Low-rank Matrix Factorization under General Mixture Noise Distributions

Xiangyong Cao[1] Yang Chen[1] Qian Zhao[1] Deyu Meng[1,2,*] Yao Wang[1] Dong Wang[1] Zongben Xu[1,2]
[1]School of Mathematics and Statistics, Xi'an Jiaotong University
[2]Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University
{caoxiangyong45, chengyang9103, timmy.zhaoqian}@gmail.com, dymeng@mail.xjtu.edu.cn
{yao.s.wang, xjtumathwd}@gmail.com, zbxu@mail.xjtu.edu.cn

Abstract

Many computer vision problems can be posed as learning a low-dimensional subspace from high dimensional data. The low rank matrix factorization (LRMF) represents a commonly utilized subspace learning strategy. Most of the current LRMF techniques are constructed on the optimization problem using L_1 norm and L_2 norm, which mainly deal with Laplacian and Gaussian noise, respectively. To make LRMF capable of adapting more complex noise, this paper proposes a new LRMF model by assuming noise as Mixture of Exponential Power (MoEP) distributions and proposes a penalized MoEP model by combining the penalized likelihood method with MoEP distributions. Such setting facilitates the learned LRMF model capable of automatically fitting the real noise through MoEP distributions. Each component in this mixture is adapted from a series of preliminary super- or sub-Gaussian candidates. An Expectation Maximization (EM) algorithm is also designed to infer the parameters involved in the proposed PMoEP model. The advantage of our method is demonstrated by extensive experiments on synthetic data, face modeling and hyperspectral image restoration.

1. Introduction

Many computer vision, machine learning, data mining and statistical problems can be formulated as the problem of extracting the intrinsic low dimensional subspace from input high-dimensional data. The extracted subspace tends to deliver the refined latent knowledge underlying data and thus has a wide range of applications including structure from motion [28], face recognition [33], collaborative filtering [14], information retrieval [7], social networks [5], object recognition [29], layer extraction [12] and plane-based pose estimation [27].

Low rank matrix factorization (LRMF) is one of the most commonly utilized techniques for subspace learning. Given a data matrix $\mathbf{Y} \in \mathcal{R}^{m\times n}$ with entries $y_{ij}s$, the LRMF problem can be mathematically formulated as

$$\min_{\mathbf{U},\mathbf{V}} ||\mathbf{W} \odot (\mathbf{Y} - \mathbf{U}\mathbf{V}^T)||, \tag{1}$$

where $\mathbf{W}$ is the indicator matrix with $w_{ij} = 0$ if y_{ij} is missing and 1 otherwise, and $\mathbf{U} \in \mathcal{R}^{m\times r}$ and $\mathbf{V} \in \mathcal{R}^{n\times r}$ are low-rank matrices ($r < min(m, n)$). The operator $\odot$ denotes the Hadamard product (the component-wise multiplication) and $||\cdot||$ corresponds to a certain noise measure.

Under the assumption of Gaussian noises, it is natural to utilize the L_2 norm (Frobenius norm) as the noise measure, which has been extensively studied in LRMF literatures [26, 3, 23, 1, 35, 24, 32, 22]. However, it has been recognized in many real applications that methods constructed on this model are sensitive to outliers and non-Gaussian noise. In order to introduce robustness, the L_1-norm-based models have attracted much attention recently [13, 8, 37, 15, 25, 11]. However, the L_1 norm is only optimal for Laplace-like noise and still very limited for handling various types of noise encountered in real problems. Taking the hyper-spectral image for example, it has been investigated in [34] that there are mainly two kinds of noise existing,

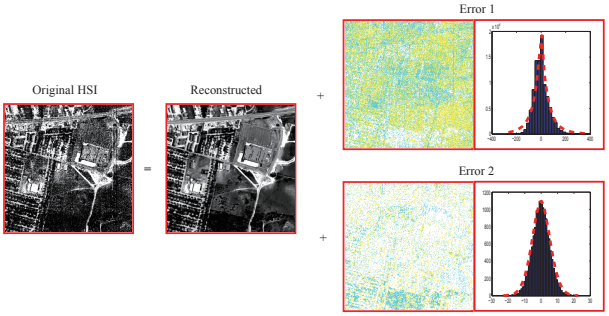

Figure 1. From left to right: Original hyperspectral image (HSI), reconstructed image, two extracted noise images with their histograms by the proposed methods. (Top: $EP_{0.2}$ noise image and histogram. Bottom: $EP_{1.8}$ noise image and histogram).

*Corresponding author.

i.e., sparse noise (stripe and deadline) and Gaussian-like noise, as depicted in Figure 1. The stripe noise is produced by the non-uniform sensor response which conducts the deviation of gray values of the original image continuously towards one direction. This noise always very sparsely appears in edges and texture areas of an image. The deadline noise, which is induced by some damaged sensor, results in the pixel value of the corresponding column to be 0 or very small value. The Gaussian-like noise is induced by some random disturbation during the transmission process of hyper-spectral signals. It is easy to see that such kind of complex noise cannot be well fit by either Laplace or Gaussian, which means that neither L_1 norm nor L_2 norm LRMF models are proper for this type of data.

Very recently, some novel models were presented to expand the availability of LRMF under more complex noise. The key idea is assuming the noise follows a more complicated mixture of Gaussians (MoG) [19], which is expected to better fit real noises, since the MoG constructs an universal approximator to any continuous density function in theory [17]. However, this method still cannot finely adapt real data noises. MoG can only approximate a complex distribution, e.g. Laplace, when the number of components goes to infinity, while in applications only a finite number of components can be specified. Besides, it also lacks a theoretically sound manner to properly select this number of Gaussian mixture. Moreover, the real noise might contain super- or sub-Gaussian components, which are beyond the representative ability of the current MoG.

In this paper, we propose a new LRMF method with a more general noise model to address the aforementioned issues. Specifically, we embed the noise into a mixture distribution of a series of sub- and super-Gaussians (i.e., general exponential power (EP) distributions), and formulate LRMF as a penalized MLE model, called PMoEP model. Then we design an expectation maximization (EM) algorithm to estimate the parameters involved in the model, and prove its convergence. The new method is not only capable of adaptively fitting complex real noise by EP noise components with proper parameters, but also able to learn the number of noise components from data, and thus can better recover the true low-rank matrix from corrupted data as verified by extensive experiments.

The rest of the paper is organized as follows: the related work regarding LRMF is introduced in Section 2. The PMoEP model and the corresponding EM algorithm are presented in Section 3. Experimental results are shown in Section 4. Finally, conclusions are drawn in Section 5. Throughout the paper, we denote scalars, vectors, matrices as the non-bold letters, bold lower case letters, and bold upper case letters, respectively.

