Robust Asymmetric Bayesian Adaptive Matrix Factorization

Xin Guo **Boyuan Pan** **Deng Cai*** **Xiaofei He**
State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, China
guoxinzju@gmail.com panby@zju.edu.cn {dengcai, xiaofeihe}@cad.zju.edu.cn

Abstract

Low rank matrix factorizations(LRMF) have attracted much attention due to its wide range of applications in computer vision, such as image impainting and video denoising. Most of the existing methods assume that the loss between an observed measurement matrix and its bilinear factorization follows symmetric distribution, like gaussian or gamma families. However, in real-world situations, this assumption is often found too idealized, because pictures under various illumination and angles may suffer from multi-peaks, asymmetric and irregular noises. To address these problems, this paper assumes that the loss follows a mixture of Asymmetric Laplace distributions and proposes robust Asymmetric Laplace Adaptive Matrix Factorization model(ALAMF) under bayesian matrix factorization framework. The assumption of Laplace distribution makes our model more robust and the asymmetric attribute makes our model more flexible and adaptable to real-world noise. A variational method is then devised for model inference. We compare ALAMF with other state-of-the-art matrix factorization methods both on data sets ranging from synthetic and real-world application. The experimental results demonstrate the effectiveness of our proposed approach.

1 Introduction

Low rank matrix factorization is one of the most popular methods for subspace learning. It uses the product of a basis matrix and a coefficient matrix under some criteria to approximate a given data matrix. Matrix factorization can be regarded as an efficient technique to reveal the low-dimensional structure of the data when the underlying rank of the two factor matrices is lower than that of the original data matrix.

Under the assumption of Gaussian noises, it is natural to utilize the L_2 norm as the noise measure, which has been extensively studied in LRMF literatures [Mitra *et al.*, 2010; Okatani *et al.*, 2011]. However, these methods are sensitive to outliers and non-Gaussian noises. In order to introduce robustness, the L_1 norm-based models have attracted much attention [Ji *et al.*, 2010; Shu *et al.*, 2014]. But the L_1 norm is only optimal for Laplace-like noises and still very limited for handling various types of noises encountered in real problems. Recently, some novel models were presented to expand the availability of LRMF under more complex noise. [Meng and De La Torre, 2013] proposed MoG (Mixture of Gaussian) to fit the noise and further it is extended into a full Bayesian model by [Chen *et al.*, 2015] and to RPCA by [Zhao *et al.*, 2014]. Later on, [Cao *et al.*, 2015] assumed noise as a more general mixture of exponential power distribution and proposed a more robust PMoEP method.

Almost all of the existing methods use symmetric distribution to fit noise. However, in many cases this is just for simplifying the model, and the noise is more likely to be asymmetric in reality. In real images, there are different types of noise sources [Meng and De La Torre, 2013]. First, there are cast shadows, so the usual Lambertian surface assumption [Smith *et al.*, 1980] is invalid. Second, due to the camera range settings there might be pixels that are saturated and there exist specular reflections (especially in people with glasses). Third, the camera noise is amplified in the dark areas. So picture under varying illumination may suffer from multi-peaks, asymmetric and irregular noises. All the existing methods based on MoG can approximate an asymmetric distribution at an expensive price. A simple situation can be considered for clear explanation. When the noises follow single component asymmetric distribution, MoG should first fit them with a large component, then fit the rest with a smaller one, and so on. This process is similar to series expansion, which leads to infinite components. It is similar when the situation is promoted to the general. So a more direct method for modeling asymmetric noise should be considered.

In this paper, we propose a novel robust Asymmetric Laplace Adaptive Matrix Factorization model under bayesian matrix factorization framework. We assume the loss follows a mixture of Asymmetric Laplace distributions. A variational method is then devised for model inference. To the best of our knowledge, this is the first low-rank matrix factorization work which takes asymmetric distribution into consideration. Experimental results on synthetic and real-world data sets demonstrate the effectiveness of our model.

The rest of the paper is organized as follows: Section 2 provides related work regarding ALAMF. Some preliminary

*Corresponding author

knowledges are introduced in Section 3. The ALAMF model and corresponding variational inference algorithm are presented in Section 4. Experimental results are shown in Section 5. Finally, concluding marks are provided in Section 6.

