Low-rank Matrix Factorization under General Mixture Noise Distributions

Xiangyong Cao, Qian Zhao, Deyu Meng*, *Member, IEEE,* Yang Chen, Zongben Xu. .

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 problems using L_1-norm and L_2-norm losses, which mainly deal with Laplacian and Gaussian noises, 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 (PMoEP) 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. Moreover, by facilitating the local continuity of noise components, we embed Markov random field into the PMoEP model and further propose the advanced PMoEP-MRF model. An Expectation Maximization (EM) algorithm and a variational EM (VEM) algorithm are also designed to infer the parameters involved in the proposed PMoEP and the PMoEP-MRF model, respectively. The superseniority of our methods is demonstrated by extensive experiments on synthetic data, face modeling, hyperspectral image restoration and background subtraction.

Index Terms—Low-rank matrix factorization, mixture of exponential power distributions, Expectation Maximization algorithm, face modeling, hyperspectral image restoration, background subtraction.

I. 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 [37], face recognition [42], collaborative filtering [18], information retrieval [11], social networks [8], object recognition [38], layer extraction [16] and plane-based pose estimation [36].

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

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

Xiangyong Cao, Qian Zhao, Deyu Meng, Yang Chen, and Zongben Xu are with the School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, China (caoxiangyong45@gmail.com, timmy.zhaoqian@gmail.com, dymeng@mail.xjtu.edu.cn, chengyang9103@gmail.com, zbxu@mail.xjtu.edu.cn)
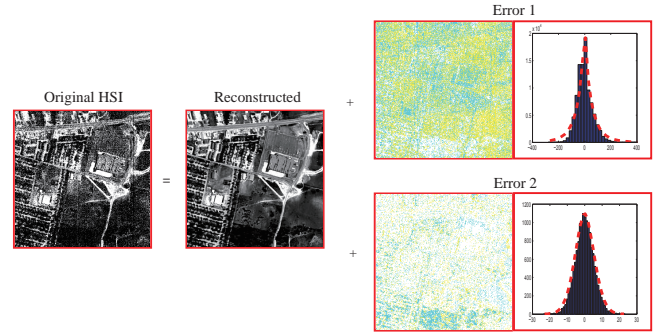*Deyu Meng is the corresponding author

Fig. 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).

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 noise, it is natural to utilize the L_2-norm (Frobenius norm) as the noise measure, which has been extensively studied in LRMF literatures [1], [5], [28], [30], [31], [35], [41], [44]. However, it has been recognized in many real applications that these methods constructed on L_2 norm are sensitive to outliers and non-Gaussian noise. In order to introduce robustness, the L_1-norm based models [12], [15], [17], [19], [34], [46] have attracted much attention recently. 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 hyperspectral image (HSI) as an example, it has been investigated in [43] that there are mainly two kinds of noise embedded in such type of data, i.e., sparse noise (stripe and deadline) and Gaussian-like noise, as depicted in Fig. 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 located on edges and in texture areas of an image. The deadline noise, which is induced by some damaged sensor, results in zero or very small pixel values of entire columns of images along some HSI bands. The Gaussian-like noise is induced by some random disturbance during the transmission process of hyperspectral 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 that the noise follows a more complicated mixture of Gaussians (MoG) [25], which is expected to better fit real noise, since the MoG constructs a universal approximator to any continuous density function in theory [24]. However, this method still cannot finely adapt real data noise. On one hand, MoG can approximate a complex distribution, e.g. Laplace, only under the assumption that the number of components goes to infinity, while in applications only a finite number of components can be specified. On the other hand, it also lacks a theoretically sound manner to properly select the number of Gaussian mixture components based on the practical noise extent mixed in data. Thus, it is crucial to construct a better strategy with more adaptive distribution modeling capability on data noises beyond MoG.

In this paper, we propose a new LRMF method with a more general noise model to address the aforementioned issues. Specifically, we encode the noise as a mixture distribution of a series of sub- and super-Gaussians (i.e., general exponential power (EP) distribution), and formulate LRMF as a penalized MLE model, called PMoEP model [6]. Moreover, by facilitating the local continuity of noise components, we embed Markov random field into the PMoEP model and propose the PMoEP-MRF model. Then we design an Expectation Maximization (EM) algorithm and a variational EM (VEM) algorithm to estimate the parameters involved in the proposed PMoEP model and PMoEP-MRF model, respectively, and prove their convergence. The two new methods are not only capable of adaptively fitting complex real noise by EP noise components with proper parameters, but also able to automatically learn the proper 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. In Section II, the related work regarding LRMF is discussed. In Section III, we first present the PMoEP model and the corresponding EM algorithm, and then conduct the convergence analysis of the proposed algorithm. The PMoEP-MRF model and the corresponding variational EM algorithm are proposed in Section IV. In Section V, extensive experiments are conducted to substantiate the superiority of the proposed models over previous methods. Finally, conclusions are drawn in Section VI. Throughout the paper, we denote scalars, vectors, and matrices as the non-bold letters, bold lower case letters, and bold upper case letters, respectively.

II. RELATED WORK

The L_2 norm LRMF with missing data has been studied for decades. Gabriel and Zamir [13] proposed a weighted SVD method as the early attempt for this task. They used alternated minimization to find the principal subspace underlying the data. Srebro and Jaakkola [35] proposed the Weighted Low-rank Approximation (WLRA) algorithm to enhance efficiency of LRMF calculation. Buchanan and Fitzgibbon [5] further proposed a regularized model that adds a regularization term

and then adopts the damped newton 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 [30] 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 $\mathbf{U}$ via least squares while updates $\mathbf{V}$ 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. [31]. 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 [44] formulated the L_2- norm LRMF as a constrained model to improve its stability in real applications. Wen et al. [41] adopted the alternating strategy to solve the L_2-norm LRMF problem. Mitra et al. [28] proposed an augmented Lagrangian method to solve the L_2-norm 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 [10] 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 along this research line. Ke and Kanade [17] initially replaced the L_2-norm with the L_1-norm for LRMF, and then solved the optimization by alternated convex programming (ACP) method. Kwak [19] 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 [12] 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. [46] 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. [39] proposed probabilistic robust matrix factorization (PRMF) that modeled the noise as a Laplace distribution, which has been later extended to fully Bayesian settings by Wang and Yeung [40]. 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 recently to make the model adaptable to more complex noise scenarios. Lakshminarayanan et al. [20] 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 [25] modeled the noise as a MoG distribution for LRMF, and later was extended to the Bayesian framework by Chen et al. [7] and to RPCA by Zhao et al. [45]. Although better than traditional methods, these methods are still very limited in dealing with complex noise in real scenarios.

III. LRMF WITH MoEP NOISE

In this section, we first present the new LRMF model with MoEP noise, called PMoEP model, and then design an EM algorithm to solve it. Finally, we give the convergence analysis of the proposed EM algorithm and the implementation issues.

A. 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 $\mathbf{v}_i$ represent the ith row vectors of $\mathbf{U}$ and $\mathbf{V}$, respectively, and e_{ij} is the noise embedded in y_{ij}. Instead of assuming that the noise obeys Gaussian [35], Laplace [17] or MoG [25] distributions as previous methods, we assume that the noise e_{ij} follows more flexible 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 kth 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 [27], 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 Fig. 2). Therefore, all previous cases including L_2, L_1, MoG and any combinations of them are just special cases of MoEP. By setting $\eta = 1/(p\sigma^p)$, the EP distribution (4) can be equivalently written as $EP_p(e; 0, p\sigma^p)$.

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 kth 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 likeli-*
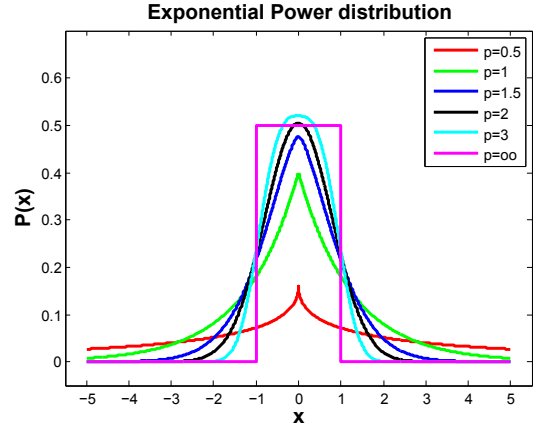

Fig. 2. The probability density function of EP distributions.

hood function can then 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. Thus, various model selection techniques can be readily employed to resolve this issue. Most conventional methods are based on the likelihood function and some information theoretic criteria, such as AIC and BIC. However, Leroux [21] showed that these criteria may overestimate the true number of components. On the other hand, Bayesian approaches [32], [47] have also been used to find a suitable number of components of the finite mixture model. But the computation burden and statistical properties of the Bayesian method limit its use to a certain extent. Here we adopt a recently proposed method by Huang et al. [14] 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, λ being a tuning parameter ($\lambda > 0$), and D_k being the number of free parameters for the kth component. In the proposed PMoEP model, D_k equals 2 (for π_k and η_k).

B. EM algorithm for PMoEP model

In this subsection, we propose an EM algorithm to solve the proposed PMoEP model (10). The EM algorithm is an iterative procedure and thus we assume that $\Theta^{(t)} = \{\{\boldsymbol{\pi}^{(t)}\}, \{\boldsymbol{\eta}^{(t)}\}, \mathbf{U}^{(t)}, \mathbf{V}^{(t)}\}$ is the estimation at the tth iteration. In the following, we will introduce the two steps of the proposed EM algorithm.