2. Related Work

The L_2 norm LRMF with missing data has been studied for decades. Gabriel and Zamir [9] proposed a weighted SVD method as the early attempts to solve the L_2 norm LRMF with missing data. They used alternated minimization to find the principal subspace of the data. Srebro and Jaakkola [26] proposed the Weighted Low-rank Approximation (WLRA) algorithm to enhance efficiency of LRMF calculation. Buchanan and Fitzgibbon [3] further proposed a regularized model that adds a regularization term and then adopts the modified Levenberg-Marquardt (LM) algorithm to estimate the subspaces. However, it cannot handle large-scale problems due to the infeasibility of computing the Hessian matrix over a large number of variables. Okatani and Deguchi [23] showed that a Wiberg marginalization strategy on $\mathbf{U}$ and $\mathbf{V}$ can provide a better and robust initialization and proposed the Wiberg algorithm that updates U via least squares while updates V by a Gauss-Newton step in each iteration. Later, the Wiberg algorithm was extended to a damped version to achieve better convergence by Okatani et al. [24]. Aguiar et al. [1] deduced a globally optimal solution to L_2-LRMF with missing data under the assumption that the missing data has a special Young diagram structure. Zhao and Zhang [35] formulated the L_2-LRMF as a constrained model to improve its stability in real applications. Wen et al. [32] adopted the alternating strategy to solve the L_2-LRMF problem. Mitra et al. [22] proposed an augmented Lagrangian method to solve the L_2-LRMF problem for higher accuracy. However, all of these methods minimize the L_2 norm or its variations and is only optimal for Gaussian-like noise.

To make subspace learning method less sensitive to outliers, some robust loss functions have been investigated. For example, De la Torre and Black [6] adopted the Geman-McClure function and then used the iterative reweighted least square (IRLS) method to solve the induced optimization problem. In the last decade, the L_1 norm has become the most popular robust loss function. Ke and Kanade [13] replaced the L_2 norm with the L_1 norm for LRMF for the first time, and then solved the optimization by alternated convex programming (ACP) method. Kwak [15] later proposed to maximize the L_1 norm of the projection of data points onto the unknown principal directions instead of minimizing the residue. Eriksson and Hengel [8] experimentally showed that the ACP approach does not converge to the desired point with high probability, and thus introduced the L_1 Wiberg approach to address this issue. Zheng et al. [37] added more constraints to the factors $\mathbf{U}$ and $\mathbf{V}$ for L_1 norm LRMF, and solved the optimization by ALM, which improved the performance in structure from motion application. Within the probabilistic framework, Wang et al. [30] proposed probabilistic robust matrix factorization (PRMF) that modeled the noise as a Laplacian distribution,

which has been later extended to fully Bayesian settings by Wang and Yeung [31]. However, these methods optimize the L_1 norm and thus are only optimal for Laplace-like noise.

Beyond Gaussian or Laplace, other types of noise assumptions have also been attempted. Lakshminarayanan et al. [16] assumed that the noise is drawn from a student-t distribution. Babacan et al. [2] proposed a Bayesian methods for low-rank matrix estimation modeling the noise as a combination of sparse and Gaussian. To handle more complex noise, Meng and De la Torre [19] modeled the noise as a MoG distribution for LRMF, and later was extended to the Bayesian framework by Chen et al. [4] and to RPCA by Zhao et al. [36]. Although better than traditional methods, these methods are still very limited in dealing with complex noise in real scenarios.

3. LRMF with MoEP noise

In this section, we first propose a new LRMF model with MoEP noise, called PMoEP model, and then design an EM algorithm to solve it.

3.1. PMoEP model

In LRMF, from a generative perspective, each element $y_{ij}(i = 1, 2, \ldots, m, j = 1, 2, \ldots, n)$ of the data matrix $\mathbf{Y}$ can be modeled as

$$y_{ij} = \mathbf{u}_i \mathbf{v}_j^T + e_{ij}, \tag{2}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the i^{th} row of $\mathbf{U}$ and $\mathbf{V}$, respectively, and e_{ij} is the noise in y_{ij}. Instead of assuming the noise obeys MoG distributions, we assume that the noise e_{ij} follows mixture of Exponential Power (EP) distributions:

$$\mathbb{P}(e_{ij}) = \sum_{k=1}^{K} \pi_k f_{p_k}(e_{ij}; 0, \eta_k), \tag{3}$$

where π_k is the mixing proportion with $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$, K is the number of the mixture components and $f_{p_k}(e_{ij}; 0, \eta_k)$ denotes the k^{th} EP distribution with parameter η_k and $p_k (p_k > 0)$. Let $\mathbf{p} = [p_1, p_2, \ldots, p_K]$, in which each p_k can be variously specified. As defined in [21], the density function of the EP distribution ($p > 0$) with zero mean is

$$f_p(e; 0, \eta) = \frac{p\eta^{\frac{1}{p}}}{2\Gamma(\frac{1}{p})} \exp\{-\eta|e|^p\}, \tag{4}$$

where η is the precision parameter, p is the shape parameter and $\Gamma(\cdot)$ is the gamma function. By changing the shape parameter p, the EP distribution describes both leptokurtic ($0 < p < 2$) and platykurtic ($p > 2$) distributions. In particular, we obtain the Laplace distribution with $p = 1$, the Gaussian distribution with $p = 2$ and the Uniform distribution with $p \to \infty$ (see Figure 2). Therefore, MoG is special case of MoEP. By setting $\eta = 1/(p\sigma^p)$, the EP distribution (4) can be equivalently written as $EP(e; 0, p\sigma^p)$.

In our model, we assume that each noise e_{ij} is equipped with an indicator variable $\mathbf{z}_{ij} = [z_{ij1}, z_{ij2}, \ldots, z_{ijK}]^T$, where $z_{ijk} \in \{0, 1\}$ and $\sum_{k=1}^{K} z_{ijk} = 1$. $z_{ijk} = 1$ implies that the noise e_{ij} is drawn from the k^{th} EP distribution. $\mathbf{z}_{ij}$ obeys a multinomial distribution $\mathbf{z}_{ij} \sim \mathcal{M}(\boldsymbol{\pi})$, where $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_K]^T$. Then we have:

$$\mathbb{P}(e_{ij}|\mathbf{z}_{ij}) = \prod_{k=1}^{K} f_{p_k}(e_{ij}; 0, \eta_k)^{z_{ijk}}, \tag{5}$$

$$\mathbb{P}(\mathbf{z}_{ij}; \boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{z_{ijk}}. \tag{6}$$

