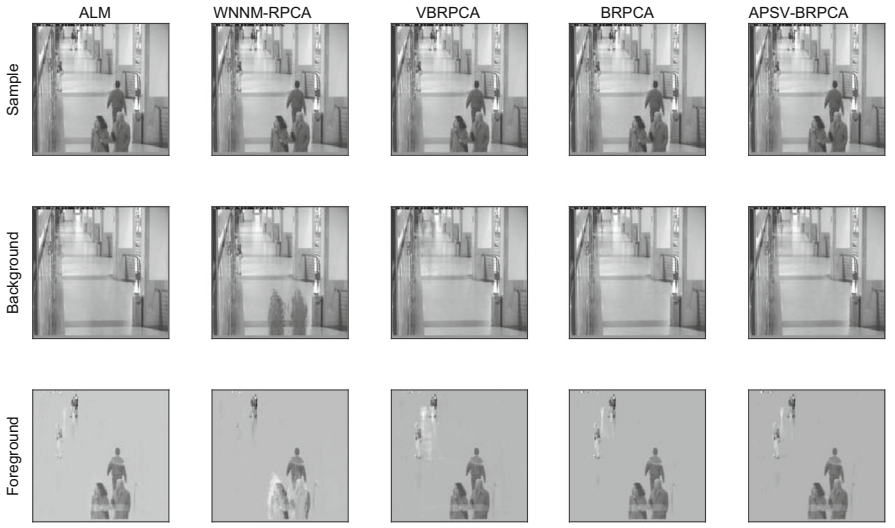

Fig. 6 Reconstruction of the background and the foreground. First row: original image; second row: reconstruction of the low-rank component (background); bottom row: reconstruction of the sparse component (foreground)

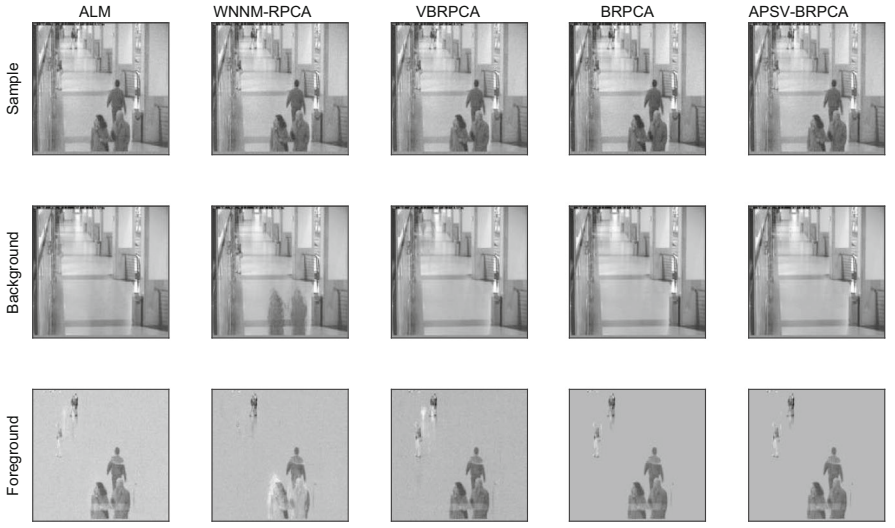

Fig. 7 Reconstruction of the background and the foreground under noisy observation. The additive white Gaussian noise has variance $\sigma^2 = 25$. First row: original image; second row: reconstruction of the low-rank component (background); bottom row: reconstruction of the sparse component (foreground)