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

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

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

$$Q(\Theta, \Theta^{(t)}) = \sum_{i,j \in \Omega, k} \gamma_{ijk}^{(t+1)} [\log f_{p_k}(y_{ij} - \mathbf{u}_i \mathbf{v}_j^T; \eta_k) + \log \pi_k]$$
$$- n\lambda \sum_{k=1}^K D_k \log \frac{\epsilon + \pi_k}{\epsilon}.$$

In the M-step, we update Θ by maximizing the Q function. For $\boldsymbol{\pi}$ and $\boldsymbol{\eta}$, it is easy to obtain the update equations by taking the first derivative of Q with respect to them respectively, and finding the zero points through:

$$\pi_k^{(t+1)} = \max\left\{0, \frac{1}{1 - \lambda\hat{D}}\left[\frac{\sum_{i,j \in \Omega} \gamma_{ijk}^{(t+1)}}{|\Omega|} - \lambda D_k\right]\right\}, \quad (13)$$

$$\eta_k^{(t+1)} = \frac{N_k}{p_k \sum_{i,j \in \Omega} \gamma_{ijk}^{(t+1)} |y_{ij} - \mathbf{u}_i^{(t)}(\mathbf{v}_j^{(t)})^T|^{p_k}}, \quad (14)$$

where $\hat{D} = \sum_{k=1}^K D_k = 2K$, $N_k = \sum_{i,j \in \Omega} \gamma_{ijk}^{(t+1)}$ and $|\Omega|$ is the number of non-missing elements. To update $\mathbf{U}, \mathbf{V}$, we need to maximize the following function:

$$-\sum_{i,j \in \Omega} \sum_{k=1}^K \gamma_{ijk}^{(t+1)} \eta_k^{(t+1)} |y_{ij} - \mathbf{u}_i^{(t)}(\mathbf{v}_j^{(t)})^T|^{p_k}, \quad (15)$$

which is equivalent to solving[1]

$$\min_{\mathbf{U}, \mathbf{V}} \sum_{k=1}^K ||\mathbf{W}_{(k)} \odot (\mathbf{Y} - \mathbf{U}\mathbf{V}^T)||_{p_k}^{p_k}, \quad (16)$$

where the element $w_{(k)ij}$ of $\mathbf{W}_{(k)} \in \mathcal{R}^{m \times n}(k = 1, \ldots, K)$ 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 (16), we resort to augmented Lagrange multipliers (ALM) method. By introducing auxiliary variable $\mathbf{L} = \mathbf{U}\mathbf{V}^T$, (16) can be equivalently rewritten as

$$\min_{\mathbf{U}, \mathbf{V}} \sum_{k=1}^K ||\mathbf{W}_{(k)} \odot (\mathbf{Y} - \mathbf{L})||_{p_k}^{p_k}, \quad s.t \ \mathbf{L} = \mathbf{U}\mathbf{V}^T. \quad (17)$$

The augmented Lagrangian function can be written as:

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

$$L(\mathbf{U}, \mathbf{V}, \mathbf{L}, \mathbf{Y}, \rho) = \sum_{k=1}^K ||\mathbf{W}_{(k)} \odot (\mathbf{Y} - \mathbf{L})||_{p_k}^{p_k}$$
$$+ \langle \boldsymbol{\Lambda}, \mathbf{L} - \mathbf{U}\mathbf{V}^T \rangle + \frac{\rho}{2} ||\mathbf{L} - \mathbf{U}\mathbf{V}^T||_F^2, \quad (18)$$

where $\boldsymbol{\Lambda} \in \mathcal{R}^{m \times n}$ is the Lagrange multiplier and ρ is a positive scalar. Then the optimization (17) can be solved by alternatively updating all involved variables and multipliers as follows

$$\begin{cases} (\mathbf{U}^{(s+1)}, \mathbf{V}^{(s+1)}) = \underset{\mathbf{U}, \mathbf{V}}{\arg\min} L(\mathbf{U}, \mathbf{V}, \mathbf{L}^{(s)}, \boldsymbol{\Lambda}^{(s)}, \rho^{(s)}), \\ \mathbf{L}^{(s+1)} = \underset{\mathbf{L}}{\arg\min} L(\mathbf{U}^{(s+1)}, \mathbf{V}^{(s+1)}, \mathbf{L}, \boldsymbol{\Lambda}^{(s)}, \rho^{(s)}), \\ \boldsymbol{\Lambda}^{(s+1)} = \boldsymbol{\Lambda}^{(s)} + \rho^{(s)}(\mathbf{L}^{(s+1)} - \mathbf{U}^{(s+1)}(\mathbf{V}^{(s+1)})^T), \\ \rho^{(s+1)} = \alpha\rho^{(s)}, \end{cases} \quad (19)$$

where α is a preset constant which is slightly larger than 1, guaranteeing the gradually increasing value for ρ in each iteration. Now we discuss how to solve the subproblems involved in the above procedure.

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

$$\min_{\mathbf{U}, \mathbf{V}} ||\mathbf{L}^{(s)} + \frac{1}{\rho^{(s)}} \boldsymbol{\Lambda}^{(s)} - \mathbf{U}\mathbf{V}^T||_F^2, \quad (20)$$

which can be accurately and efficiently solved by the SVD method.

(2) *Update* $\mathbf{L}$. We need to solve the following problem:

$$\min_{\mathbf{L}} \sum_{k=1}^K ||\mathbf{W}_{(k)} \odot (\mathbf{Y} - \mathbf{L})||_{p_k}^{p_k} + \langle \boldsymbol{\Lambda}^{(s)}, \mathbf{L} \rangle$$
$$+ \frac{\rho^{(s)}}{2} ||\mathbf{L} - \mathbf{U}^{(s+1)}(\mathbf{V}^{(s+1)})^T||_F^2. \quad (21)$$

This problem seems to be more difficult due to its non-convexity and non-smoothness. However, we can divide it into mn independent scalar optimization problems as follows:

$$\begin{cases} \min_{l_{ij}} \sum_k \eta_k \gamma_{ijk} |y_{ij} - l_{ij}|^{p_k} + \frac{\rho^{(s)}}{2} l_{ij}^2 \\ \qquad + ((\boldsymbol{\Lambda}_{ij}^{(s)}) - \rho^{(s)} \mathbf{u}_i \mathbf{v}_j^T) l_{ij}, \quad (i,j) \in \Omega \\ \min_{l_{ij}} \frac{\rho^{(s)}}{2} l_{ij}^2 + ((\boldsymbol{\Lambda}^{(s)})_{ij} - \rho^{(s)} \mathbf{u}_i \mathbf{v}_j^T) l_{ij}. \quad (i,j) \notin \Omega \end{cases} \quad (22)$$

Letting $s_{ij} = y_{ij} - l_{ij}$, (22) is equivalent to

$$\begin{cases} \min_{s_{ij}} \frac{1}{2}(t_{ij} - s_{ij})^2 + \frac{1}{\rho^{(s)}} \sum_l \eta_l \gamma_{ijl} |s_{ij}|^{p_l}, \ (i,j) \in \Omega \\ \min_{s_{ij}} \frac{1}{2}(t_{ij} - s_{ij})^2, \ (i,j) \notin \Omega \end{cases} \quad (23)$$

where $t_{ij} = -\mathbf{u}_i \mathbf{v}_j^T + y_{ij} + \frac{1}{\rho^{(s)}}(\boldsymbol{\Lambda}_{ij}^{(s)})$. Then, for each $(i,j) \in \Omega$, (23) is equivalent to the following subproblem:

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

This problem requires to optimize a scalar variable, and we take its first derivative with respect to s_{ij} and then adopt the well-known Newton method to easily approach a local minimum of it. The procedure of updating $\mathbf{L}$ by ALM method can then be listed in Algorithm 1.

Algorithm 1 ALM method for solving (16).

Input: The initialization of $\mathbf{L}^{(0)}$, $\mathbf{\Lambda}^{(0)}$ and $s=0$.
Output: $\mathbf{U}$ and $\mathbf{V}$.
 1: **while** not converged **do**
 2:    Updating $\mathbf{U}^{(s+1)}$ and $\mathbf{V}^{(s+1)}$ via Eq. (20);
 3:    Updating $\mathbf{L}^{(s+1)}$ via Eqs. (23) and (24).
 4:    Updating $\mathbf{\Lambda}^{(s+1)}$ via Eq. (19).
 5:    Updating $\alpha^{(s+1)}$ via Eq. (19).
 6: **end while**

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 the minimization form of (16). In this case, the optimization problem (18) has the following form

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

and then any off-the-shelf weighted L_2 norm LRMF method can be adopted to solve it. It should be noted that the PMoG method so conducted is different from the previous MoG method [25] due to its augmented automatic mixture-component-number learning capability.

The proposed EM algorithm for PMoEP model can now be summarized in Algorithm 2.

Algorithm 2 EM Algorithm for PMoEP LRMF.