Denoting $\mathbf{E} = (e_{ij})_{m\times n}$, $\mathbf{Z} = (\mathbf{z}_{ij})_{m\times n}$ and $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{\eta}, \mathbf{U}, \mathbf{V}\}$ with $\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_K]^T$, the *complete likelihood function* can be written as

$$\mathbb{P}(\mathbf{E}, \mathbf{Z}; \boldsymbol{\Theta}) = \prod_{i,j\in\Omega} \prod_{k=1}^{K} [\pi_k f_{p_k}(e_{ij}; 0, \eta_k)]^{z_{ijk}}, \tag{7}$$

where Ω is the index set of the non-missing entries in $\mathbf{Y}$. Then the *log-likelihood function* is

$$l(\boldsymbol{\Theta}) = \log \mathbb{P}(\mathbf{E}; \boldsymbol{\Theta}) = \log \sum_{\mathbf{Z}} \mathbb{P}(\mathbf{E}, \mathbf{Z}; \boldsymbol{\Theta}), \tag{8}$$

and the *complete log-likelihood function* is

$$\begin{aligned} l^C(\boldsymbol{\Theta}) &= \log \mathbb{P}(\mathbf{E}, \mathbf{Z}; \boldsymbol{\Theta}) \\ &= \sum_{i,j\in\Omega} \sum_{k=1}^{K} z_{ijk} [\log \pi_k + \log f_{p_k}(e_{ij}; 0, \eta_k)]. \end{aligned} \tag{9}$$

As aforementioned in introduction, determining the number of components K is an important problem for the mixture model. Here we adopt an effective method proposed by Huang et al. [10] for this aim of selecting EP mixture number, and construct the following penalized MoEP (PMoEP) model:

$$\max_{\boldsymbol{\Theta}} \left\{ l_P^C(\boldsymbol{\Theta}) = l^C(\boldsymbol{\Theta}) - P(\boldsymbol{\pi}; \lambda) \right\}, \tag{10}$$

where

$$P(\boldsymbol{\pi}; \lambda) = n\lambda \sum_{k=1}^{K} D_k \log \frac{\epsilon + \pi_k}{\epsilon}, \tag{11}$$

with ϵ being a very small positive number ($\epsilon = 10^{-6}$ or $o(n^{-\frac{1}{2}} \log^{-1} n)$), λ being a tuning parameter ($\lambda > 0$), and D_k being the number of free parameters for the k^{th} component.

3.2. EM algorithm

In this subsection, we propose an EM algorithm to solve our PMoEP model (10). We assume that $\boldsymbol{\Theta}^{(t)} = \{\boldsymbol{\pi}^{(t)}, \boldsymbol{\eta}^{(t)}, \mathbf{U}^{(t)}, \mathbf{V}^{(t)}\}$ is the estimate at the t^{th} iteration.

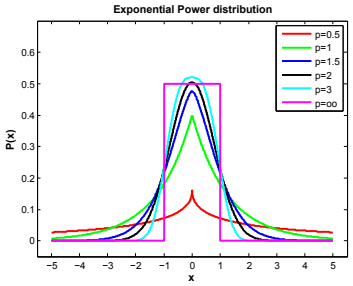

Figure 2. The probability density function of EP distributions.

In the E step, we compute the conditional expectation of z_{ijk} given e_{ij} by the Bayes' rule and get

$$\gamma_{ijk}^{(t+1)} = \frac{\pi_k^{(t)} f_{p_k}(y_{ij} - \mathbf{u}_i^{(t)}(\mathbf{v}_j^{(t)})^T)|0, \eta_k^{(t)})}{\sum_{l=1}^{K} \pi_l^{(t)} f_{p_l}(y_{ij} - \mathbf{u}_i^{(t)}(\mathbf{v}_j^{(t)})^T)|0, \eta_l^{(t)}))}. \quad (12)$$

Then, it is easy to construct the so-called Q function:

$$\begin{aligned} Q(\mathbf{\Theta}, \mathbf{\Theta}^{(t)}) = & \sum_{i,j\in\Omega} \sum_{k=1}^{K} \gamma_{ijk}^{(t+1)} [\log f_{p_k}(e_{ij}; 0, \eta_k) + \log \pi_k] \\ & - n\lambda \sum_{k=1}^{K} D_k \log \frac{\epsilon + \pi_k}{\epsilon}. \end{aligned} \quad (13)$$

In the M-step, we update $\mathbf{\Theta}$ by maximizing the Q function. To obtain the update for $\boldsymbol{\pi}$, we need to introduce a Lagrange multiplier τ to enforce the constraint $\sum_{k=1}^{K} \pi_k = 1$ and then maximize the following Lagrange function:

$$\begin{aligned} & \sum_{i,j\in\Omega} \sum_{k=1}^{K} \gamma_{ijk}^{(t+1)} \log \pi_k - n\lambda \sum_{k=1}^{K} D_k \log \frac{\epsilon + \pi_k}{\epsilon} \\ & + \tau(\sum_k \pi_k - 1). \end{aligned} \quad (14)$$

By taking the first derivative of (14) with respect to π_k and setting it to zero, we get

$$\pi_k^{(t+1)} = \max\left\{0, \frac{1}{1 - \lambda\hat{D}} \left[\frac{\sum_{i,j\in\Omega} \gamma_{ijk}^{(t+1)}}{|\Omega|} - \lambda D_k\right]\right\}, \quad (15)$$

where $\hat{D} = \sum_{k=1}^{K} D_k = 2K$. To obtain the update equation of $\boldsymbol{\eta}$, we take the first derivative of Q with respect to η_k, and find the zero point:

$$\eta_k^{(t+1)} = \frac{N_k}{p_k \sum_{i,j\in\Omega} \gamma_{ijk}^{(t+1)} |y_{ij} - \mathbf{u}_i^{(t)}(\mathbf{v}_j^{(t)})^T|^{p_k}}, \quad (16)$$

where $N_k = \sum_{i,j\in\Omega} \gamma_{ijk}^{(t+1)}$. To obtain an estimate for $\mathbf{U}, \mathbf{V}$, it is natural to maximize the following function:

$$\sum_{i,j\in\Omega} \sum_{k=1}^{K} \gamma_{ijk}^{(t+1)} \log f_k(y_{ij} - \mathbf{u}_i^{(t)}(\mathbf{v}_j^{(t)})^T; \eta_k^{(t+1)}). \quad (17)$$

Obviously, maximizing (17) is equivalent to solving [1]

$$\min_{\mathbf{U},\mathbf{V}} \sum_{k=1}^{K} ||\mathbf{W}_{(k)} \odot (\mathbf{Y} - \mathbf{U}\mathbf{V}^T)||_{p_k}^{p_k}, \quad (18)$$

where the element $w_{(k)ij}$ of $\mathbf{W}_{(k)} \in \mathcal{R}^{m\times n}$ is

$$w_{(k)ij} = \begin{cases} (\eta_k^{(t+1)} \gamma_{ijk}^{(t+1)})^{\frac{1}{p_k}}, & i,j \in \Omega \\ 0, & i,j \notin \Omega \end{cases}.$$

