




Robust CP Tensor Factorization With Skew Noise

Xingfang Huang, Shuang Xu , Chunxia Zhang , and Jianshe Zhang 

Abstract—The low-rank tensor factorization (LRTF) technique has received increasing popularity in data science, especially in computer vision applications. Many robust LRTF models have been presented recently. However, none of them take the skewness of data into account. This letter proposes a novel LRTF model for skew data analysis by modeling noise as a Mixture of Asymmetric Laplacians (MoAL). The numerical experiments show that the new model MoAL-LRTF outperforms several state-of-the-art counterparts. The codes for all the experiments are available at <https://xsxjtu.github.io/Projects/MoAL/main.html>.

Index Terms—Tensor factorization, mixture of asymmetric Laplacians, expectation-maximization (EM) algorithm.

I. INTRODUCTION

THE low-rank model is an extremely important tool in fields of signal processing and machine learning [1]. Essentially, many problems can be cast into low-rank models including multimodal data fusion [2], image denoising [3], representation learning [4] and etc. Recently, low-rank tensor factorization (LRTF) has emerged as a generalization of low-rank matrix factorization (LRMF) [1] because lots of data (e.g., RGB images and videos) are naturally represented in the tensor format so that the spatial information can be maintained. There are different formulations for LRTF, but this letter focuses on CANDECOMP/PARAFAC (CP) decomposition [5] on account of its widespread applications.

Given a K -order tensor $\mathcal{X} \in R^{I_1 \times I_2 \times \dots \times I_K}$, its CP decomposition is defined by

$$\mathcal{X} \approx \sum_{d=1}^r \mathbf{u}_{:,d} \circ \mathbf{v}_{:,d} \circ \dots \circ \mathbf{t}_{:,d}, \quad (1)$$

where r denotes the rank of the tensor \mathcal{X} . Here, $\mathbf{u}_{:,d}, \mathbf{v}_{:,d}, \dots, \mathbf{t}_{:,d}$ are column vectors of $\mathbf{U}, \mathbf{V}, \dots, \mathbf{T}$ with length I_1, I_2, \dots, I_K , respectively. The symbol \circ represents outer product.

Manuscript received March 16, 2020; revised April 23, 2020; accepted April 26, 2020. Date of publication April 30, 2020; date of current version June 3, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102201, in part by the Fundamental Research Funds for the Central Universities under Grant xzy022019059, and in part by the National Natural Science Foundation of China under Grant 11671317, Grant 61976174, and Grant 61877049. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Demetrio Labate. (Corresponding author: Chunxia Zhang.)

Xingfang Huang is with the School of Statistics and Mathematics, Nanjing Audit University, Nanjing 211815, China (e-mail: xfhuang@nau.edu.cn).

Shuang Xu, Chunxia Zhang, and Jianshe Zhang are with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: shuangxu@stu.xjtu.edu.cn; cxzhang@mail.xjtu.edu.cn; jszhang@mail.xjtu.edu.cn).

This letter has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors.

Digital Object Identifier 10.1109/LSP.2020.2991581

In this manner, the element $x_{ij\dots k}$ of \mathcal{X} can be represented as $x_{ij\dots k} \approx \sum_{d=1}^r u_i^d v_j^d \dots t_k^d$. It is worth pointing out that the loss function of the CP decomposition is an L_2 -norm function, where the underlying assumption is that the observed tensor is corrupted by Gaussian noise. Some evidence has shown that CP decomposition cannot match up our expectations in practice because real-world data are generally corrupted by heavy-tailed noise instead of Gaussian noise [6].