Input: Data $\mathbf{Y}$; The algorithm parameters: rank r and λ.
Output: Parameter $\mathbf{\Theta}$, the number of mixture components K_{final} and posterior probability $\gamma = (\gamma_{ijk})_{m \times n \times K_{final}}$.
Initialization: $\mathbf{\Theta}^{(t)} = \{\boldsymbol{\pi}^{(t)}, \boldsymbol{\eta}^{(t)}, \mathbf{U}^{(t)}, \mathbf{V}^{(t)}\}$, the number of initial mixture components K_{start}, preset candidates $\mathbf{p} = [p_1, \ldots, p_{K_{start}}]$, tolerance ϵ and $t=0$.
 1: **while** not converged **do**
 2:    Updating $\boldsymbol{\gamma}^{(t)}$ via Eq. (12);
 3:    Updating $\boldsymbol{\pi}^{(t)}$ via Eq. (13), and removing the component with $\pi_k^{(t)}=0$;
 4:    Updating $\boldsymbol{\eta}^{(t)}$ via Eq. (14);
 5:    Updating $\mathbf{U}^{(t)}, \mathbf{V}^{(t)}$ via Algorithm 1.
 6:    $t = t+1$;
 7: **end while**

C. Convergence Analysis of EM algorithm

In this subsection, we show the convergence property of the proposed EM algorithm for PMoEP model.

Theorem 3.1: Let $l_P^C(\mathbf{\Theta}) = l(\mathbf{\Theta}) - P(\boldsymbol{\pi}; \lambda)$, where $l(\mathbf{\Theta})$ is defined in (8). If we assume that $\{\mathbf{\Theta}^{(t)}\}$ is the sequence generated by Algorithm 2 and the sequence of likelihood values $\{l_P^C(\mathbf{\Theta}^{(t)})\}$ is bounded above, then there exits a constant $l^\star$ such that

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

where

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

and

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

The proof is listed in Appendix A.

D. Implementation Issues

In the proposed PMoEP algorithm, there are three involved preset parameters, K_{start}, p and λ. Throughout all our experiments, we just simply set K_{start} as a not large number as $4-10$ based on a coarse empirical estimate on the noise complexity inside data. Once K_{start} is initialized, the length of vector $\mathbf{p} = [p_1, p_2, \ldots, p_{K_{start}}]$ in PMoEP is determined. In all our experiments, the elements in $\mathbf{p}$ are selected ranging over the interval between 0.1 and 2. For the setting of parameter λ, we first provide a series of candidates λ and then adopt the modified BIC to select a good λ among these candidates based on the modified BIC criterion. This criterion has been proven to be able to yield consistent component number estimation of the finite Gaussian mixture model [14]. Specifically, the modified BIC criterion is defined as

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

Then we can select the proper $\hat{\lambda}$ by

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

where $|\Omega|$ is the number of non-missing elements, $\hat{K}$ 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) for a given λ.

IV. PMoEP with Markov Random Field

In this section, we first propose an advanced PMoEP-MRF model. Then, we introduce a variational EM (VEM) algorithm to solve it. Finally, we also show the convergence analysis for the proposed algorithm.

A. PMoEP-MRF Model

In some practical applications, we often have certain noise prior knowledge. By introducing the prior into modeling, noise can be more appropriately modeled and thus the performance of the model is expected to be further improved. In video data, we can utilize the spatial and temporal smoothness prior. Specifically, for a certain pixel in one video frame, the pixels located near it both spatially and temporally tend to have similar distribution to it. Therefore, by facilitating the local continuity of noise components, we can embed Markov Random Field (MRF) into the PMoEP model. Note that the random variable $\mathbf{z}_{ij}$ determines the cluster label of noise e_{ij} in PMoEP model, and the aforementioned spatial and temporal relationships among adjacent pixels imply that they incline to

possess similar $\mathbf{z}_{ij}$ values. Therefore, we integrate into the distribution of $\mathbf{z}_{ij}$ with such prior smoothness knowledge as:

$$\mathbf{z}_{ij} \sim \mathcal{M}(\mathbf{z}_{ij}; \boldsymbol{\pi}) \prod_{(p,q)\in\mathcal{N}(i,j)} \psi(\mathbf{z}_{ij}, \mathbf{z}_{pq}), \qquad (31)$$

where

$$\psi(\mathbf{z}_{ij}, \mathbf{z}_{pq}) = \frac{1}{C} \prod_k \exp\left[\tau(2z_{ijk}-1)(2z_{pqk}-1)\right], \qquad (32)$$

where τ is a positive scalar parameter (we set $\tau = 10$ in experiments), C is a normalization constant of $\psi(\mathbf{z}_{ij}, \mathbf{z}_{pq})$ and $\mathcal{N}(i,j)$ is the neighborhood of the (i,j) entry. Specifically, when z_{ijk} and z_{pqk} achieve the same value (0 or 1), $\psi(\mathbf{z}_{ij}, \mathbf{z}_{pq})$ will have higher value, and thus this term readily encode the expected prior information. After defining the new distribution of $\mathbf{z}_{ij}$, the distribution of $\mathbf{Z}$ can be written as

$$\mathbb{P}(\mathbf{Z}; \boldsymbol{\pi}) = \frac{1}{C} \prod_{i,j\in\Omega,k} \pi_k^{z_{ijk}} \\ \prod_{i,j\in\Omega,k} \prod_{(p,q)\in\mathcal{N}(i,j)} \exp\left[\tau(2z_{ijk}-1)(2z_{pqk}-1)\right]. \qquad (33)$$

Then, the *complete likelihood function* can be written as

$$\mathbb{P}(\mathbf{E}, \mathbf{Z}; \boldsymbol{\Theta}) = \mathbb{P}(\mathbf{E}|\mathbf{Z}; \boldsymbol{\eta})\mathbb{P}(\mathbf{Z}; \boldsymbol{\pi}) \\ = \frac{1}{C} \prod_{i,j\in\Omega,k} [\pi_k f_{p_k}(y_{ij}-\mathbf{u}_i\mathbf{v}_j^T; 0, \eta_k)]^{z_{ijk}} \\ \prod_{i,j\in\Omega,k} \prod_{(p,q)\in\mathcal{N}(i,j)} \exp[\tau(2z_{ijk}-1)(2z_{pqk}-1)], \qquad (34)$$

and the *complete log-likelihood function* is

$$l^C(\boldsymbol{\Theta}) = \log\mathbb{P}(\mathbf{E}, \mathbf{Z}; \boldsymbol{\Theta}) \\ = \sum_{i,j\in\Omega,k} z_{ijk}[\log\pi_k + \log f_{p_k}(y_{ij}-\mathbf{u}_i\mathbf{v}_j^T; 0, \eta_k)] \\ + \tau \sum_{i,j\in\Omega,k} \sum_{(p,q)\in\mathcal{N}(i,j)} (2z_{ijk}-1)(2z_{pqk}-1) + const. \qquad (35)$$

In the next section, we will introduce a variational EM algorithm to solve this PMoEP-MRF model in detail.

B. Variational EM algorithm for PMoEP-MRF model

Since EM requires the computation of conditional distribution $\mathbb{P}(\mathbf{Z}|\mathbf{E})$ which is not tractable. In such PMoEP-MRF model, we resort to the variational method that aims at optimizing a lower bound of $\log\mathcal{L}(\mathbf{E})$, denoted by

$$\mathcal{J}(R_{\mathbf{E}}) = \log\mathcal{L}(\mathbf{E}) - KL[R_{\mathbf{E}}(\mathbf{Z}), \mathbb{P}(\mathbf{Z}|\mathbf{E})], \qquad (36)$$

where KL denotes the Kullback$-$Leibler divergence, $\mathbb{P}(\mathbf{Z}|\mathbf{E})$ is the true conditional distribution of the indicator variables $\mathbf{Z}$ given $\mathbf{E}$, and $R_{\mathbf{E}}(\mathbf{Z})$ is an approximation of the conditional distribution. $\mathcal{J}(R_{\mathbf{E}})$ equals to $\log\mathcal{L}(\mathbf{E})$ if and only if $R_{\mathbf{E}}(\mathbf{Z}) = \mathbb{P}(\mathbf{Z}|\mathbf{E})$.

As shown above, we are not able to calculate $\mathbb{P}(\mathbf{Z}|\mathbf{E})$, so we will look for the best (in terms of KL divergence) $R_{\mathbf{E}}(\mathbf{Z})$ in a certain class of distributions. Specifically, we constrain the variational distribution $R_{\mathbf{E}}(\mathbf{Z})$ to have the following form:

$$R_{\mathbf{E}}(\mathbf{Z}) = \prod_{i,j} R(\mathbf{z}_{ij}; \gamma_{ij}), \qquad (37)$$

where $R(\mathbf{z}_{ij}; \gamma_{ij}) = \prod_{ij} \prod_k \gamma_{ijk}^{z_{ijk}}$, $\sum_k \gamma_{ijk} = 1$, and $\boldsymbol{\gamma}$ is the variational parameter. Then, the lower bound $\mathcal{J}(R_{\mathbf{E}})$ to be maximized can be written as

$$\mathcal{J}(R_{\mathbf{E}}) = E_{R_{\mathbf{E}}(\mathbf{Z})}\{\log\mathbb{P}(\mathbf{E}, \mathbf{Z})\} - E_{R_{\mathbf{E}}(\mathbf{Z})}\{R_{\mathbf{E}}(\mathbf{Z})\}, \\ = \sum_{i,j\in\Omega,k} [\log\pi_k + \log f_{p_k}(e_{ij}; 0, \eta_k)] \\ + \tau \sum_{i,j\in\Omega,k} \sum_{(p,q)\in\mathcal{N}(i,j)} (2\gamma_{ijk}-1)(2\gamma_{pqk}-1) \\ - \sum_{i,j\in\Omega,k} \gamma_{ijk}\log\gamma_{ijk} + const. \qquad (38)$$

We can easily adopt alternative search strategy for the maximization problem on $\mathcal{J}(R_{\mathbf{E}})$ by alternatively solving the sub-problems: (i) with respect to $R_{\mathbf{E}}$ and (ii) with respect to parameters $\mathbf{U}, \mathbf{V}, \boldsymbol{\pi}, \boldsymbol{\eta}$. The following Proposition 4.1 and 4.2 provide the solutions of optimization problem (i) and (ii), respectively.