To solve (18), we resort to augmented Lagrange multipliers (ALM) technique. Let $\mathbf{L} = \mathbf{U}\mathbf{V}^T$, (18) is then equivalent to

$$\begin{aligned} & \min_{\mathbf{U},\mathbf{V}} \sum_{k=1}^{K} ||\mathbf{W}_{(k)} \odot (\mathbf{Y} - \mathbf{L})||_{p_k}^{p_k} \\ & s.t \quad \mathbf{L} = \mathbf{U}\mathbf{V}^T. \end{aligned} \quad (19)$$

Then the augmented Lagrangian function can be written as:

$$\begin{aligned} L(\mathbf{U}, \mathbf{V}, \mathbf{L}, \mathbf{Y}, \rho) = & \sum_{k=1}^{K} ||\mathbf{W}_{(k)} \odot (\mathbf{Y} - \mathbf{L})||_{p_k}^{p_k} + \\ & \langle \mathbf{\Lambda}, \mathbf{L} - \mathbf{U}\mathbf{V}^T \rangle + \frac{\rho}{2} ||\mathbf{L} - \mathbf{U}\mathbf{V}^T||_F^2, \end{aligned} \quad (20)$$

where $\mathbf{\Lambda} \in \mathcal{R}^{m\times n}$ is the Lagrangian multiplier and $\rho > 0$ is a penalty parameter. Now we shall alternatively optimize $\mathbf{U}, \mathbf{V}, \mathbf{L}, \mathbf{\Lambda}, \rho$. At the $(s+1)^{th}$ iteration, the optimization process is

$$\begin{cases} (\mathbf{U}^{(s+1)}, \mathbf{V}^{(s+1)}, \mathbf{L}^{(s+1)}) = \arg\min_{\mathbf{U},\mathbf{V},\mathbf{L}} L(\mathbf{U}, \mathbf{V}, \mathbf{L}, \mathbf{\Lambda}^{(s)}, \rho^{(s)}), \\ \mathbf{\Lambda}^{(s+1)} = \mathbf{\Lambda}^{(s)} + \rho^{(s)}(\mathbf{L}^{(s+1)} - \mathbf{U}^{(s+1)}(\mathbf{V}^{(s+1)})^T), \\ \rho^{(s+1)} = \alpha\rho^{(s)}, \end{cases}$$

where α is a positive value. The first subproblem can be optimized alternately as follows.

(1) *Update* $\mathbf{U}, \mathbf{V}$. The following subproblem needs to be solved:

$$\min_{\mathbf{U},\mathbf{V}} ||\mathbf{L}^{(s)} + \frac{1}{\rho^{(s)}} \mathbf{\Lambda}^{(s)} - \mathbf{U}\mathbf{V}^T||_F^2, \quad (21)$$

which can be efficiently solved by SVD method.

(2) *Update* $\mathbf{L}$. It is easy to see that we need to optimize a series of subproblems for $(i,j) \in \Omega$ with the following form:

$$\begin{aligned} & \min_{e_{ij}} \sum_k \eta_k \gamma_{ijk} |y_{ij} - l_{ij}|^{p_k} + \frac{\rho^{(s)}}{2} l_{ij}^2 \\ & + ((\mathbf{\Lambda}^{(s)})_{ij} - \rho^{(s)} \mathbf{u}_i \mathbf{v}_j^T) l_{ij}. \end{aligned} \quad (22)$$

Let $q_{ij} = y_{ij} - l_{ij}$, (22) is equivalent to

$$\begin{aligned} & \min_{q_{ij}} \frac{1}{2} \left[(-\mathbf{u}_i \mathbf{v}_j^T + y_{ij} + \frac{1}{\rho^{(s)}} (\mathbf{\Lambda}^{(s)})_{ij}) - q_{ij}\right]^2 \\ & + \frac{1}{\rho^{(s)}} \sum_l \eta_l \gamma_{ijl} |q_{ij}|^{p_l}, \end{aligned} \quad (23)$$

[1] The p-norm of a matrix is defined as $||X||_p = (\sum_{i,j} |x_{ij}|^p)^{\frac{1}{p}}$.

Let $w_{ij} = -\mathbf{u}_i\mathbf{v}_j^T + y_{ij} + \frac{1}{\rho^{(s)}}(\mathbf{\Lambda}^{(s)})_{ij}$. Then, we should solve $|\Omega|$ such subproblems:

$$\min_{q_{ij}} \frac{1}{2}(w_{ij} - q_{ij})^2 + \frac{1}{\rho}\sum_{l=1}^{K} \eta_l \gamma_{ijl} |q_{ij}|^{p_l}. \tag{24}$$

In order to solve (24), we take the first derivative with respect to s_{ij} and then adopt the well-known Newton method to find its zero point.

Remark: If f_k is specified as the density of a Gaussian distribution, the PMoEP model degenerates to the penalized MoG (PMoG) model. The optimization process of the PMoG model is almost the same as the PMoEP except minimizing (18). In this case, the optimization problem (18) becomes

$$\min_{\mathbf{U},\mathbf{V}} ||\mathbf{W} \odot (\mathbf{Y} - \mathbf{U}\mathbf{V}^T)||_2^2, \tag{25}$$

and then any off-the-shelf weighted L_2 norm LRMF method can be adopted to solve it.

We then summarize the EM algorithm in Algorithm 1.

Algorithm 1 EM Algorithm for PMoEP LRMF

Input:

Data $\mathbf{Y} = \{y_{ij}\}_{m\times n}$. Initialize $\mathbf{\Theta}^{(t)} = \{\boldsymbol{\pi}^{(t)}, \boldsymbol{\eta}^{(t)}, \mathbf{U}^{(t)}, \mathbf{V}^{(t)}\}$, the number of components K_{start}, preset candidates $\mathbf{p} = [p_1, \ldots, p_{K_{start}}]$, tolerance ϵ and $t = 0$.

Output:

Parameter $\hat{\mathbf{\Theta}}$, the number of final components $\hat{K}$ and posterior probability $\boldsymbol{\gamma} = \{\gamma_{ijk}\}_{m\times n\times \hat{K}}$.

1: **repeat**
2: (*E step*): Update the posterior probability $\boldsymbol{\gamma}^{(t)}$ by (12).
3: (*M step for* $\boldsymbol{\pi}$): Update parameter $\boldsymbol{\pi}^{(t)}$ by (15), and remove the component with $\pi_k^{(t)} = 0$.
4: (*M step for* $\boldsymbol{\eta}$): Update parameters $\boldsymbol{\eta}^{(t)}$ by (16).
5: (*M step for* $\mathbf{U}$,$\mathbf{V}$): Update $\mathbf{U}^{(t)}$, $\mathbf{V}^{(t)}$ by maximizing (18) using ALM strategy (19)-(24).
6: **until** *converge*;

The convergence property of standard EM algorithm for maximizing non-penalized log-likelihood function has been discussed in many literatures [18]. Here, we will show the convergence property of the EM algorithm for maximizing penalized log-likelihood function by the following theorem.