To solve this problem, many robust LRTF models have been proposed [6]–[10]. One of the most promising strategies is to model noise as a Mixture of Gaussians (MoG). For example, Meng *et al.* [11] and Chen *et al.* [6] apply MoG to LRMF and LRTF, respectively. Their experiments demonstrate that the MoG technique is far more effective than other state-of-the-art counterparts. Its good performance can be explained by a property of MoG, i.e., MoG is a universal approximator to any continuous distribution [12]. In the context of low-rank models, this property enables MoG-based models to fit the unknown noise item well. From the viewpoint of optimization, MoG-based models can optimize a loss function in the re-weighting manner. Therefore, it is rational to employ MoG to replace a specific distribution, such as Gaussian and Laplacian distributions. Although MoG technique has significantly boosted the performance of LRTF, there are still two intrinsic shortcomings. First, it may require infinite Gaussian components to approximate a distribution (e.g., Laplace distribution) in theory [12]. In practice, finite components have to be adopted instead. This obviously violates the assumption of the universal approximation property [13], [14]. Second, MoG distribution is symmetric. As stated in [15], [16], however, there is no strictly symmetric noise in real-world data.

In this letter, we present a novel LRTF model to improve the performance of LRTF for analyzing data corrupted by skew noise. The key idea is to model noise as a Mixture of Asymmetric Laplacians (MoAL). We utilize the expectation-maximization (EM) algorithm to infer our model. The merits of our model are two-fold. First, it can automatically estimate the skewness of observed data, and it outperforms other competitive methods if noise is corrupted by skew noise. Second, MoAL potentially does better than MoG to approximate complex distributions.

II. MODEL FORMULATION

A. Motivation

The past decade has witnessed the rapid development of robust LRTF models. The most used ones may be L_1 -norm based (i.e., Laplacian noise) and MoG-based LRTF models [6], [10]. However, as stated in [15], [16], skew and

heavy-tailed data are prevalent in real applications. To the best of our knowledge, however, there is not any LRTF model which is specially designed for analyzing skew data. To fill this gap, we employ an MoAL to model noise. The components of an MoAL are asymmetric Laplacian (AL) distributions whose probability density function [17] is given by $AL(x|\alpha, \lambda, \kappa) = \lambda\kappa(1 - \kappa) \exp\{-|x - \alpha|\lambda\rho\}$, where $\rho = (1 - \kappa)\mathbb{I}(x - \alpha < 0) + \kappa\mathbb{I}(x - \alpha \geq 0)$. The notations $\alpha, \lambda \in (0, +\infty)$ and $\kappa \in (0, 1)$ denote the location, scale and skewness parameters, respectively. Note that $AL(\alpha, \lambda, 0.5)$ becomes a Laplace distribution. Therefore, AL distribution meets our demands owing to its properties of skewness and heavy tail.

B. MoAL-LRTF

The letter focuses on the CP decomposition of a 3-order tensor \mathcal{X} due to its widespread applications, where $x_{ijk} = \sum_{d=1}^r u_i^d v_j^d t_k^d + e_{ijk}$. Aiming at handling skew data better, we assume that the noise item is distributed as an MoAL, viz.,

$$p(e_{ijk}) = \sum_{s=1}^S \pi_s AL_s(e_{ijk}|0, \lambda_s, \kappa_s), \quad (2)$$

where π_s is the mixing coefficient of the s th component satisfying $\sum_{s=1}^S \pi_s = 1$, and S denotes the number of components. To facilitate the modeling process, we introduce a latent binary random vector $\mathbf{z}_{ijk} = (z_{ijk1}, \dots, z_{ijkS})$, where $z_{ijk} = \{0, 1\}$ indicates whether the (i, j, k) th entry is corrupted by the s th component or not. In this way, the probability function of $\mathcal{Z} = \{z_{ijk}, i = 1, \dots, I_1, j = 1, \dots, I_2, k = 1, \dots, I_3\}$ can be represented as $p(\mathcal{Z}) = \prod_{i=1}^{I_1} \prod_{j=1}^{I_2} \prod_{k=1}^{I_3} \prod_{s=1}^S \pi_s^{z_{ijk}}$.