Proposition 4.1: (Variational E-step) Given parameters $\boldsymbol{\Theta} = \{\mathbf{U}, \mathbf{V}, \boldsymbol{\pi}, \boldsymbol{\eta}\}$, the optimal variational parameters $\hat{\gamma}_{ij} = \arg\max_{\boldsymbol{\gamma}} \mathcal{J}(R_{\mathbf{E}})$ satisfy the following fixed point relation:

$$\gamma_{ijk} \propto \pi_k f_{p_k}(e_{ij}; 0, \eta_k) \exp\{\tau \sum_{(p,q)\in\mathcal{N}(i,j)} \gamma_{pqk}\}. \qquad (39)$$

PROOF. Based on (38), we maximize $\mathcal{J}(R_{\mathbf{E}})$ with respect to γ_{ij}s, subject to $\sum_k \gamma_{ijk} = 1$, for all i, j, i.e. to maximize $\mathcal{J}(R_{\mathbf{E}}) + \sum_{ij}[\lambda_{ij}(\sum_k \gamma_{ijk} - 1)]$ where λ_{ij} is the Lagrangian multiplier. The derivative with respect to γ_{ijk} is

$$\log\pi_k + \log f_{p_k}(e_{ij}; 0, \eta_k) + \tau \sum_{(p,q)\in\mathcal{N}(i,j)} \gamma_{pqk} - \log\gamma_{ijk} - 1 + \lambda_{ij}.$$

This derivative is null iff γ_{ijk} satisfy the relation given in the proposition, and $\exp(-1+\lambda_{ij})$ is the the normalizing constant. ∎

Proposition 4.2: (Variational M-step) Given the variational parameters γ_{ij}s, the values of parameters $\mathbf{U}, \mathbf{V}, \boldsymbol{\pi}, \boldsymbol{\eta}$ that maximize $\mathcal{J}(R_{\mathbf{E}})$ can be calculated in the same way as the M step in the EM algorithm of PMoEP model.

The proposed variational EM algorithm for PMoEP-MRF model can then be summarized in Algorithm 3.

Algorithm 3 VEM Algorithm for the PMoEP-MRF Model.

Input: Data $\mathbf{Y}$, rank r, τ and λ.
Output: Parameter $\boldsymbol{\Theta}$, mixture components number K_{final} and $\gamma = (\gamma_{ijk})_{m\times n\times K_{final}}$.
Initialization: $\boldsymbol{\Theta}^{(0)} = \{\boldsymbol{\pi}^{(0)}, \boldsymbol{\eta}^{(0)}, \mathbf{U}^{(0)}, \mathbf{V}^{(0)}\}$, the initial mixture components number K_{start}, preset candidates $\mathbf{p} = [p_1, \ldots, p_{K_{start}}]$, tolerance ϵ and $t = 0$.
1: **while** not converged **do**
2:     Updating $\gamma^{(t)}$ via the fixed-point Eq. (39);
3:     Updating $\boldsymbol{\pi}^{(t)}$ via Eq. (13), and removing the component with $\pi_k^{(t)} = 0$;
4:     Updating $\boldsymbol{\eta}^{(t)}$ via Eq. (14);
5:     Updating $\mathbf{U}^{(t)}, \mathbf{V}^{(t)}$ via Algorithm 1;
6:     $t = t + 1$.
7: **end while**

TABLE I
THE PARAMETER SELECTION FOR PMoG AND PMoEP.

	Parameter Selection	
	PMoG	PMoEP
Gaussian	$K_{final}=1$, $\lambda_{select}=0.01$	$K_{final}=1$, $p_{select}=2$, $\lambda_{select}=0.15$
Exponential Power	$K_{final}=3$, $\lambda_{select}=0.001$	$K_{final}=1$, $p_{select}=0.2$, $\lambda_{select}=0.3$
Laplace	$K_{final}=3$, $\lambda_{select}=0.001$	$K_{final}=1$, $p_{select}=1$, $\lambda_{select}=0.1$
Sparse	$K_{final}=2$, $\lambda_{select}=0.005$	$K_{final}=2$, $p_{select}=[2,2]$, $\lambda_{select}=0.005$
Mixture 1	$K_{final}=2$, $\lambda_{select}=0.01$	$K_{final}=2$, $p_{select}=[1.5,2]$, $\lambda_{select}=0.005$
Mixture 2	$K_{final}=1$, $\lambda_{select}=0.001$	$K_{final}=2$, $p_{select}=[0.5,2]$, $\lambda_{select}=0.005$

C. Convergence Analysis of Variational EM algorithm

In this subsection, we show the convergence property of the proposed EM algorithm for PMoEP model.

Theorem 4.3: Given λ, Algorithm 3 generates a sequence $\{\{\gamma_{ij}^{(t)}\}, \Theta^{(t)}\}\}_{t=1}^{\infty}$ which increases $\mathcal{J}(R_{\mathbf{E}})$ such that

$$\mathcal{J}(R_{\mathbf{E}}; \{\gamma_{ij}^{(t+1)}\}, \Theta^{(t+1)}) \geq \mathcal{J}(R_{\mathbf{E}}; \{\gamma_{ij}^{(t)}\}, \Theta^{(t)}). \quad (40)$$

PROOF. This is a direct consequence of Propositions 4.1 and 4.2, which both guarantee that $\mathcal{J}(R_{\mathbf{E}})$ monotonically increases in iteration. ∎

It is easy to see that $\mathcal{J}(R_{\mathbf{E}}; \{\gamma_{ij}^{(t)}\}, \Theta^{(t)})$ is upper bounded, and thus the convergence of Algorithm 3 can be guaranteed.

V. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed PMoEP method, its special case PMoG and the PMoEP-MRF method, we conducted a series of experiments on both synthetic and real data. Five state-of-the-art LRMF methods were considered for comparison, including Mixture of Gaussion method (MoG [25]), Laplace noise methods (CWM [26], RegL1ALM [46]) and Gaussian noise methods (Damped Wiberg (DW) [31] and SVD). All experiments were implemented in Matlab R2014a on a PC with 3.60GHz CPU and 12GB RAM.

A. Synthetic simulations

Several synthetic experiments with different noise settings were designed to compare the performance of the proposed methods and other competing methods. 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 $\mathbf{V}_{gt} \in \mathcal{R}^{20\times 4}$, and $\mathbf{Y}_{gt} = \mathbf{U}_{gt}\mathbf{V}_{gt}^T$ is the ground truth matrix. Then, we randomly specified 20% elements of $\mathbf{Y}_{gt}$ as missing entries. Next, we added different types of noise to the non-missing entries as follows: (1) *Gaussian noise*: $\mathcal{N}(0, 0.04)$. (2) *Exponential power noise*:[2] $EP_{0.2}(0, 0.2^p p), p = 0.2$. (3) *Laplace noise*: $\mathcal{L}(0, 0.2)$. (4) *Sparse noise*: 12.5% of the non-missing entries

---

[2]The method of drawing samples from a general exponential power distribution is introduced in Appendix B.

were corrupted with uniformly distributed noise on [-20,20]. (5) *Mixture noise 1*: 25% of the entries were corrupted with uniformly distributed noise on [-5,5], 25% were contaminated with Gaussian noise $\mathcal{N}(0, 0.04)$ and the remaining 50% are corrupted with Gaussian noise $\mathcal{N}(0, 0.01)$. (6) *Mixture noise 2*: 37.5% of the entries were corrupted with $EP(0, 0.1^p p), p = 0.5$, 50% were contaminated with Laplace noise $\mathcal{L}(0, 0.3)$ and the remaining 50% were corrupted with Gaussian noise $\mathcal{N}(0, 0.01)$. Then we get the noisy matrix $\mathbf{Y}_{no}$. 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 a method recovers the correct subspaces.

TABLE II
PERFORMANCE EVALUATION ON SYNTHETIC DATA. THE BEST RESULTS IN TERMS OF EACH CRITERION ARE HIGHLIGHTED IN BOLD.