Theorem 1. *Let $l_P^G(\mathbf{\Theta}) = l(\mathbf{\Theta}) - P(\boldsymbol{\pi};\lambda)$, where $l(\Theta)$ is defined in (8). If we assume that $\{\mathbf{\Theta}^{(t)}\}$ is the sequence generated by Algorithm 1 and the sequence of likelihood values $\{l_P^G(\mathbf{\Theta}^{(t)})\}$ is bounded above, then there exits a constant $l^\star$ such that*

$$\lim_{t\to\infty} l_P^G(\mathbf{\Theta}^{(t)}) = l^\star, \tag{26}$$

where

$$\mathbf{\Theta}^{(t)} = \arg\max_{\mathbf{\Theta}} \left\{ \Omega(\mathbf{\Theta}|\mathbf{\Theta}^{(t-1)}) + P(\boldsymbol{\pi}^{(t-1)};\lambda) - P(\boldsymbol{\pi};\lambda) \right\}, \tag{27}$$

and

$$\Omega(\mathbf{\Theta}|\mathbf{\Theta}^{(t-1)}) = \sum_{\mathbf{Z}} \mathbb{P}(\mathbf{Z}|\mathbf{E};\mathbf{\Theta}^{(t-1)}) \log \frac{\mathbb{P}(\mathbf{E},\mathbf{Z};\mathbf{\Theta})}{\mathbb{P}(\mathbf{E},\mathbf{Z};\mathbf{\Theta}^{(t-1)})}. \tag{28}$$

The proof of Theorem 1 can be found in the supplementary materials. The theorem actually indicates that the penalized likelihod $l_P^G(\mathbf{\Theta}^{(t)})$ is not decreased after an EM iteration.

3.3. Selection of Tuning Parameter λ

So far, the tuning parameter λ is treated as fixed in Algorithm 1. In real applications, one needs to select a good one among many condidates. For standard LASSO and SCAD penalized regression, there are many ways to select a proper λ, such as cross-validation and Bayesian information criterion(BIC). Following [10] , we consider the modified BIC criteria defined as

$$\text{BIC}(\lambda) = \sum_{i,j\in\Omega} \log\{\sum_{k=1}^{\hat{K}} \hat{\pi}_k f_k(e_{ij};\hat{\eta}_k)\} - \frac{1}{2}(\sum_{k=1}^{\hat{K}} D_k)\log|\Omega|. \tag{29}$$

and then select $\hat{\lambda}$ by

$$\hat{\lambda} = \arg\max_{\lambda} \text{BIC}(\lambda), \tag{30}$$

where $|\Omega|$ is the number of the observations, $\hat{K}$ is the estimate of the number of components, $\hat{\pi}_k$ is the estimate of parameter π_k and $\hat{\eta}_k$ is the estimate of parameter η_k for maximizing (10) with a given λ.

4. Experiments

To evaluate the performance of the proposed PMoEP and its special case PMoG, we conducted a series of synthetic and real experiments including face modeling and hyperspectral image restoration. Several state-of-the-art methods, including MoG [19], CWM [20], Damped Wiberg (DW) [24], RegL1ALM [37] and Singular Value Decomposition (SVD) were considered for comparison. All experiments were implemented in Matlab R2012a on a PC with 3.40GHz CPU and 8GB RAM.

4.1. Synthetic experiments

Similar to [19, 36], several synthetic experiments were designed to compare the proposed methods with other popular ones under different noise settings. For each experiment, we first randomly generated 30 low rank matrices with size 40×20 and rank 4. Each of these matrices was generated by the multiplication of two low-rank matrices $\mathbf{U}_{gt} \in \mathcal{R}^{40\times 4}$ and $\mathbf{V}_{gt} \in \mathcal{R}^{20\times 4}$, i.e., $\mathbf{Y}_{gt} = \mathbf{U}_{gt}\mathbf{V}_{gt}^T$.

Then, we randomly specified 20% of entries in $\mathbf{Y}_{gt}$ as missing data and further added different types of noise in the non-missing entries as follows: (1) *Gaussian noise*: 80% of the entries were corrupted with $\mathcal{N}(0, 0.04)$. (2) *Sparse noise*: 10% of the entries were corrupted with uniformly distributed noise between [-20,20]. (3) *EP noise*: 80% of the entries were corrupted with $EP(0, 0.2^p p), p = 0.2$. (4) *Mixture noise*: 20% of the entries were corrupted with uniformly distributed noise between [-5,5], 20% are contaminated with Gaussian noise $\mathcal{N}(0, 0.04)$ and the remaining 40% are corrupted with Gaussian noise $\mathcal{N}(0, 0.01)$. We denote the noisy matrix as $\mathbf{Y}_{no}$. Six measures were utilized for performance assessment:

$$C1 = ||\mathbf{W} \odot (\mathbf{Y}_{no} - \tilde{\mathbf{U}}\tilde{\mathbf{V}}^T)||_1,\ C2 = ||\mathbf{W} \odot (\mathbf{Y}_{no} - \tilde{\mathbf{U}}\tilde{\mathbf{V}}^T)||_2,$$
$$C3 = ||\mathbf{Y}_{gt} - \tilde{\mathbf{U}}\tilde{\mathbf{V}}^T||_1,\ C4 = ||\mathbf{Y}_{gt} - \tilde{\mathbf{U}}\tilde{\mathbf{V}}^T||_2,$$
$$C5 = subspace(\mathbf{U}_{gt}, \tilde{\mathbf{U}}),\ C6 = subspace(\mathbf{V}_{gt}, \tilde{\mathbf{V}}),$$

where $\tilde{\mathbf{U}}, \tilde{\mathbf{V}}$ are the outputs of the corresponding competing method, and $subspace(\mathbf{U}_1,\mathbf{U}_2)$ denotes the angle between subspaces spanned by the columns of $\mathbf{U}_1$ and $\mathbf{U}_2$. Note that $C1$ and $C2$ are the optimization objective function for L_1 and L_2 norm LRMF problems, while the latter four measures ($C3 - C6$) are more faithful to evaluate whether the method recovers the correct subspaces.