Now, we can write the conditional distribution of the observed data \mathcal{X} , that is,

$$p(\mathcal{X}|\mathcal{Z}) = \prod_{i,j,k \in \Omega} \prod_{s=1}^S \left[AL_s \left(x_{ijk} \left| \sum_{d=1}^r u_i^d v_j^d t_k^d, \lambda_s, \kappa_s \right. \right) \right]^{z_{ijk}}, \quad (3)$$

where the index set Ω collects all non-missing elements of \mathcal{X} . Our aim is to maximize the log-likelihood function with respect to (w.r.t.) $\mathbf{U} = (\mathbf{u}^1, \dots, \mathbf{u}^r)$, $\mathbf{V} = (\mathbf{v}^1, \dots, \mathbf{v}^r)$, $\mathbf{T} = (\mathbf{t}^1, \dots, \mathbf{t}^r)$, $\mathbf{\Lambda} = \{\lambda_1, \dots, \lambda_S\}$, $\mathbf{K} = \{\kappa_1, \dots, \kappa_S\}$, $\mathbf{\Pi} = \{\pi_1, \dots, \pi_S\}$, that is,

$$\max \sum_{i,j,k \in \Omega} \sum_{s=1}^S z_{ijk} \log AL_s \left(x_{ijk} \left| \sum_{d=1}^r u_i^d v_j^d t_k^d, \lambda_s, \kappa_s \right. \right). \quad (4)$$

C. Inference of MoAL-LRTF

As we know, the EM algorithm is the most effective tool to infer models involving latent variables. In E-step, it infers the posterior distribution of the latent variables $p(\mathcal{Z}|\mathcal{X})$, and computes the Q -function, i.e., $\mathbb{E}_{\mathcal{Z}|\mathcal{X}}[\log p(\mathcal{X}|\mathcal{Z})]$. In M-step, it maximizes the Q -function w.r.t. unknown parameters. The E- and M-steps are alternatively implemented until the EM algorithm converges. In what follows, residuals are denoted by $r_{ijk} = x_{ijk} - \sum_{d=1}^r u_i^d v_j^d t_k^d$, and $AL_s(r_{ijk}|0, \lambda_s, \kappa_s)$ is abbreviated as $AL_s(r_{ijk})$ to ease presentation.

1) *E-Step*: According to Bayesian formula, we have

$$\begin{aligned} p(z_{ijk}|x_{ijk}) &= \frac{p(x_{ijk}|z_{ijk})p(z_{ijk})}{\sum_{a=1}^S p(x_{ijk}|z_{ijka})p(z_{ijka})} \\ &= \frac{\pi_s AL_s(r_{ijk})}{\sum_{a=1}^S \pi_a AL_a(r_{ijk})} \equiv \gamma_{ijks}. \end{aligned} \quad (5)$$

Then, the Q -function can be obtained as

$$\begin{aligned} Q &= \mathbb{E}_{\mathcal{Z}|\mathcal{X}}[\log p(\mathcal{X}|\mathcal{Z})] \\ &= \sum_{i,j,k \in \Omega} \sum_{s=1}^S \gamma_{ijks} \log AL_s \left(x_{ijk} \left| \sum_{d=1}^r u_i^d v_j^d t_k^d, \lambda_s, \kappa_s \right. \right) \\ &= \sum_{i,j,k \in \Omega} \sum_{s=1}^S \gamma_{ijks} \{ \log \pi_s + \log \lambda_s \kappa_s (1 - \kappa_s) \\ &\quad - |r_{ijk}| \lambda_s \rho_{ijks} \}. \end{aligned} \quad (6)$$

2) *M-Step*: In M-step, the unknown parameters are alternatively updated. For the mixing coefficient π_s , it is ought to solve the constraint optimization problem

$$\max_{\pi_s} \sum_{i,j,k \in \Omega} \sum_{s=1}^S \gamma_{ijks} \log \pi_s, \text{ s.t. } \sum_{s=1}^S \pi_s = 1. \quad (7)$$