2 Related Work

Improving model robustness in matrix factorization is a hot research in machine learning for decades. Several privious works such as [Croux and Filzmoser, 1998; Ke and Kanade, 2005] are good explorations for matrix factorization but unappealing for large-scale applications. Recently, [Candès *et al.*, 2011] proposed principal component pursuit (PCP) and [Zhou *et al.*, 2010] proposed stable principal component pursuit (SPCP), which utilize the nuclear norm for normalization and can be regarded as a breakthrough in this research topic. [Qian *et al.*, 2016] encodes features correlation by Wasserstein distance to improve robustness. The convexity of the L_1 and nuclear norms enables the application of efficient convex program solvers [Lin *et al.*, 2010].

There are some other efficient methods like probabilistic algorithms. The most representative models for (non-robust) matrix factorization are PMF [Salakhutdinov and Mnih, 2007] and BPMF [Salakhutdinov and Mnih, 2008]. Based on Students t-distribution, [Lakshminarayanan *et al.*, 2011] proposed a robust extension of BPMF for collaborative filtering. Another recent attempt is PRMF [Wang *et al.*, 2012] which uses the expectation-maximization (EM) algorithm. The highlights of PRMF are its high efficiency and online extension. Inspired by the work of Bayesian robust PCA (BRPCA) [Ding *et al.*, 2011] and variational Bayesian low-rank factorization (VBLR) [Babacan *et al.*, 2012], PRMF has been later extended to fully Bayesian settings (BRMF and MBRMF) by [Wang and Yeung, 2013]. Subspace-MoG [Meng and De La Torre, 2013] modeled the noise as a MoG distribution for LRMF and was extended to MoEP distribution by [Cao *et al.*, 2015] and to MRPCA by [Zhao *et al.*, 2014]. The previous work that is most closely related to ours are AMF [Chen *et al.*, 2015], which is an extension from subspace-MoG to the Bayesian framework.

3 Preliminary

3.1 Notations

We introduce some notations for probability distributions. $\mathcal{N}(\mu, \sigma)$ denotes the univariate Gaussian distribution with mean μ and variance σ, $\mathcal{B}(\alpha, \beta)$ the beta distribution with parameters α and β, $IG(\alpha, \beta)$ the inverse-gamma distribution with shape parameter α and the scale parameter β, $Multi(\theta)$ the multinomial distribution, $GEM(\alpha)$ the stick-breaking distribution, and $Ray(\mu, \sigma)$ the Rayleigh distribution with the location parameter μ and the scale parameter σ.

3.2 Asymmetric Laplace Distribution

We firstly give out the definition of Asymmetric Laplace distribution.

Definition 1. A random variable has an $Asymmetric\ Laplace$ distribution (denoted as $\mathcal{AL}(\mu, \sigma, \tau)$), if its probability density function is

$$f(x;\mu,\sigma,\tau) = \frac{\tau(1-\tau)}{\sigma} exp(-\frac{1}{2}|\frac{x-\mu}{\sigma}|+(\tau-\frac{1}{2})(\frac{x-\mu}{\sigma})) \tag{1}$$

or alternatively:

$$f(x;\mu,\sigma,\tau) = \frac{\tau(1-\tau)}{\sigma} \begin{cases} exp(\tau\frac{x-\mu}{\sigma}) & x \le \mu \\ exp((\tau-1)\frac{x-\mu}{\sigma}) & x > \mu \end{cases} \tag{2}$$

Here, μ is a location parameter, $\sigma > 0$, is a scale parameter, and τ is an asymmetry parameter. When $\tau = \frac{1}{2}$, the distribution degenerates to the Laplace distribution.

For computational efficiency, Asymmetric Laplace distribution is expressed as a composition of two simple distributions, which is described as follow.

Theorem 1. Let x and a be random variables such that

$$x|a \sim \mathcal{N}(\mu + \frac{(\frac{1}{2}-\tau)\sigma}{a\tau(1-\tau)}, \frac{\sigma^2}{a\tau(1-\tau)})$$

and $$a \sim IG(1, \tfrac{1}{2})$$

Then

$$x \sim \mathcal{AL}(\mu, \sigma, \tau)$$

Theorem 1 follows from Result2 of [Wand *et al.*, 2011]. Theorem 1 is the footstone of our model solving. The direct use of asymmetric Laplacian distribution in probabilistic graphical model will lead to model insolvability.

The following integral families comprise the full set of non-analytic integrals which arise in the models considered in this paper.