	PMoEP	PMoG	MoG	DW	CWM	RegL1ALM
			Gaussian Noise			
C1	40.97	41.00	41.00	41.00	39.23	**36.60**
C2	**4.16**	**4.16**	**4.16**	**4.16**	5.67	5.27
C3	**3.27**	**3.27**	**3.27**	**3.27**	6.01	4.94
C4	**3.90e+1**	3.91e+1	3.91e+1	3.91e+1	5.09e+1	5.09e+1
C5	**4.22e-2**	**4.22e-2**	**4.22e-2**	**4.22e-2**	5.71e-2	5.33e-2
C6	**3.01e-2**	**3.01e-2**	**3.01e-2**	**3.01e-2**	4.55e-2	3.79e-2
			Exponential Power Noise			
C1	3.60e+2	3.42e+2	3.23e+2	4.30e+2	**3.21e+2**	3.65e+2
C2	1.30e+3	1.04e+3	1.18e+3	**6.27e+2**	8.51e+2	8.51e+2
C3	**1.72e+2**	4.49e+4	2.17e+3	5.06e+3	1.73e+2	7.77e+4
C4	**2.32e+2**	4.67e+2	2.60e+2	6.29e+2	2.40e+2	9.68e+2
C5	**3.31e-1**	5.67e-1	4.11e-1	9.19e-1	3.39e-1	1.16
C6	**2.19e-1**	4.97e-1	2.31e-1	8.94e-1	2.61e-1	1.11
			Laplacian Noise			
C1	7.63e+1	7.29e+1	7.13e+1	7.76e+1	7.24e+1	**6.80e+1**
C2	1.72e+1	2.57e+1	2.44e+1	**1.68e+1**	2.16e+1	2.10e+1
C3	**1.27e+1**	2.02e+1	1.84e+1	1.31e+1	1.69e+1	1.42e+1
C4	**7.54e+1**	9.37e+1	8.99e+1	7.69e+1	8.33e+1	7.85e+1
C5	**9.17e-2**	1.15e-1	1.07e-1	9.22e-2	1.07e-1	9.80e-2
C6	**6.30e-2**	8.25e-2	7.84e-2	6.49e-2	8.24e-2	6.56e-2
			Sparse Noise			
C1	**8.12e+2**	**8.12e+2**	1.17e+3	8.20e+2	8.73e+2	
C2	1.08e+4	1.08e+4	1.08e+4	**5.12e+3**	1.06e+4	5.95e+3
C3	**2.37e-12**	**2.37e-12**	7.94e-12	3.09e+4	9.75e+1	1.59e+6
C4	**2.54e-5**	2.55e-5	3.48e-5	2.12e+3	6.03e+1	4.89e+3
C5	**3.87e-8**	3.87e-8	6.63e-8	1.48	2.83e-1	1.47
C6	**2.28e-8**	2.29e-8	4.44e-8	1.39	6.25e-2	1.54
			Mixture Noise1			
C1	4.49e+2	4.55e+2	5.25e+2	5.25e+2	**4.33e+2**	4.35e+2
C2	1.36e+3	1.25e+3	**8.49e+2**	8.51e+2	1.12e+3	1.16e+3
C3	**1.53e+2**	6.52e+4	8.98e+2	8.93e+2	3.01e+2	1.56e+4
C4	**1.66e+2**	4.38e+2	6.02e+2	6.00e+2	2.87e+2	5.15e+2
C5	**3.28e-1**	5.79e-1	6.47e-1	6.60e-1	4.30e-1	7.88e-1
C6	**1.18e-1**	3.78e-1	5.01e-1	5.01e-1	2.93e-1	6.84e-1
			Mixture Noise2			
C1	9.01e+1	8.93e+1	8.76e+1	9.60e+1	8.83e+1	**8.32e+1**
C2	3.37e+1	4.04e+1	3.99e+1	**2.72e+1**	3.53e+1	3.42e+1
C3	**1.72e+01**	2.62e+1	2.49e+1	2.13e+1	2.40e+1	1.87e+1
C4	**8.57e+01**	1.04e+2	1.01e+2	9.71e+1	9.77e+1	8.87e+1
C5	**1.02e-01**	1.23e-1	1.24e-1	1.09e-1	1.21e-1	1.07e-1
C6	**6.39e-02**	8.41e-2	8.14e-2	7.02e-2	8.96e-2	6.62e-2

We set the rank of all the competing methods to 4 and adopt the random initialization strategy for all the methods. For each method, we first run with 20 random initializations and then
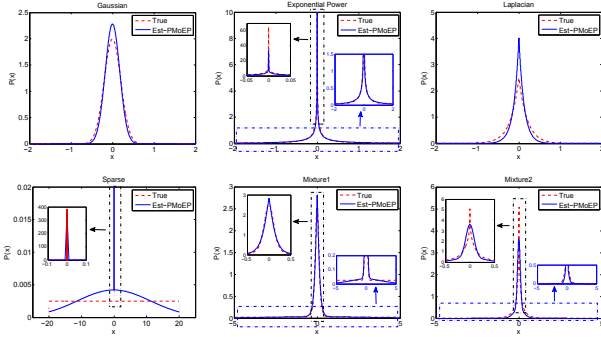
Fig. 3. Visual comparison of the ground truth (denote by True) noise probability density functions and those estimated (denote by Est) by the PMoEP method in the synthetic experiments. The embedded sub-figures depict the zoom-in of the indicated portions.
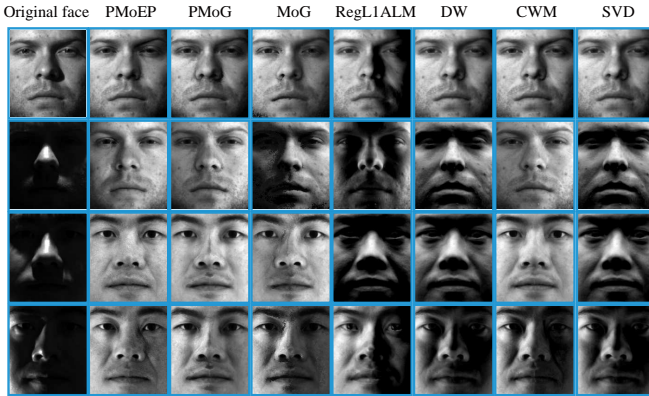

Fig. 4. From left to right: original face images, reconstructed faces by PMoEP, PMoG, MoG, RegL1ALM, DW, CWM and SVD.
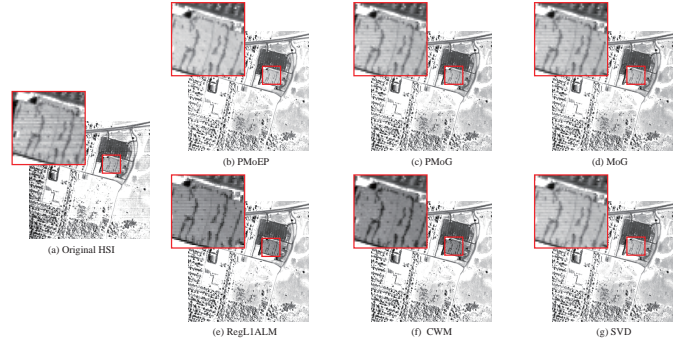

Fig. 5. Restoration results of band 103 in Urban data set: (a) original bands. (b)-(g) reconstructed bands by PMoEP, PMoG, MoG, RegL1ALM, CWM and SVD.

PMoEP method. It can be easily observed that the estimated noise distributions well match the true ones, which naturally conducts its good reconstruction capability to the true low-rank matrix.

B. Face modeling

This experiment aims to test the effectiveness of PMoG and PMoEP methods in face modeling application. We choose the first and the second subset of the Extended Yale B database[3], and each subset consists of 64 faces of one person with size 192×168 and then generate two data matrices, each of which is with size 32256×64. Typical images are shown in the first column of Fig. 4.

We set the rank as 4 [3] and adopt two initialization strategies, namely random and SVD for all competing methods. Then we report the best result among the results in terms of the object value of the corresponding model utilized by each method. Some reconstructed faces of different methods are visually compared in Fig. 4.

From Fig. 4, 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 and PMoG methods perform better than other ones on faces containing a large dark region. Such face images contain both significant cast shadow and saturation noises, which correspond to the highly dark and bright areas in face, and camera noise [29], which is much amplified in the dark areas. Compared with other competing methods, PMoEP method is capable of better extracting such complex noise configurations, and thus leads to its better face reconstruction performance.

C. Hyperspectral Image Restoration

In this section, we evaluate the performance of our proposed PMoEP method on hyperspectral image restoration problem. Two real hyperspectral image (HSI) data sets[4] were used.

The first dataset is Urban HSI data. This dataset contains 210 bands, each of which is 307×307, and some bands are seriously polluted by atmosphere and water and corrupted by

select the best result with respect to the corresponding objective value of the method. The performance of each method was evaluated as the average results over the 30 random matrices in terms of the six measures, and the results are summarized in Table II. We also report the final selections of the mixture number K_{final} and the corresponding parameter λ_{select} for PMoG and PMoEP in Table I.

From Table II, we can observe that L_2-norm methods DW, MoG, PMoG and our proposed PMoEP methods achieve the best performance than others in Gaussian noise case. In Laplace noise case, our PMoEP method performs best and L_1 method RegL1ALM achieves similar results. When the noise is Exponential Power, PMoEP evidently outperforms other competing methods in term of criteria C3−C6. In sparse noise case, PMoEP and PMoG perfom the best and MoG achieves comparable good results with PMoEP. Moreover, when the noise gets more complex, PMoEP achieves the best performance, which attributes to the high flexibility of PMoEP to model unknown complex noise. These results then substantiate that our proposed PMoEP method can estimate a better subspace from the noisy data than other competing methods.