It is easy to acquire the solution as

$$\pi_s = \frac{N_s}{N}, \text{ where } N_s = \sum_{i,j,k \in \Omega} \gamma_{ijks}, \quad (8)$$

where N denotes the cardinality of Ω . For the scale parameter λ_s , it can be solved by $\partial Q / \partial \lambda_s = 0$, and the solution is

$$\lambda_s = \frac{N_s}{\sum_{i,j,k \in \Omega} \rho_{ijks} \gamma_{ijks} |r_{ijk}|}. \quad (9)$$

As for the skewness parameter κ_s , we can obtain a 2-order equation

$$\eta_s \kappa_s^2 - (2N_s + \eta_s) \kappa_s + N_s = 0, \quad (10)$$

by setting $\partial Q / \partial \kappa_s = 0$ and $\eta_s = \lambda_s \sum_{i,j,k \in \Omega} \gamma_{ijks} r_{ijk}$. Since κ_s satisfies $0 < \kappa_s < 1$, there is a unique root for our problem, i.e.,

$$\kappa_s = \frac{2N_s + \eta_s - \sqrt{4N_s^2 + \eta_s^2}}{2\eta_s}. \quad (11)$$

At last, we rewrite Eq. (6) w.r.t. \mathbf{U} , \mathbf{V} and \mathbf{T} as

$$\begin{aligned} &- \sum_{i,j,k \in \Omega} \sum_{s=1}^S \lambda_s \gamma_{ijks} \rho_{ijks} \left| x_{ijk} - \sum_{d=1}^r u_i^d v_j^d t_k^d \right| \\ &= - \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} \tilde{w}_{ijk} \left| x_{ijk} - \sum_{d=1}^r u_i^d v_j^d t_k^d \right|, \end{aligned} \quad (12)$$

where $\tilde{w}_{ijk} = \sum_{s=1}^S \lambda_s \gamma_{ijks} \rho_{ijks}$ if $i, j, k \in \Omega$, and 0 otherwise. Hence, it corresponds to an L_1 norm based LRTF problem. It is not easy to solve this optimization due to the non-convexity.

Algorithm 1: MoAL-LRTF.**Input:** The observed data $\mathcal{X} \in R^{I_1 \times I_2 \times I_3}$ **Output:** The recovered low-rank tensor $\hat{\mathcal{X}}$

- 1: Initialize unknown parameters.
- 2: **while** not converged **do**
- 3: E-step: Update γ_{ijk} by Eq. (5).
- 4: M-step1: Update $\pi_s, \lambda_s, \kappa_s$ by Eqs. (8), (9), (11).
- 5: M-step2: Update $\mathbf{U}, \mathbf{V}, \mathbf{T}$ by solving Eq. (13).
- 6: **end while**

For simplicity, we convert it into an L_2 norm based LRTF problem, namely,

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{T}} \left\| \mathcal{W} \odot \left(\mathcal{X} - \sum_{d=1}^r \mathbf{u}_{:d} \circ \mathbf{v}_{:d} \circ \mathbf{t}_{:d} \right) \right\|_{L_2}^2, \quad (13)$$

where $w_{ijk} = [\tilde{w}_{ijk} / (|x_{ijk} - \sum_{d=1}^r u_i^d v_j^d t_k^d| + \xi)]^{1/2}$ if $i, j, k \in \Omega$ and 0 otherwise. Here, ξ is a small constant for numerical stability. Note that the problem in Eq. (13) can be solved by alternative least squares (ALS). The key idea is to regard each slice of the tensor as a linear combination of the corresponding slices of all the rank-1 tensors. Let \mathcal{X}' denote by $\mathcal{W} \odot \mathcal{X}$. Taking \mathbf{T} as an example, the vectorized form of the k th slice of the tensor \mathcal{X}' along the third dimension can be expressed as $\mathbf{m}_{:k} = \text{vec}(\mathbf{X}'_{::k}) \in R^{I_1 I_2}$, where $\text{vec} : R^{a \times b} \rightarrow R^{ab}$ stands for the vectorization operator. Its corresponding vectorized slices of all the rank-1 tensors can be written as

$$\mathbf{F}_k = [\text{vec}(\mathbf{W}_{::k} \odot (\mathbf{u}_{:1} \circ \mathbf{v}_{:1})), \dots, \text{vec}(\mathbf{W}_{::k} \odot (\mathbf{u}_{:r} \circ \mathbf{v}_{:r}))] \in R^{I_1 I_2 \times r}. \quad (14)$$