Definition 2. Non-analytic Integral Families

$$\mathcal{J}^+(A,B,C) = \int_0^\infty x^A exp(Bx - Cx^2)dx, \tag{3}$$

$$\mathcal{L}^+(A,B,C) = \int_0^\infty \log(x) x^A exp(Bx - Cx^2)dx. \tag{4}$$

where, $A \ge -1, -\infty < B < \infty, C > 0$

[Wand *et al.*, 2011] discusses the stable and efficient computation of $\mathcal{J}^+(A,B,C)$. The similar properties of $\mathcal{L}^+(A,B,C)$ are omitted due to limited space.

4 ALAMF

In this section, we first present the graphical model and the generative process of our robust Asymmetric Laplace Adaptive Matrix Factorization(ALAMF) model. We then present details of the model inference.

4.1 Bayesian Model

We assume that the noise follows:

$$p(\epsilon_{mn}) = \sum_{k=1}^{\infty} \theta_k \mathcal{AL}(\epsilon_{mn}|0, \sigma_k, \tau_k) \tag{5}$$

where the mixing proportion of each $\mathcal{AL}$ component is obtained from the stick-breaking process. As a consequence,

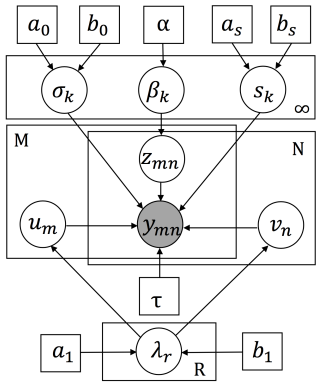


Figure 1: Graph model of ALAMF

the noise entries will cluster themselves into K groups without the need for a complicated selection procedure.

According to Theorem 1, we decompose Asymmetric Laplace distribution as a composition of Gaussian distribution and inverse-gamma distribution. This produces the component s_k in our Graphical model. As we can see, the new coming component s_k has a forward influence on the location parameter value (U_m and V_n) and the scale parameter (σ_k) of y_{mn}. U_m, V_n and σ_k then have feedback on s_k due to the iterative update in model inference. The rate of asymmetry is controlled by asymmetric hyper-parameter τ, which is a directed impact factor on almost all the components of our model.

The overall generative process is summarized as follows:
1. Draw component mixing proportions $\theta \sim GEM(\alpha)$
2. For each cluster k of noise:
   - Draw variance $\sigma_k^2 \sim IG(a_0, b_0)$
   - Draw $s_k \sim IG(a_s, b_s)$, where $a_s = 1, b_s = \frac{1}{2}$
3. For each dimension r of U and V(i.e. for each column of U and V):
   - Draw variance $\lambda_r \sim IG(a_1, b_1)$
4. For each element in U and V:
   - Draw $u_{mr}, v_{nr} \sim \mathcal{N}(0, \lambda_r)$
5. For each data element y_{mn}:
   - Draw noise cluster label $z_{mn} \sim Multi(\theta)$
   - Draw observation

$$y_{mn} \sim \mathcal{N}(u_{m\cdot}v_{n\cdot}^T + \frac{(\frac{1}{2}-\tau)\sigma_{z_{mn}}}{s_k\tau(1-\tau)}, \frac{\sigma_{z_{mn}}^2}{s_k\tau(1-\tau)})$$

Here $\theta_k = \beta_k \prod_{l=1}^{k-1}(1-\beta_l)$ and β_k is drawn indepently from $\mathcal{B}(1,\alpha)$ according to the stick-breaking construction. Based on the generative process, the joint distribution can be expressed as:

$$\begin{aligned} &p(U,V,Y,z,\sigma,\lambda,\beta,s) \\ &= p(Y|U,V,z,\sigma,s)p(U|\lambda)p(V|\lambda)p(\lambda|a_1,b_1) \\ &\quad p(\sigma|a_0,b_0)p(z|\beta)p(\beta|\alpha)p(s|a_s,b_s) \end{aligned} \tag{6}$$

4.2 Model Inference

Since all the distributions in our model belong to the exponential family, we can take advantage of the efficient variational inference.