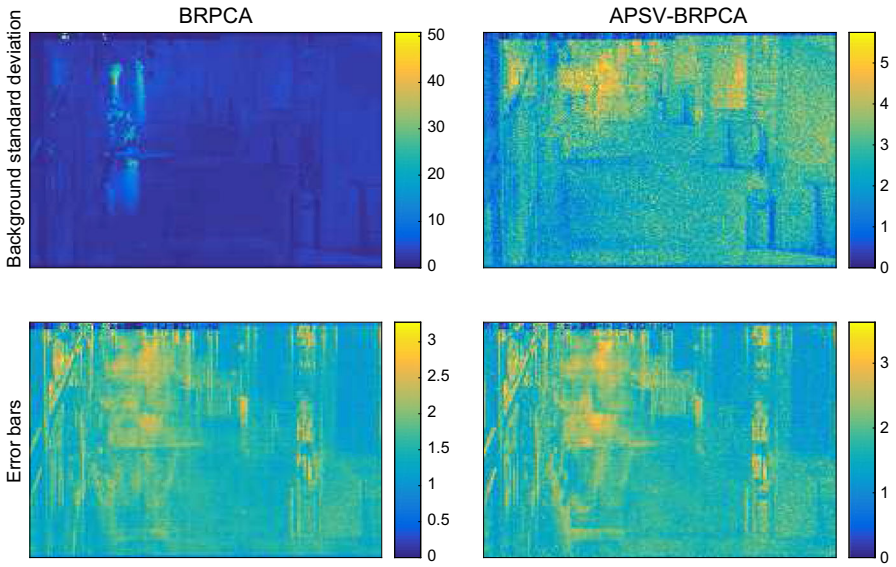

Fig. 8 The noise-free case: background standard deviation (first row) and error bars (second row) of the proposed APSV-BRPCA algorithm and BRPCA algorithm

7 Conclusion

In this paper, we propose a novel Bayesian RPCA model by introducing new penalty strategies on the singular values of the low-rank component. The key ingredient of the proposed method is the adaptive penalty on the singular values, which uses a more realistic low-rank prior model that goes beyond the simple low-rank prior. Specifically, in the proposed Bayesian RPCA model, an Exponent-Gamma framework is employed, which adaptively imposes weights on different singular values so that it potentially improves the flexibility of the exiting Bayesian models. The parameters can be estimated by variational Bayesian inference and MCMC-based Bayesian inference. Meanwhile, the proposed model is closely related to the deterministic models, such as the reweighted ℓ_1 minimization. We systematically compare the recovery performance of different Bayesian and deterministic models to demonstrate the advantages of the proposed model. In comparison with the numerical experiment results of these models, we observe that the proposed Bayesian model has more comprehensive ability than other methods in noise cases. Moreover, the real case shows that the proposed Bayesian model achieves the lowest error bars and is more robust. These experimental evaluations with simulated and real data demonstrate the superiority of the proposed Bayesian model.

One main limitation of the proposed model in practical applications is that the measurement noise is assumed to be Gaussian distribution. In the real cases, the noises in measurements are complex and may copy with non-Gaussian distributions. In the future research, we shall focus on the non-Gaussian noises. Besides, to achieve faster convergence and better reconstruction performance, some optimal algorithms, such

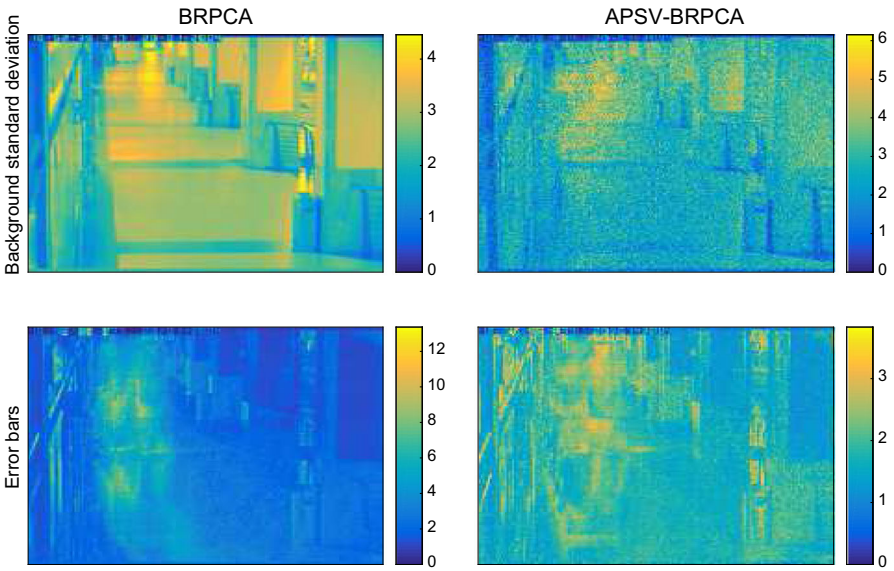

Fig. 9 The noise case where the additive white Gaussian noise has variance 25 and mean 0: background standard deviation (first row) and error bars (second row) of the proposed APSV-BRPCA algorithm and BRPCA algorithm

as [31] and [33], are going to be used to select the optimal values of the parameters in the future research.