Consequently, the update of i th row vector of \mathbf{T} is obtained as

$$\mathbf{t}_{i:} = (\mathbf{F}_k^\dagger \mathbf{m}_{:k})^T \in R^r. \quad (15)$$

Note that the symbols \dagger and T denote the pseudo-inverse and transpose operators, respectively. Finally, the matrix \mathbf{T} is recovered row-by-row with Eq. (15). As for \mathbf{U} and \mathbf{V} , they can be updated in the similar way. Algorithm 1 summarizes the workflow of the proposed MoAL-LRTF model.

III. EXPERIMENTS

A series of experiments are conducted in this section to study the behavior of our proposed MoAL-LRTF, compared with MoAL-LRMF [15], MoG-LRTF [6], MoG-LRMF [11] and CP [18]. The first three methods were implemented with their official codes. As for CP, we utilized the function `cp_wopt` in the Matlab tensor toolbox [19].

A. Experiment 1

The aim of this synthetic experiment is to verify the performance of all models on a low-rank recovery task. At first, three matrices $\mathbf{U} \in R^{I_1 \times r}$, $\mathbf{V} \in R^{I_2 \times r}$, $\mathbf{T} \in R^{I_3 \times r}$ were randomly sampled from the standard Gaussian distribution $\mathcal{N}(0, 1)$. Then, the ground truth tensor was constructed by $\mathcal{X} = \sum_{d=1}^r \mathbf{u}_{:d} \circ \mathbf{v}_{:d} \circ \mathbf{t}_{:d}$. At last, 20% of the elements in \mathcal{X} were stochastically set as missing data, and the tensor \mathcal{X} was corrupted by

TABLE I

MEAN L_1 ERROR OF ALL MODELS. THE BEST AND SECOND BEST VALUES ARE HIGHLIGHTED BY BOLD AND ITALIC TYPEFACE, RESPECTIVELY

	Noise	LRTF			LRMF	
		MoAL(Ours)	MoG	CP	MoAL	MoG
Case I	I	0.0368	0.0407	0.0366	0.1064	0.1013
	II	0.1413	0.1957	<i>0.1903</i>	0.4330	0.4696
	III	0.0827	<i>0.1008</i>	0.1359	0.2942	0.3401
	IV	0.0892	<i>0.1461</i>	0.1485	0.3645	0.3907
Case II	I	0.0056	0.0057	0.0073	0.0283	0.0302
	II	0.0105	<i>0.0206</i>	0.0355	0.0758	0.1531
	III	0.0114	<i>0.0128</i>	0.0261	0.0592	0.0742
	IV	0.0088	<i>0.0154</i>	0.0290	0.0601	0.1056

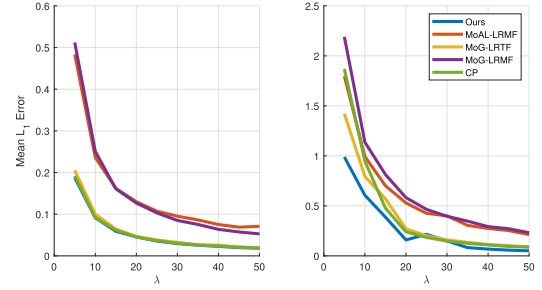


Fig. 1. The performance of all methods at different noise levels.

noise. Here, we considered four kinds of noise: I. Laplacian noise: $AL(0, 25, 0.5)$; II. AL noise: $AL(0, 25, 0.1)$; III. Mixture noise I: $0.2\mathcal{N}(0, 0.1^2) + 0.2\mathcal{N}(0, 0.15^2) + 0.6AL(0, 25, 0.1)$; IV. Mixture noise II: $0.2AL(0, 25, 0.2) + 0.2AL(0, 25, 0.15) + 0.6AL(0, 25, 0.1)$.