The promising performance of PMoEP method in these cases can be easily explained by Fig. 3, which compares the ground truth noise distributions and the estimated ones by the
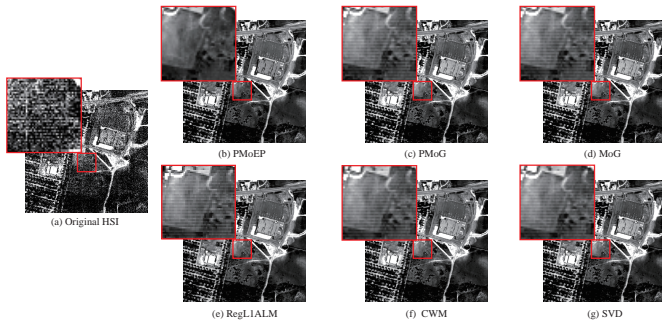
Fig. 6. Restoration results of band 206 in Urban data set: (a) original bands. (b)-(g) reconstructed bands by PMoEP, PMoG, MoG, RegL1ALM, CWM and SVD.
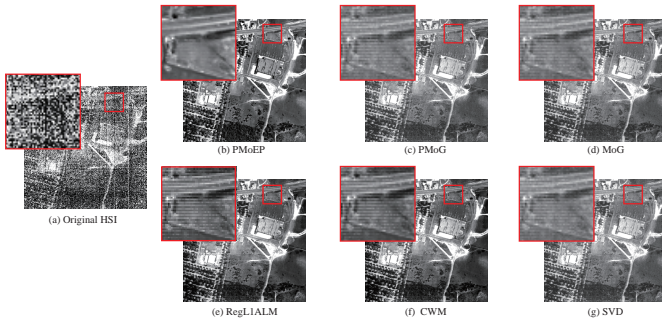

Fig. 7. Restoration results of band 207 in Urban data set: (a) original bands. (b)-(g) reconstructed bands by PMoEP, PMoG, MoG, RegL1ALM, CWM and SVD.
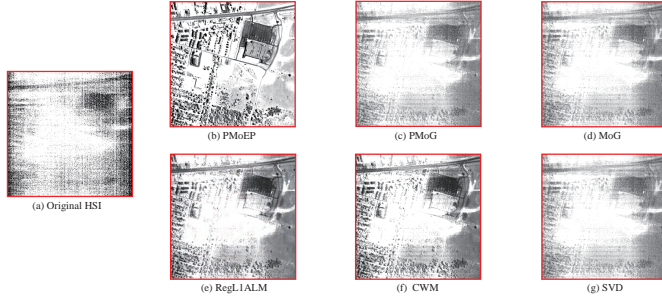

Fig. 8. Restoration results of band 107 in Urban data set: (a) original bands. (b)-(g) reconstructed bands by PMoEP, PMoG, MoG, RegL1ALM, CWM and SVD.
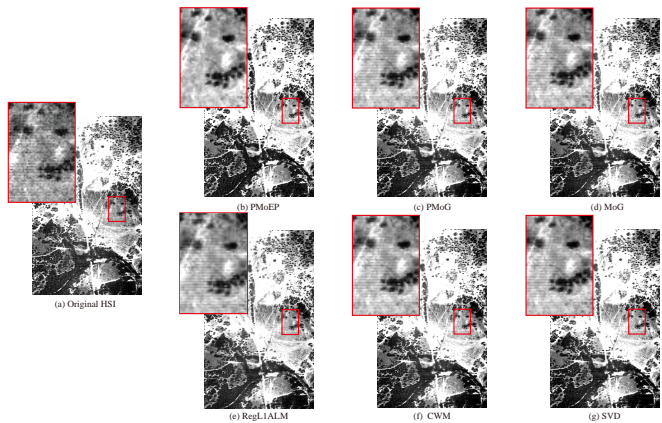

Fig. 9. Restoration results of band 152 in Terrain data set: (a) original bands. (b)-(g) reconstructed bands by PMoEP, PMoG, MoG, RegL1ALM, CWM and SVD.

Fig. 10. Restoration results of band 206 in Terrain data set: (a) original bands. (b)-(g) reconstructed bands by PMoEP, PMoG, MoG, RegL1ALM, CWM and SVD.
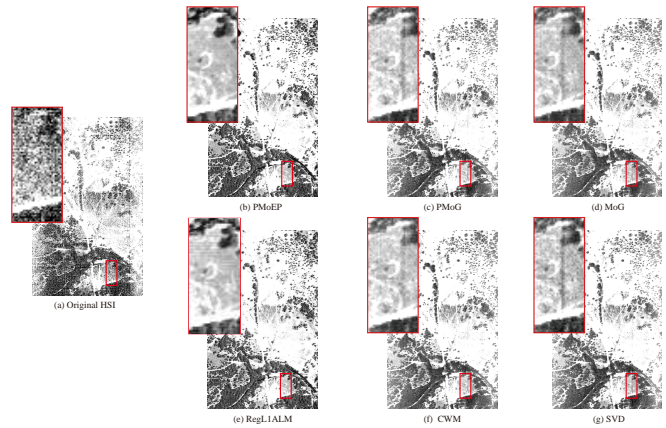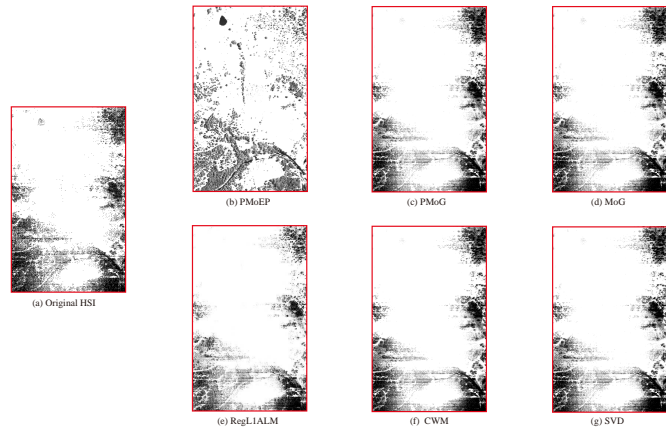

Fig. 11. Restoration results of band 139 in Terrain data set: (a) original bands. (b)-(g) reconstructed bands by PMoEP, PMoG, MoG, RegL1ALM, CWM and SVD.

noises with complex structures, as shown in Fig. 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. The second one is the Terrain dataset. The original images are of size $500 \times 307 \times 210$. We use all the bands in our experiments and thus generate a 153500×210 data matrix. Therefore, we get two data matrices used to test our methods. All the competing methods were implemented, except DW method which encounters the 'out of memory' problem.

The reconstructed hyperspectral images of bands 103, 206, 207 and 107 in Urban dataset and bands 152, 206 and 139 in Terrain dataset are shown in Fig. $5-8$ and Fig. $9-11$, respectively. For easy observation, an area of interest is amplified in the restored images obtained by all the competing methods. It can be easily seen from the figures that for some bands containing evident stripes and deadlines, the image restored by the proposed PMoEP method is clean and smooth, while the results obtained by the other competing ones contain evident stripe area. In addition, as is demonstrated in Fig. 8 and Fig. 11, the PMoEP method can effectively recover the seriously polluted bands, while the other methods failed on them. These results show that our proposed PMoEP method can not only remove complicated noises embedded in HSI, but also can
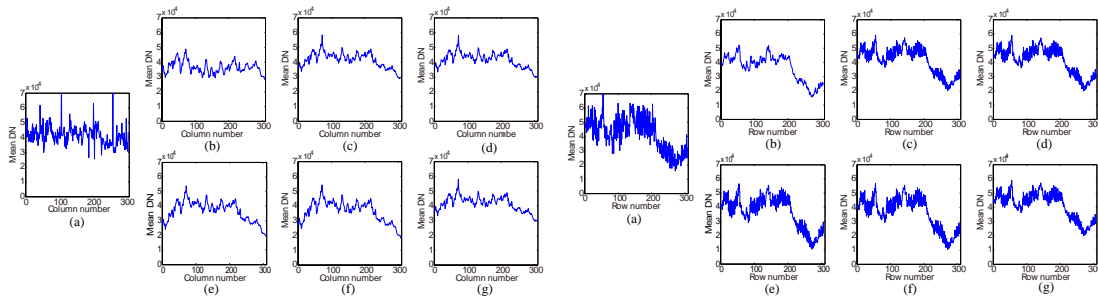
Fig. 12. Vertical (left) and Horizontal (right) mean profiles of band 207 in the Urban data set: (a) original, (b) PMoEP, (c) PMoG, (d) MoG, (e) RegL1ALM, (f) CWM, (g) SVD.
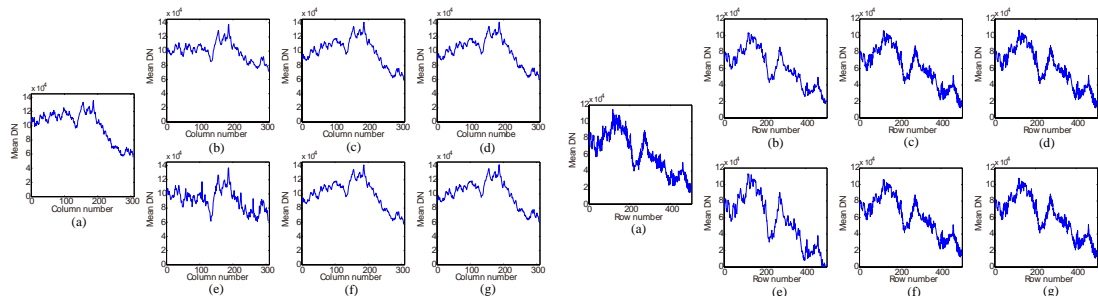