In this set of experiments, to alleviate the local optimum issue, we adopt the multiple random initialization strategy. More specifically, for MoG, DW, CWM and RegL1ALM methods, we first run with 20 random initializations for each generated matrice and then select the best result with respect to the objective value. For PMoG and PMoEP, the experimental setting is almost the same with the other competing methods except that for each random initialization, we first provide a series of λs and then select the one with the largest BIC as the result for this initialization. Finally, we select the initialization with the largest likelihood value as the result for this generated matrice. The performance of each method on each simulation was evaluated as the average results over the 30 random generated matrices in terms of the six measures, and the results are summarized in Table 1. Additionally, the average time of one run of each method in all cases are also summarized in the table (For PMoEP and PMoG, one run means λ is given).

For all the methods, we set the rank of the low-rank component to 4 and adopt the random initialization strategy for $\mathbf{U}$ and $\mathbf{V}$. Particularly, for PMoG and PMoEP, two parameters K_{start} and λ need suitable initializations. From the extensive off-line experiments, we indeed find that the K_{start} should not be set too large (usually from 4 to 10) and λ is uniformly selected from [0,0.3]. In addition, once K_{start} is initialized for PMoEP, the length of parameter vector $\mathbf{p} = [p_1, p_2, \ldots, p_{K_{start}}]$ in PMoEP is determined. In these synthetic experiments, each element p_k is selected from 0.1 to 2.

	PMoEP	PMoG	MoG [19]	CWM [20]	DW [24]	RegL1ALM [37]
Sparse Noise						
C1	8.32e+2	7.98e+2	8.03e+2	**7.97e+2**	1.05e+3	8.63e+2
C2	1.07e+4	1.06e+4	1.08e+4	9.97e+3	**4.79e+3**	5.96e+3
C3	3.23e+1	**2.13e-12**	1.57e+1	4.10e+2	5.62e+14	1.19e+6
C4	6.73e+1	**2.42e-5**	3.45e+1	9.69e+1	1.30e+7	4.26e+3
C5	1.40e-1	**3.85e-8**	3.30e-2	3.93e-1	1.47e+1	1.49e+1
C6	6.16e-2	**2.33e-8**	1.96e-6	1.13e-1	1.44e+1	1.45e+1
Gaussian Noise						
C1	8.08e+1	7.83e+1	8.20e+1	7.71e+1	8.16e+1	**7.35e+1**
C2	**1.65e+1**	2.41e+1	1.67e+1	2.17e+1	1.67e+1	2.14e+1
C3	**1.31e+1**	2.48e+1	1.33e+1	2.24e+1	1.32e+1	2.03e+1
C4	7.89e+1	1.06e+2	7.90e+1	9.95e+1	**7.86e+1**	9.74e+1
C5	**8.54e-2**	1.17e-1	9.24e-2	1.22e-1	8.67e-2	1.10e-1
C6	5.82e-2	8.43e-2	6.30e-2	9.96e-2	**5.77e-2**	7.06e-2
EP Noise						
C1	3.64e+2	3.19e+2	**3.18e+2**	3.30e+2	4.33e+2	3.53e+2
C2	1.29e+3	1.25e+3	1.02e+3	1.35e+3	**6.49e+2**	8.75e+2
C3	**1.58e+2**	3.28e+3	6.98e+4	1.99e+2	1.04e+7	6.66e+4
C4	**2.40e+2**	2.65e+2	3.52e+2	2.47e+2	2.57e+3	8.22e+2
C5	**3.02e-1**	4.21e-1	4.19e-1	3.62e-1	1.08e+1	1.14e+1
C6	**2.02e-1**	2.87e-1	3.28e-1	2.53e-1	9.05e-1	1.01e+1
Mixture Noise						
C1	4.40e+2	4.38e+2	4.27e+2	4.31e+2	5.18e+2	**4.26e+2**
C2	1.27e+3	1.33e+3	1.28e+3	1.10e+3	**8.26e+2**	1.12e+3
C3	**1.42e+2**	1.90e+2	1.88e+2	3.77e+2	2.16e+8	1.45e+4
C4	**1.68e+2**	1.91e+2	1.84e+2	3.15e+2	7.08e+3	5.08e+2
C5	**3.47e-1**	3.90e-1	3.93e-1	5.41e-1	8.44e-1	7.96e-1
C6	1.79e-1	1.78e-1	**1.62e-1**	3.91e-1	7.46e-1	6.17e-1
Time(s)	2.34	0.28	0.46	**0.062**	0.352	0.11

Table 1. Performance evaluation on synthetic data. The best results in terms of the six criteria are highlighted in bold.

It is easy to observe from Table 1 that, the Damped Wiberg method, which is a L_2-norm-based method, performs best or second best in the Gaussian noise case among all the competing methods. The MoG method performs well across all kinds of noisy cases although it does not always achieve the best performance. Additionally, the PMoG method can not only overcome the drawback of the selection of the number of components of MoG method, but also obtain almost the same performance as the MoG method under properly set parameters. Specifically, the PMoG performs best in the sparse noise case. In the mixture noise cases, the proposed PMoEP method has almost the best or second best performance in estimating a better subspace from the noisy data. In particular, when the noise obeys the EP distribution, this method always performs best in term of criteria C3-C6.

The promising performance of our proposed PMoEP method in these cases can be easily explained by Figure 3, which compares the ground truth noises and the estimated ones by the PMoEP method. It can be easily observed that the estimated noise distributions well match the true ones.

4.2. Real data experiments

In this subsection, we use our PMoG and PMoEP methods in two real applications with complex noise, namely, face modeling and hyperspectral image restoration. The competing methods include MoG [19], CWM [20], DW [24], RegL1ALM [37] and SVD.

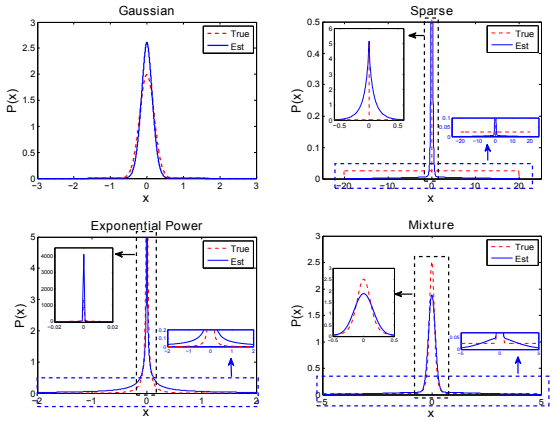

Figure 3. Visual comparison of the ground truth (denote by True) noise probability density functions and those estimated (denote by Est) by the PMoEP method in the synthetic experiments. The embedded sub-figures depict the zoom-in of the indicated portions.

4.2.1 Face modeling experiments

This experiment aims to test the effectiveness of the PMoG and PMoEP methods in face modeling applications. We choose the first and the second subset of the Extended Yale B database[2], and each subset consists of 64 faces of one person. The size of one face image is 192×168 and thus the data matrices are with size 32256×64. Typical images used for comparison are plotted in the first column of Figure 5.