The first noise is heavy-tailed, and the others are not only heavy-tailed but also asymmetric. We set the rank $r = 5$. As for the size of \mathcal{X} , we considered two cases, that is, Case I: $(I_1, I_2, I_3) = (10, 10, 10)$, and Case II: $(I_1, I_2, I_3) = (50, 50, 50)$. The performance is evaluated by the mean L_1 error, i.e., $\|\mathcal{X} - \hat{\mathcal{X}}\|_{L_1} / I_1 I_2 I_3$, where $\hat{\mathcal{X}}$ is the recovered tensor. To eliminate randomness, we reported the median of the mean L_1 errors over 30 independent experiments.

As shown in Table I, it lists the metrics of all models in different situations. It is found that our method MoAL-LRTF always achieves the lowest mean L_1 error. Notice that MoG-based methods require infinite Gaussian components to approximate Laplacian noise. In contrast, MoAL-based methods need only one component to fit Laplacian noise well. And the other kinds of noise are asymmetric. In this situation, MoAL can automatically detect the skewness of noise whereas MoG does not consider this factor. Therefore, it is reasonable for MoAL to perform better when data is corrupted by Laplacian noise or skew noise.

B. Experiment 2

Here, we tested the performance of all methods with different noise levels. The experimental data were corrupted by Laplacian and AL noise, and the scale parameter λ is set to 5, 10, \dots , 50. The noise is more heavy-tailed as λ gets smaller. Fig. 1 visualizes the mean L_1 error curves versus λ . It is demonstrated that our model (blue line) MoAL-LRTF is the best performer no matter how λ changes.

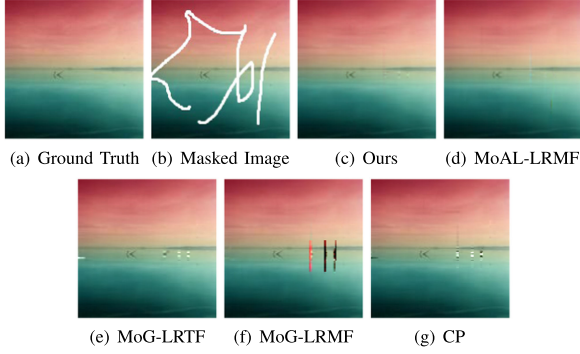


Fig. 2. Visual inspection of the image inpainting task (image A). More results can be found in the supplementary materials.

TABLE II
PSNR AND SSIM VALUES OF THE IMAGE INPAINTING TASK. THE BEST AND SECOND BEST VALUES ARE HIGHLIGHTED BY BOLD AND ITALIC TYPEFACE, RESPECTIVELY

Image	Metrics	LRTF			LRMF	
		MoAL(Ours)	MoG	CP	MoAL	MoG
A	PSNR	45.2588	38.0568	35.6572	<i>43.1923</i>	26.3785
	SSIM	0.9982	0.9932	0.9899	<i>0.9974</i>	0.9700
B	PSNR	29.6984	28.6884	28.4888	<i>29.4371</i>	28.6504
	SSIM	0.9360	<i>0.9312</i>	<i>0.9312</i>	0.9251	0.9233
C	PSNR	<i>20.3112</i>	19.8447	18.3693	20.5644	18.0956
	SSIM	<i>0.8816</i>	0.8815	0.8758	0.8822	0.8584

C. Experiment 3

In this subsection, we apply all low-rank models to an image inpainting task. Because images often have low-rank structure, the image inpainting task can be done by tensor (or matrix) completion if taking the corrupted pixels as missing values.