Fig. 13. Vertical (left) and Horizontal (right) mean profiles of band 152 in the Terrain data set: (a) original, (b) PMoEP, (c) PMoG, (d) MoG, (e) RegL1ALM, (f) CWM, (g) SVD.
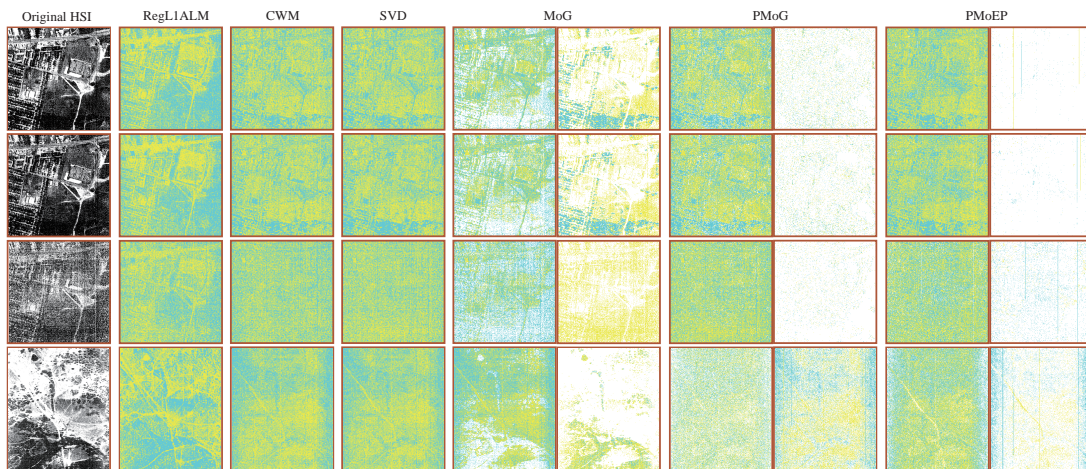

Fig. 14. From top to bottom, band 204, 206, 207 of Urban, band 208 of Terrain. From left to right: original bands, and extracted noise by RegL1ALM, CWM, SVD, MoG, PMoG and PMoEP. The noises with positive and negative values are depicted in yellow and blue, respectively. This figure should be viewed in color and the details are better seen by zooming on a computer screen.

perform robust in the presence of extreme outlier cases like in Fig. 8 and Fig. 11.

Then we give more quantitative comparison by showing the vertical mean profiles and horizontal mean profiles of band 207 in Urban dataset and band 152 in Terrain dataset before and after reconstruction in Fig. 12 and Fig. 13. The horizontal axis of Fig. 12 represents the column (left) and row (right) number, and the vertical axis represents the mean DN value of each column (left) and row (right). It is easy to observe that the curves in Fig. 12(a) and 13(a) (right) have drastic fluctuations for the original image. This is deviated from the prior knowledge that the adjacent bands should possess similar shapes since they are captured under relatively similar sensor settings. After the reconstruction, the fluctuations in vertical direction have been reduced by most of the methods. While in the horizontal direction (see Fig. 12 (right) and Fig. 13 (right)), the PMoEP method provides evidently smoother curves, which indicates that the stripes in the horizontal direction have been removed more effectively by our method. The results are consistent with the recovered HSIs in Fig. 7 and Fig. 9.

The better performance of PMoEP over other methods is due to its more powerful ability in noise modeling. Specifically, as depicted in Fig. 14, PMoEP can more properly extract noise information from the corrupted images with physical meanings, such as sparse strips, sparse deadlines, and dense Gaussian noise, while other competing methods fail to do so.
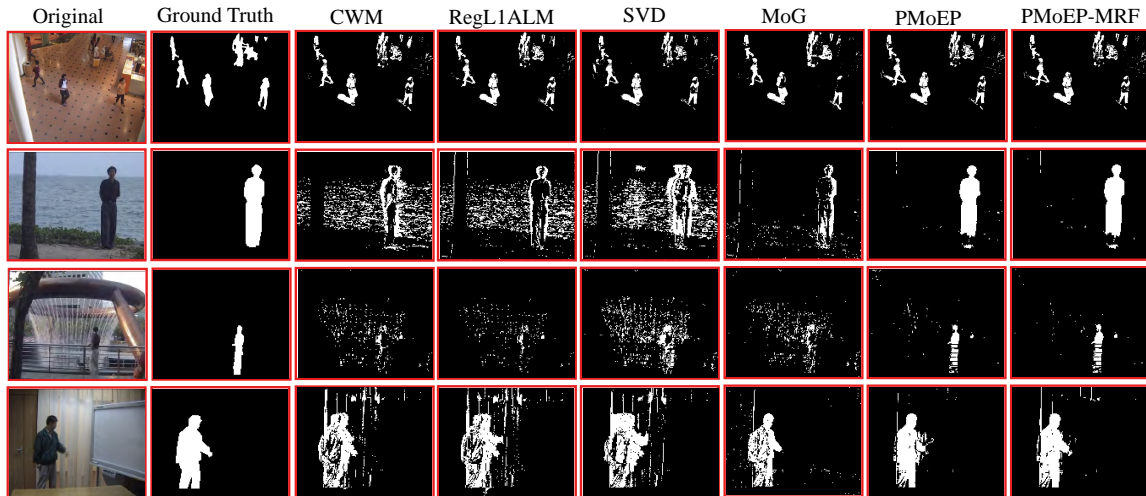
Fig. 15. Foreground Detection results of different methods on sample frames.

D. Background Subtraction

In this section, we evaluate the performance of our proposed methods on background subtraction problem. The background subtraction from a video sequence captured by a static camera can be modeled as a low-rank matrix analysis problem [42]. All the nine standard video sequences[5] provided by Li et.al [22] were adopted in our evaluation, including simple and complex background. Ground truth foreground regions of 20 frames were provided for each sequence.

We compared our PMoEP and PMoEP-MRF methods with the state-of-the-art LRMF methods: SVD, RegL1ALM, CWM and MoG methods. To conduct the experiments, we first ran each method on each video sequence to estimate the background. Then we obtained the recovered foreground by calculating the absolute values of the difference between the original frame and the estimated background. For MoG, PMoEP and PMoEP-MRF methods, we obtained the foreground by selecting the noise component with largest variance.

For quantitative evaluation, we first introduce some evaluation indices. We measure the recovery accuracy of the support in the foreground by comparing the true support S with the detected support $\tilde{S}$. We regard it as a classification problem and thus can evaluate the results using precision and recall, which are defined as:

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN},$$

where TP, FP, TN and FN represent the numbers of true positive, false positive, true negative and false negative, respectively. For simplicity, we adopt $F\text{-}measure$ that combines precision and recall together:

$$F\text{-}measure = 2 \times \frac{precision \times recall}{precision + recall}.$$

The higher F-measure value means the better recovery accuracy of the support. Additionally, the recovered support $\tilde{S}$ is obtained by thresholding the recovered foreground E

with a threshold value that gives the maximal F-measure. For all competing methods, we adopt two initialization strategies, namely, random and SVD. Then we report the best result among the two initializations. The results are summarized in Table III.

TABLE III
PERFORMANCE EVALUATION ON VIDEO DATA. THE BEST AND SECOND BEST RESULTS FOR EACH VIDEO DATASET ARE HIGHLIGHTED IN BOLD AND IN ITALIC BOLD, RESPECTIVELY.

Video	SVD	RegL1ALM	CWM	MoG	PMoEP	PMoEP-MRF
F-measure						
Campus	0.4716	**0.5308**	*0.5301*	0.4633	0.5065	0.5115
Lobby	0.7623	0.7679	*0.7681*	**0.7724**	0.7650	0.7444
ShoppingMall	0.6990	*0.7138*	**0.7173**	0.6387	0.7037	0.7015
Bootstrap	0.6234	**0.6749**	0.6533	0.4234	0.6404	*0.6635*
Hall	0.4104	0.4659	0.4624	0.4523	*0.5372*	0.5438
Curtain	0.5273	0.5342	0.5316	0.7869	**0.7895**	*0.7888*
Fountain	0.4989	0.5298	0.5262	0.5782	*0.6843*	0.7295
WaterSurface	0.3416	0.2840	0.2920	0.5979	*0.8515*	0.8651
Escalator	0.2675	0.2998	0.2972	0.2675	*0.3255*	0.3408
Average	0.5113	0.5334	0.5309	0.5534	*0.6448*	0.6543

From Table III, it can be easily seen that our proposed PMoEP and PMoEP-MRF methods outperform other methods in the sequences of Hall, Curtain, Fountain, WaterSurface and Escalator, of which the background is with complex shapes. For the sequences with simple background, including Bootstrap, ShoppingMall, Campus and Lobby, the performances of all the methods are almost the same. On average, the PMoEP method achieves the second best performance. Compared with the PMoEP method, the PMoEP-MRF method slightly improves the average performance due to the modeling of spatial and temporal smoothness prior knowledge under foreground using Markov random field.

The better performance of PMoEP and PMoEP-MRF methods can be visually shown in Fig. 15. It can be easily seen from the figure that the proposed PMoEP and PMoEP-MRF can perform comparably well as other methods in simple

foreground cases, while evidently better in much complicated scenarios, e.g., videos with dynamic background.

VI. CONCLUSIONS

In this paper, we model the noise of the LRMF problem as a Mixture of Exponential Power (MoEP) distributions and proposes a penalized MoEP (PMoEP) model by combining the penalized likelihood method with the MoEP distributions. Moreover, by facilitating the local continuity of noise components along both space and time of a video, we embed Markov random field into PMoEP and then propose the PMoEP-MRF model. Compared with the current LRMF methods, our PMoEP method performs better in a wide variety of synthetic and real complex noise scenarios including face modeling, hyperspectral image restoration, and background subtraction applications. Additionally, our methods are capable of automatically learning the number of components from data, and thus can be used to deal with more complex applications. In the future, we'll attempt to extend the noise modeling methodology under PMoEP to more computer vision and machine learning tasks, e.g., the high-order low rank tensor factorization problems.