Acknowledgements The authors would like to express their sincere gratitude to the anonymous referees, the editor for many valuable suggestions and comments that helped to improve the paper. This work was supported by the National Natural Science Foundation of China (Grant No. 91746107) and the China Scholarship Council.

References

1. S.D. Babacan, M. Luessi, R. Molina, A.K. Katsaggelos, Sparse Bayesian methods for low-rank matrix estimation. IEEE Trans. Signal Process. **60**(8), 3964–3977 (2012)
2. S.D. Babacan, S. Nakajima, M.N. Do, Bayesian group-sparse modeling and variational inference. IEEE Trans. Signal Process. **62**(11), 2906–2921 (2015)
3. C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Berlin, 2006)
4. J. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2008)
5. E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM **58**(3), 11 (2009)
6. E.J. Candès, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization. J. Fourier Anal. Appl. **14**(5–6), 877–905 (2008)
7. X. Cao, Y. Chen, Q. Zhao, D. Meng, Y. Wang, D. Wang, Z. Xu, Low-rank matrix factorization under general mixture noise distributions, in *IEEE International Conference on Computer Vision*, (2016), pp. 1493–1501
8. G.C. Cawley, N.L.C. Talbot, Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters. J. Mach. Learn. Res. **8**, 841–861 (2007)
9. F. Chen, R. Hu, H. Yu, S. Wang, Reduced set density estimator for object segmentation based on shape probabilistic representation. J. Vis. Commun. Image Represent. **23**(7), 1085–1094 (2012)