We set the rank to 4 [19, 36] and adopt two initialization strategies, random and SVD, for all competing methods. Then we report the best result among the initializations in terms of objective value. Addtionally, for PMoG and PMoEP methods, we initialize K_{start} and provide a series of λ. Then, we use the modified BIC criterion to select the λ corresponding the largest BIC value. Specifically, for PMoG, K_{start} is set to 6 and λ is selected from $\{0.01, 0.05, 0.12, 0.15, 0.18\}$; while for PMoEP, K_{start} is set to 4, the corresponding $\mathbf{p} = [p_1, p_2, p_3, p_4]$ are set to $[0.2, 0.3, 0.4, 0.5]$ and λ is chosen from $\{0.0001, 0.005, 0.01, 0.05, 0.12, 0.24\}$. Figure 4 (a) shows the choice of λ by the BIC for PMoEP. Figure 5 displays the reconstructed faces of all the methods.

From Figure 5, it is easy to observe that, the proposed PMoEP and PMoG methods, as well as the other competing ones, can remove the cast shadows and saturations in faces. However, our PMoEP method performs better than other ones on faces with a large dark region. The reason can be seen in Figure 6. Comparing with the PMoG and the MoG methods, our PMoEP method is capable of modeling more complex noise, as is shown in Figure 6. Specifically, our proposed PMoEP method is more capable of extracting significant cast shadow and saturation noises than the

[2]http://vision.ucsd.edu/ leekc/ExtYaleDatabase/ExtYaleB.html

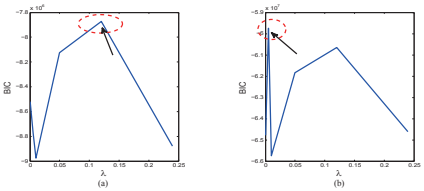

Figure 4. Using the Modified BIC to select the tuning parameter λ of the PMoEP method. (a) Face modeling. (b) Hyperspectral image restoration.

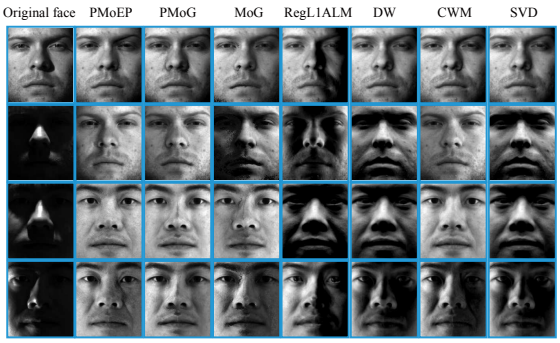

Figure 5. From left to right: original face images, reconstructed faces by PMoEP, PMoG, MoG, RegL1ALM, Damped Wiberg(DW), CWM and SVD.

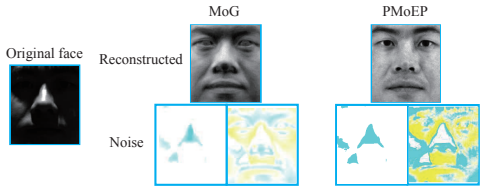

Figure 6. From left to right: original faces, reconstructed faces and extracted noise by MoG and PMoEP. The noise with positive and negative values are depicted in yellow and blue.

MoG method. Therefore, our PMoEP method leads to the best face reconstruction performance among all comparison methods.

4.2.2 Hyperspectral Image Restoration experiments

A Hyperspectral Image (HSI) dataset called Urban[3] is used in experiment. This dataset contains 210 bands, each of which is of size 307×307, and some bands are seriously polluted by atmosphere and water or corrupted by the mixture of sparse noise (stripes and deadlines) and Gaussian noise [34], as shown in Figure 1. We reshape each band as a vector, and stack all the vectors into a matrix, resulting in the final data matrix with size 94249×210. All the methods were implemented, except for the Damped Wiberg method which encounters the "out of memory" problem.

In this experiment, we set the rank to 4 and use

[3]http://www.tec.army.mil/hypercube.

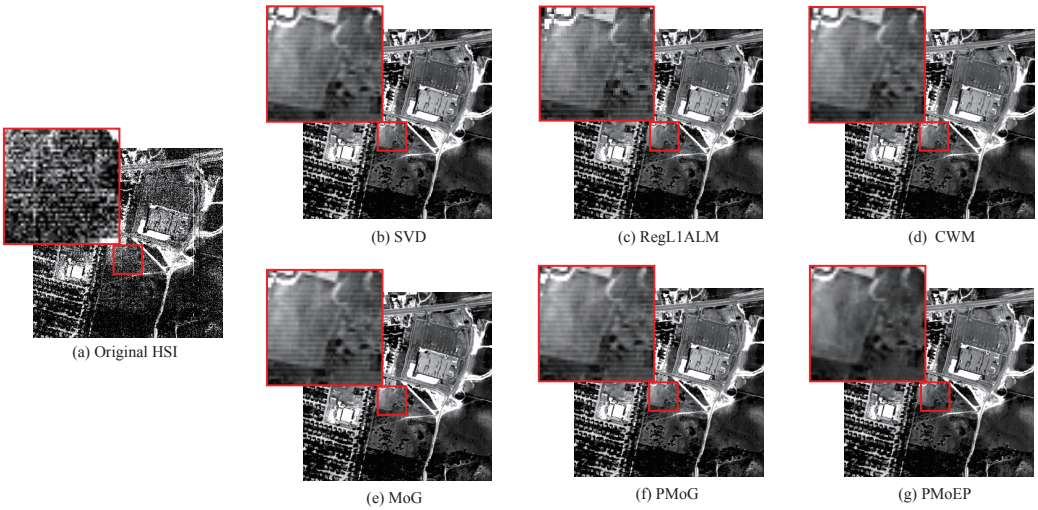

Figure 7. (a) original band (206). (b)-(g) reconstructed bands by SVD, RegL1ALM, CWM, MoG, PMoG and PMoEP.