We selected a typical image (i.e., image A) as an example, as displayed in Fig. 2(a). The image is scrawled manually by white lines. More results conducted with images B and C were displayed in supplemental materials. We employ peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) to evaluate the performance of all models. Table II reports the PSNR and SSIM values of all models. Obviously, MoAL-LRTF outperforms all the other models w.r.t. both PSNR and SSIM. Moreover, both MoAL-based models behave better than MoG-based ones. As for visual effects, Fig. 2(c)–(g) display the recovered images. It is found that there are evident artifacts in the images recovered by MoG-LRTF, MoG-LRMF and CP. In contrast, there is high agreement between the images reconstructed by MoAL-based methods and the ground truth.

It is very interesting to investigate the reason why our proposed method outperforms others, even though the input tensor is not corrupted by noise. Fig. 3 displays the histograms and probability density curves estimated by the maximum likelihood principle. The curves in left and right panels are with AL and Gaussian assumptions, respectively. With the AL assumption, the parameters $(\alpha, \lambda, \kappa)$ for missing and non-missing data are (0.5686, 13.0644, 0.5485) and (0.5216, 10.6586, 0.5532), respectively. Regarding the Gaussian assumption, the mean and standard deviation (μ, σ) for missing and non-missing data are (0.5428, 0.1949) and (0.4774, 0.2333), respectively. From

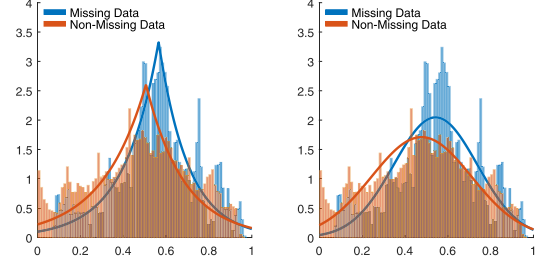


Fig. 3. Histograms and estimated probability density curves.

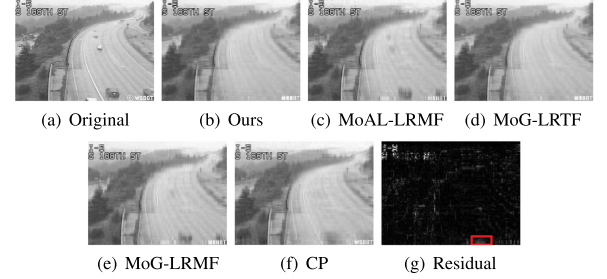


Fig. 4. Visual inspection of the background extraction task.

these statistical measures and Fig. 3, the good performance of MoAL-LRTF can be explained below. First, it is shown that AL distributions fit skew data better than Gaussian distributions. Second, the parameters $(\alpha, \lambda, \kappa)$ for the AL distributions on missing and non-missing data differ slightly, while it is not the case for Gaussian distributions. In other words, missing and non-missing data have similar AL distributions but dissimilar Gaussian distributions. Hence, it is more likely for MoAL-based methods to predict missing data with the knowledge learned from non-missing data.

D. Experiment 4

Besides the synthetic experiments, we applied low-rank models to a background extraction task. Given a video (or say, image sequence), its background can be modeled by a low-rank tensor, while its foreground is regarded as noise. Therefore, background extraction tasks can be typically solved by low-rank models. In this experiment, a real-world dataset named by freeway is employed. It contains 44 frames of size 316×236 . The results are displayed in Fig. 4. Obviously, MoAL-LRTF is the best performer. Though MoG-LRTF performs well, the residual image between MoAL- and MoG-LRTF shows that MoG-LRTF cannot remove the shadow of the car (see the boxed region in Fig. 4(g)).

IV. CONCLUSION

We propose a novel model MoAL-LRTF for skew data analysis. The core idea is to model the unknown noise with an MoAL and to infer the model with the EM algorithm. The experiments conducted with both synthetic and real-world datasets show that MoAL-LRTF is a competitive tool to handle skew data with low-rank structure.