ACKNOWLEDGEMENTS

APPENDIX A
PROOF OF THEOREM 1

PROOF. (i) First, we calculate that

$$
\begin{aligned}
l_P^C(\boldsymbol{\Theta}) - l_P^C(\boldsymbol{\Theta}^{(t)}) &= l(\boldsymbol{\Theta}) - l(\boldsymbol{\Theta}^{(t)}) + P(\pi^{(t)}; \lambda) - P(\pi; \lambda) \\
&= \log \sum_{\mathbf{Z}} \mathbb{P}(\mathbf{Z}|\mathbf{E}; \boldsymbol{\Theta}^{(t)}) \frac{\mathbb{P}(\mathbf{E}|\mathbf{Z}; \boldsymbol{\Theta})\mathbb{P}(\mathbf{Z}; \boldsymbol{\Theta})}{\mathbb{P}(\mathbf{Z}|\mathbf{E}; \boldsymbol{\Theta}^{(t)})} \\
&\quad - \log \mathbb{P}(\mathbf{E}; \boldsymbol{\Theta}^{(t)}) + P(\pi^{(t)}; \lambda) - P(\pi; \lambda) \\
&\geq \sum_{\mathbf{Z}} \mathbb{P}(\mathbf{Z}|\mathbf{E}; \boldsymbol{\Theta}^{(t)}) \log \frac{\mathbb{P}(\mathbf{E}|\mathbf{Z}; \boldsymbol{\Theta})\mathbb{P}(\mathbf{Z}; \boldsymbol{\Theta})}{\mathbb{P}(\mathbf{Z}|\mathbf{E}; \boldsymbol{\Theta}^{(t)})} \\
&\quad - \log \mathbb{P}(\mathbf{E}; \boldsymbol{\Theta}^{(t)}) + P(\pi^{(t)}; \lambda) - P(\pi; \lambda) \\
&= \sum_{\mathbf{Z}} \mathbb{P}(\mathbf{Z}|\mathbf{E}; \boldsymbol{\Theta}^{(t)}) \log \frac{\mathbb{P}(\mathbf{E}|\mathbf{Z}; \boldsymbol{\Theta})\mathbb{P}(\mathbf{Z}; \boldsymbol{\Theta})}{\mathbb{P}(\mathbf{Z}|\mathbf{E}; \boldsymbol{\Theta}^{(t)})\mathbb{P}(\mathbf{E}; \boldsymbol{\Theta}^{(t)})} \\
&\quad + P(\pi^{(t)}; \lambda) - P(\pi; \lambda).
\end{aligned}
$$

Let $\Omega(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}) = \sum_{\mathbf{Z}} \mathbb{P}(\mathbf{Z}|\mathbf{E}; \boldsymbol{\Theta}^{(t)}) \log \frac{\mathbb{P}(\mathbf{E}|\mathbf{Z}; \boldsymbol{\Theta})\mathbb{P}(\mathbf{Z}; \boldsymbol{\Theta})}{\mathbb{P}(\mathbf{Z}|\mathbf{E}; \boldsymbol{\Theta}^{(t)})\mathbb{P}(\mathbf{E}; \boldsymbol{\Theta}^{(t)})}$, then

$$
l_P^C(\boldsymbol{\Theta}) \geq l_P^C(\boldsymbol{\Theta}^{(t)}) + \Omega(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}) + P(\pi^{(t)}; \lambda) - P(\pi; \lambda).
$$

(ii) In the M step of Algorithm 1, it is obvious that

$$
\begin{aligned}
\boldsymbol{\Theta}^{(t+1)} &= \arg\max_{\boldsymbol{\Theta}} \left\{ \sum_{\mathbf{Z}} \mathbb{P}(\mathbf{Z}|\mathbf{E}; \boldsymbol{\Theta}^{(t)}) \log \mathbb{P}(\mathbf{E}, \mathbf{Z}; \boldsymbol{\Theta}) - P(\pi; \lambda) \right\} \\
&= \arg\max_{\boldsymbol{\Theta}} \left\{ \sum_{\mathbf{Z}} \mathbb{P}(\mathbf{Z}|\mathbf{E}; \boldsymbol{\Theta}^{(t)}) \frac{\log \mathbb{P}(\mathbf{E}, \mathbf{Z}; \boldsymbol{\Theta})}{\log \mathbb{P}(\mathbf{E}, \mathbf{Z}; \boldsymbol{\Theta}^{(t)})} - P(\pi; \lambda) \right\} \\
&= \arg\max_{\boldsymbol{\Theta}} \left\{ \Omega(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}) + P(\pi^{(t)}; \lambda) - P(\pi; \lambda) \right\}.
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
&\Omega(\boldsymbol{\Theta}^{(t+1)}|\boldsymbol{\Theta}^{(t)}) + P(\pi^{(t)}; \lambda) - P(\pi^{(t+1)}; \lambda) \\
&\geq \Omega(\boldsymbol{\Theta}^{(t)}|\boldsymbol{\Theta}^{(t)}) + P(\pi^{(t)}; \lambda) - P(\pi^{(t)}; \lambda) = 0
\end{aligned}
\tag{41}
$$

Then, we can easily derive that

$$
l_P^C(\boldsymbol{\Theta}^{(t+1)}) \geq l_P^C(\boldsymbol{\Theta}^{(t)}).
$$

Based on (42), the sequence $\{l_P^C(\boldsymbol{\Theta}^{(t)})\}_{t=1}^\infty$ is nondecreasing and bounded above. Therefore, there exits a constant $l^\star$ such that

$$
\lim_{t\to\infty} l_P^C(\boldsymbol{\Theta}^{(t)}) = l^\star.
$$

∎

APPENDIX B
EXPONENTIAL POWER DISTRIBUTION

A. Three different forms of Exponential Power Distribution

The Exponential Power Distribution ($\mu = 0$) has the following three equivalent forms:

$$
f_p(x; 0, \sigma) = \frac{1}{2\sigma p^{\frac{1}{p}} \Gamma(1 + \frac{1}{p})} \exp\left\{-\frac{|x|^p}{p\sigma^p}\right\}.
$$

Let $\tau = (p\sigma^p)^{\frac{1}{p}}$, then

$$
f_p(x; 0, \tau) = \frac{1}{2\tau \Gamma(1 + \frac{1}{p})} \exp\left\{-|\frac{x}{\tau}|^p\right\}.
$$

Let $\eta = \frac{1}{\tau^p}$, then

$$
f_p(x; 0, \eta) = \frac{\eta^{\frac{1}{p}}}{2\Gamma(1 + \frac{1}{p})} \exp\left\{-\eta|x|^p\right\}.
$$

Noting that $\Gamma(1 + \frac{1}{p}) = \frac{1}{p}\Gamma(\frac{1}{p})$, then we can represent the above three forms in equivalent forms.

B. Draw Samples from Exponential Power Distribution

The second form of exponential power distribution is

$$
f_p(x; 0, \tau) = \frac{1}{2\tau \Gamma(1 + \frac{1}{p})} \exp\left\{-|\frac{x}{\tau}|^p\right\}.
$$

Sampling from the exponential power distribution contains two cases: $p \geq 1$ and $0 < p < 1$.

1) case 1: $p \geq 1$: We adopt the method proposed in [9], [23], [27].

2) case 2: $0 < p < 1$: When $0 < p < 1$, the method proposed in [33] is used. We sample the distribution in two steps:

$$
(w|p) \sim \frac{1+p}{2} Ga(2 + \frac{1}{p}, 1) + \frac{1-p}{2} Ga(1 + \frac{1}{p}, 1), \tag{42}
$$

$$
(\beta|\tau, w, p) \sim \frac{1}{\tau w^{\frac{1}{p}}} \left\{1 - |\frac{\beta}{\tau w^{\frac{1}{p}}}|\right\}_+, \tag{43}
$$

where w is a intermediate variable. (42) can be sampled directly but (43) is difficult. Therefore, we adopt the slice sampling strategy in [4].

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] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.

[4] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[5] 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*, pages 316–322, 2005.

[6] X. Cao, Y. Chen, Q. Zhao, D. Meng, Y. Wang, D. Wang, and Z. Xu. Low-rank matrix factorization under general mixture noise. In *Proceedings of IEEE International Conference on Computer Vision*, 2015.

[7] 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.

[8] 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.

[9] M. Chiodi. Generation of pseudo random variates from a normal distribution of order p. *Statistica Applicata (Italian Journal of Applied Statistics)*, 7(4):401–416, 1995.

[10] 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.

[11] 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.

[12] A. Eriksson and A. Van Den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l1 norm. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 771–778, 2010.

[13] 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.

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

[15] 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.

[16] 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.

[17] Q. Ke and T. Kanade. Robust l1 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.

[18] 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.

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

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

[21] B. G. Leroux et al. Consistent estimation of a mixing distribution. *The Annals of Statistics*, 20(3):1350–1360, 1992.

[22] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.

[23] G. Marsaglia and T. A. Bray. A convenient method for generating normal variables. *Siam Review*, 6(3):260–264, 1964.

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

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

[26] 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.

[27] 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.

[28] 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.

[29] J. Nakamura. *Image sensors and signal processing for digital still cameras*. CRC press, 2005.

[30] 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.

[31] 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.

[32] D. Ormoneit and V. Tresp. Averaging, maximum penalized likelihood and bayesian estimation for improving gaussian mixture probability density estimates. *Neural Networks, IEEE Transactions on*, 9(4):639–650, 1998.

[33] N. G. Polson, J. G. Scott, and J. Windle. The bayesian bridge. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):713–733, 2014.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi. Practical low-rank matrix approximation under robust l1-norm. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1417, 2012.

[47] Z. Zivkovic and F. Van Der Heijden. Recursive unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):651–656, 2004.