10. Y. Chen, X. Cao, Q. Zhao, D. Meng, Z. Xu, Denoising hyperspectral image with non-i.i.d. noise structure. IEEE Trans. Cybern. **48**(3), 1054–1066 (2018)
11. X. Ding, L. He, L. Carin, Bayesian robust principal component analysis. IEEE Trans. Image Process. **20**(12), 3419–3430 (2011)
12. Z. Fan, X. Yong, W. Zuo, Y. Jian, J. Tang, Z. Lai, D. Zhang, Modified principal component analysis: an integration of multiple similarity subspace models. IEEE Trans. Neural Netw. Learn. Syst. **25**(8), 1538–1552 (2017)
13. J. Gao, Robust l_1 principal component analysis and its Bayesian variational inference. Neural Comput. **20**(2), 555–578 (2008)
14. A. Ghaani Farashahi, Cyclic wave packet transform on finite abelian groups of prime order. Int. J. Wavelets Multiresolut. Inf. Process. **12**(06), 1450041 (2014)
15. A. Ghaani Farashahi, Wave packet transforms over finite cyclic groups. Linear Algebra Appl. **489**, 75–92 (2016)
16. P. Giordani, H.A.L. Kiers, A comparison of three methods for principal component analysis of fuzzy interval data. Comput. Stat. Data Anal. **51**(1), 379–397 (2006)
17. S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, L. Zhang, Weighted nuclear norm minimization and its applications to low level vision. Int. J. Comput. Vis. **121**(2), 183–208 (2017)
18. G. Han, J. Wang, X. Cai, Background subtraction based on modified online robust principal component analysis. Int. J. Mach. Learn. Cybern. **8**(6), 1839–1852 (2017)
19. N. Han, Y. Song, Z. Song, Bayesian robust principal component analysis with structured sparse component. Comput. Stat. Data Anal. **109**, 144–158 (2017)
20. N. Han, Z. Song, Bayesian multiple measurement vector problem with spatial structured sparsity patterns. Digit. Signal Process. **75**, 184–201 (2018)
21. N. Han, Z. Song, Y. Li, Cluster-based image super-resolution via jointly low-rank and sparse representation. J. Vis. Commun. Image Represent. **38**, 175–185 (2016)
22. Y. Hu, D. Zhang, J. Ye, X. Li, X. He, Fast and accurate matrix completion via truncated nuclear norm regularization. IEEE Trans. Pattern Anal. Mach. Intell. **35**(9), 2117–2130 (2013)
23. O.Y. Lee, J.W. Lee, J.O. Kim, Combining self-learning based super-resolution with denoising for noisy images. J. Vis. Commun. Image Represent. **48**, 66–76 (2017)
24. Z. Lin, M. Chen, Y. Ma, The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv:1009.5055 (2010)
25. G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 171–184 (2013)
26. S. Liu, H. Wu, Y. Huang, Y. Yang, J. Jia, Accelerated structure-aware sparse Bayesian learning for 3D electrical impedance tomography. IEEE Trans. Ind. Inform. **15**(9), 5033–5041 (2019)
27. S. Liu, Y.D. Zhang, T. Shan, R. Tao, Structure-aware Bayesian compressive sensing for frequency-hopping spectrum estimation with missing observations. IEEE Trans. Signal Process. **66**(8), 2153–2166 (2018)
28. C. Lu, J. Tang, S. Yan, Z. Lin, Nonconvex nonsmooth low-rank minimization via iteratively reweighted nuclear norm. IEEE Trans. Image Process. **25**(2), 829–839 (2016)
29. J. Luttinen, A. Ilin, J. Karhunen, Bayesian robust *pca* for incomplete data, in *International Conference on Independent Component Analysis and Signal Separation*, (2009), pp. 66–73
30. A.B. Musa, A comparison of ℓ_1-regularization, PCA, KPCA and ICA for dimensionality reduction in logistic regression. Int. J. Mach. Learn. Cybern. **5**(6), 861–873 (2014)
31. N. Nedic, D. Prsic, L. Dubonjic, V. Stojanovic, V. Djordjevic, Optimal cascade hydraulic control for a parallel robot platform by PSO. Int. J. Adv. Manuf. Technol. **72**(5–8), 1085–1098 (2014)
32. N. Nedic, D. Prsic, C. Fragassa, V. Stojanovic, A. Pavlovic, Simulation of hydraulic check valve for forestry equipment. Int. J. Heavy Veh. Syst. **24**(3), 260–276 (2017)
33. D. Prsic, N. Nedic, V. Stojanovic, A nature inspired optimal control of pneumatic-driven parallel robot platform. Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci. **231**(1), 59–71 (2017)
34. S. Serneels, T. Verdonck, Principal component analysis for data containing outliers and missing elements. Comput. Stat. Data Anal. **52**(3), 1712–1727 (2008)
35. A. Sharma, K. Paliwal, S. Imoto, S. Miyano, Principal component analysis using QR decomposition. Int. J. Mach. Learn. Cybern. **4**(6), 679–683 (2013)
36. V. Stojanovic, V. Filipovic, Adaptive input design for identification of output error model with constrained output. Circuits Syst. Signal Process. **33**(1), 97–113 (2014)

37. V. Stojanovic, N. Nedic, D. Prsic, L. Dubonjic, Optimal experiment design for identification of ARX models with constrained output in non-Gaussian noise. Appl. Math. Model. **40**(13–14), 6676–6689 (2016)
38. K.C. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. Pac. J. Optim. **6**(3), 615–640 (2011)
39. A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, Y. Ma, Toward a practical face recognition system: robust alignment and illumination by sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **34**(2), 372–386 (2011)
40. S. Yi, Z. Lai, Z. He, Y. Liu, Joint sparse principal component analysis. Pattern Recognit. **61**, 524–536 (2017)
41. Z. Zhang, B.D. Rao, Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning. IEEE J. Sel. Top. Signal Process. **5**(5), 912–926 (2011)
42. Q. Zhao, D. Meng, Z. Xu et al., l_1-norm low-rank matrix factorization by variational *B*ayesian method. IEEE Trans. Neural Netw. Learn. Syst. **26**(4), 825–839 (2015)
43. Q. Zhao, D. Meng, Z. Xu, W. Zuo, L. Zhang, Robust principal component analysis with complex noise, in *International Conference on Machine Learning*, (2014), pp. 55–63