REFERENCES

- [1] N. D. Sidiropoulos, L. D. Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017. [Online]. Available: <https://doi.org/10.1109/TSP.2017.2690524>
- [2] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: An overview of methods, challenges, and prospects," *Proc. IEEE*, vol. 103, no. 9, pp. 1449–1477, Sep. 2015. [Online]. Available: <https://doi.org/10.1109/JPROC.2015.2460697>
- [3] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 2862–2869. [Online]. Available: <https://doi.org/10.1109/CVPR.2014.366>
- [4] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.
- [5] H. A. Kiers, "Towards a standardized notation and terminology in multiway analysis," *J. Chemometrics*, vol. 14, no. 3, pp. 105–122, 2000.
- [6] X. Chen *et al.*, "A generalized model for robust tensor factorization with noise modeling by mixture of gaussians," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5380–5393, Nov. 2018. [Online]. Available: <https://doi.org/10.1109/TNNLS.2018.2796606>
- [7] M. Zhang, Y. Gao, C. Sun, J. L. Salle, and J. Liang, "Robust tensor factorization using maximum correntropy criterion," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 4184–4189. [Online]. Available: <https://doi.org/10.1109/ICPR.2016.7900290>
- [8] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S. Amari, "Bayesian robust tensor factorization for incomplete multiway data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 736–748, Apr. 2016. [Online]. Available: <https://doi.org/10.1109/TNNLS.2015.2423694>
- [9] H. Huang and C. H. Q. Ding, "Robust tensor factorization using R1 norm," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2008, pp. 1–8. [Online]. Available: <https://doi.org/10.1109/CVPR.2008.4587392>
- [10] D. Meng, B. Zhang, Z. Xu, L. Zhang, and C. Gao, "Robust low-rank tensor factorization by cyclic weighted median," *Sci. China Inf. Sci.*, vol. 58, no. 5, pp. 1–11, 2015. [Online]. Available: <https://doi.org/10.1007/s11432-014-5223-4>
- [11] D. Meng and F. De la Torre, "Robust matrix factorization with unknown noise," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 1337–1344. [Online]. Available: <https://doi.org/10.1109/ICCV.2013.169>
- [12] V. Maz'ya and G. Schmidt, "On approximate approximations using Gaussian kernels," *IMA J. Numer. Anal.*, vol. 16, no. 1, pp. 13–29, 1996.
- [13] X. Cao, Q. Zhao, D. Meng, Y. Chen, and Z. Xu, "Robust low-rank matrix factorization under general mixture noise distributions," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4677–4690, Oct. 2016. [Online]. Available: <https://doi.org/10.1109/TIP.2016.2593343>
- [14] P. Chen, N. Wang, N. L. Zhang, and D. Yeung, "Bayesian adaptive matrix factorization with automatic model selection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1284–1292. [Online]. Available: <https://doi.org/10.1109/CVPR.2015.7298733>
- [15] S. Xu, C. Zhang, and J. Zhang, "Adaptive quantile low-rank matrix factorization," *Pattern Recognit.*, vol. 103, 2020, Art. no. 107310. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S003132032030114X>
- [16] R. Zhu, D. Niu, L. Kong, and Z. Li, "Expectile matrix factorization for skewed data analysis," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 259–266. [Online]. Available: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/15029>
- [17] K. Yu and J. Zhang, "A three-parameter asymmetric Laplace distribution and its extension," *Commun. Statist. - Theory Methods*, vol. 34, no. 9–10, pp. 1867–1879, 2005. [Online]. Available: <https://doi.org/10.1080/03610920500199018>
- [18] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics Intell. Lab. Syst.*, vol. 106, no. 1, pp. 41–56, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0169743910001437>
- [19] B. W. Bader and T. G. Kolda, "Algorithm 862: MATLAB tensor classes for fast algorithm prototyping," *ACM Trans. Math. Softw.*, vol. 32, no. 4, pp. 635–653, 2006. [Online]. Available: <https://doi.org/10.1145/1186785.1186794>