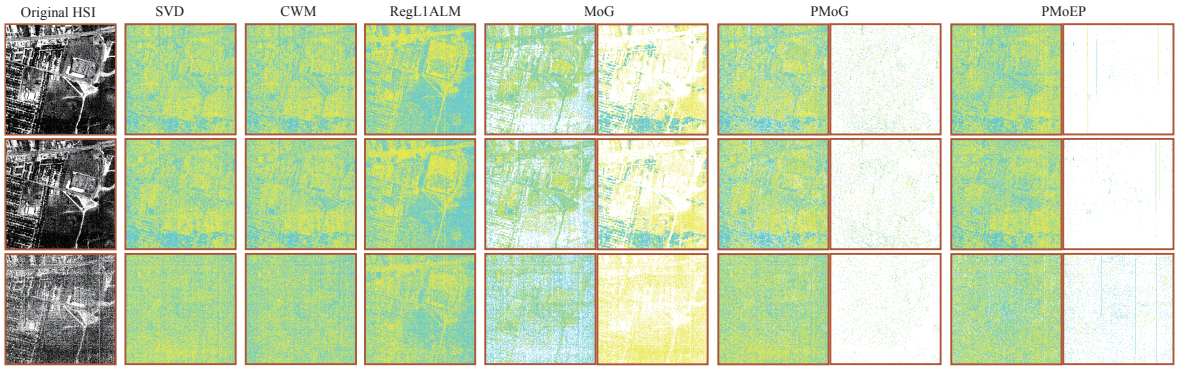

Figure 8. From left to right: original bands, and extracted noise by SVD, CWM, RegL1ALM, MoG, PMoG and PMoEP. The noises with positive and negative values are depicted in yellow and blue, respectively. This figure should be viewed in color and the details are better seen by zooming on a computer screen.

SVD as the initialization method for $\mathbf{U}$ and $\mathbf{V}$. For P-MoEP method, K_{start} is set to 5, the corresponding $\mathbf{p} = [p_1, p_2, p_3, p_4, p_5]$ are set to $[0.2, 0.5, 1, 1.8, 2]$ and λ is selected from $\{0.0001, 0.005, 0.01, 0.05, 0.12, 0.24\}$. Similarly, we utilize the modified BIC method to select the best λ corresponding to the largest BIC value. Figure 4 (b) shows the choice of λ by the BIC.

Our experimental results show that our method can generally improve the image quality of these HSIs. To show more details, an example image located at a band of this HSI is illustrated in Figure 7. An area of interest is amplified in the restored image obtained by all competing methods for easy comparison. It can be easily seen that the image restored by our PMoEP method better removes the noise, especially the strips noise, while the results obtained by other competing ones contain evident stripe noise area. Moreover, our PMoEP method separates the noise information more accurately. Specifically, PMoEP can properly discover the two types of noise under this kind of HSI data: stripes noise and Gaussian noise, while the other competing methods fail to do so, as is shown in Figure 8.

5. Conclusion

In this paper, we model the noise of the LRMF problem as MoEP distributions and propose the penalized MoEP (P-MoEP) model by applying the penalized likelihood method to MoEP distributions. Compared with the current LRMF methods, our PMoEP method can better fit data noise in a wide variety of synthetic and real complex noise scenarios, including face modeling and hyperspectral image restoration. Additionally, our method is capable of automatically learning the number of components from data, and thus facilitates to dealing with more complex applications. In future, we will try to extend the PMoEP model to high-order low rank tensor factorization problems and other practical applications.

Acknowledgements. This research was supported by 973 Program of China with No.3202013CB329404, the NSFC projects with No.11131006, 91330204, 61373114 and 11501440.

References

[1] P. M. Aguiar, J. Xavier, and M. Stosic. Spectrally optimal factorization of incomplete matrices. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[2] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos. Sparse bayesian methods for low-rank matrix estimation. *IEEE Transactions on Signal Processing*, 60(8):3964–3977, 2012.

[3] A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 316–322, 2005.

[4] P. Chen, N. Wang, N. L. Zhang, and D.-Y. Yeung. Bayesian adaptive matrix factorization with automatic model selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1284–1292, 2015.

[5] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[6] F. De La Torre and M. J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

[7] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JAsIs*, 41(6):391–407, 1990.

[8] A. Eriksson and A. Van Den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l 1 norm. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 771–778, 2010.

[9] K. R. Gabriel and S. Zamir. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21(4):489–498, 1979.

[10] T. Huang, H. Peng, and K. Zhang. Model selection for gaussian mixture models. *arXiv preprint arXiv:1301.3558*, 2013.

[11] H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust video denoising using low rank matrix completion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1791–1798, 2010.

[12] Q. Ke and T. Kanade. A subspace approach to layer extraction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–255, 2001.

[13] Q. Ke and T. Kanade. Robust l 1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 739–746, 2005.

[14] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[15] N. Kwak. Principal component analysis based on l1-norm maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1672–1680, 2008.

[16] B. Lakshminarayanan, G. Bouchard, and C. Archambeau. Robust bayesian matrix factorisation. In *International Conference on Artificial Intelligence and Statistics*, pages 425–433, 2011.

[17] V. Maz'ya and G. Schmidt. On approximate approximations using gaussian kernels. *IMA Journal of Numerical Analysis*, 16(1):13–29, 1996.

[18] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[19] D. Meng and F. D. L. Torre. Robust matrix factorization with unknown noise. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1337–1344, 2013.

[20] D. Meng, Z. Xu, L. Zhang, and J. Zhao. A cyclic weighted median method for l1 low-rank matrix factorization with missing entries. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[21] A. M. Mineo, M. Ruggieri, et al. A software tool for the exponential power distribution: The normalp package. *Journal of Statistical Software*, 12(4):1–24, 2005.

[22] K. Mitra, S. Sheorey, and R. Chellappa. Large-scale matrix factorization with missing data under additional constraints. In *Advances in Neural Information Processing Systems*, pages 1651–1659, 2010.

[23] T. Okatani and K. Deguchi. On the wiberg algorithm for matrix factorization in the presence of missing components. *International Journal of Computer Vision*, 72(3):329–337, 2007.

[24] T. Okatani, T. Yoshida, and K. Deguchi. Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 842–849, 2011.

[25] X. Shu, F. Porikli, and N. Ahuja. Robust orthonormal subspace learning: Efficient recovery of corrupted low-rank matrices. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3874–3881, 2014.

[26] N. Srebro, T. Jaakkola, et al. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning*, volume 3, pages 720–727, 2003.

[27] P. Sturm. Algorithms for plane-based pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 706–711, 2000.

[28] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[29] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[30] N. Wang, T. Yao, J. Wang, and D.-Y. Yeung. A probabilistic approach to robust matrix factorization. In *Proceedings of European Conference on Computer Vision*, pages 126–139, 2012.

[31] N. Wang and D.-Y. Yeung. Bayesian robust matrix factorization for image and video processing. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1785–1792, 2013.

[32] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.

[33] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

[34] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan. Hyperspectral image restoration using low-rank matrix recovery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(8):4729–4743, 2014.

[35] K. Zhao and Z. Zhang. Successively alternate least square for low-rank matrix factorization with bounded missing data. *Computer Vision and Image Understanding*, 114(10):1084–1096, 2010.

[36] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and L. Zhang. Robust principal component analysis with complex noise. In *Proceedings of the 31st International Conference on Machine Learning*, pages 55–63, 2014.

[37] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi. Practical low-rank matrix approximation under robust l 1-norm. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1417, 2012.