Based on the mean-field variational approach, we devise the following variational distribution:

$$\begin{aligned} q(U,V,z,s,\sigma,\lambda,\beta) = &\prod_{m=1}^{M} q(u_{m\cdot}) \prod_{n=1}^{N} q(v_{n\cdot}) \prod_{r=1}^{R} q(\lambda_r) \\ &\prod_{k=1}^{K} q(\beta_k) \prod_{k=1}^{K} q(\sigma_k) \prod_{m=1}^{M}\prod_{n=1}^{N} q(z_{mn}) \prod_{k=1}^{K} q(s_k) \end{aligned} \tag{7}$$

The optimization problem of minimizing the KL divergence is equivalent to maximizing the following lower bound:

$$\mathcal{L} = E[log\, p(U,V,Y,z,\sigma,\lambda,\beta,s) - log\, q(U,V,z,s,\sigma,\lambda,\beta)] \tag{8}$$

By solving (8), we get the update rules as follow:
Update β_k, σ_k, z_{mn} and s_k

- For β_k

$$q_{\beta_k}(\gamma_{k,1},\gamma_{k,2}) \propto \beta_k^{\gamma_{k,1}-1}(1-\beta_k)^{\gamma_{k,2}-1} \tag{9}$$

It is easy to see that $q_{\beta_k}(\gamma_{k,1},\gamma_{k,2})$ is a beta distribution, and $\gamma_{k,1},\gamma_{k,2}$ are the positive real shape parameters.

$$\begin{aligned} \gamma_{k,1} &= 1 + \sum_{(m,n)\in\Omega} \phi_{mnk} \\ \gamma_{k,2} &= \alpha + \sum_{(m,n)\in\Omega} \sum_{t=k+1}^{K} \phi_{mnt} \end{aligned} \tag{10}$$

Here ϕ_{mnk} is defined in Eq(14).

- For σ_k

$$q(\sigma_k^2) \propto (\sigma_k)^{-A} exp(B\sigma_k^{-1} - C\sigma_k^{-2}) \tag{11}$$

where

$$\begin{aligned} A &= 2a_0 + 2 + \sum_{(m,n)\in\Omega} \phi_{mnk} \\ B &= (\frac{1}{2}-\tau) \sum_{(m,n)\in\Omega} \phi_{mnk} E(y_{mn} - u_{m\cdot}v_{n\cdot}^T) \\ C &= b_0 + \frac{\tau(1-\tau)}{2} E(s_k) \sum_{(m,n)\in\Omega} \phi_{mnk} E(y_{mn} - u_{m\cdot}v_{n\cdot}^T)^2 \end{aligned} \tag{12}$$

Three important estimators used in this paper are listed as follows:

$$\begin{aligned} &E(\sigma_k^{-1}) = \mathcal{J}^+(A-2,B,C)/\mathcal{J}^+(A-3,B,C) \\ &E(\sigma_k^{-2}) = \mathcal{J}^+(A-1,B,C)/J^+(A-3,B,C) \\ &E(\log(\sigma_k^{-1})) = \mathcal{L}^+(A-3,B,C)/\mathcal{J}^+(A-3,B,C) \end{aligned} \tag{13}$$

- For z_{mn}

For convenience, we use ϕ_{mnk} to denote the probability $q(z_{mn}=k)$.

$$\begin{aligned}\phi_{mnk} \propto exp(&E(\log\beta_k)+\sum_{t=1}^{k-1}E(\log(1-\beta_t))\\&-\frac{1}{2}\tau(1-\tau)E(\sigma_k^{-2})E(s_k)E(y_{mn}-u_{m\cdot}v_{n\cdot}^T)^2\\&+(\frac{1}{2}-\tau)E(\sigma_k^{-1})E(y_{mn}-u_{m\cdot}v_{n\cdot}^T)\\&-\frac{(\frac{1}{2}-\tau)^2}{2\tau(1-\tau)}E(s_k^{-1})+E(\log(\sigma_k^{-1}))+\frac{1}{2}E(\log(s_k)))\end{aligned} \tag{14}$$

- For s_k

$$q(s_k) \propto s^{\gamma-1}exp(-\frac{1}{2}(\alpha s+\beta s^{-1}) \tag{15}$$

where,

$$\begin{aligned}\alpha &= \tau(1-\tau)E(\sigma_k^{-2})\sum_{(m,n)\in\Omega}\phi_{mnk}E(y_{mn}-u_{m\cdot}v_{n\cdot}^T)^2\\ \beta &= 1+\frac{(\frac{1}{2}-\tau)^2}{\tau(1-\tau)}\sum_{(m,n)\in\Omega}\phi_{mnk}\\ \gamma &= \frac{1}{2}\sum_{(m,n)\in\Omega}\phi_{mnk}-1\end{aligned} \tag{16}$$

It is easy to see that $q(s_k)$ follows generalized inverse Gaussian (GIG) distribution. Three important estimators used in this paper are listed as follows:

$$\begin{aligned}E(s_k) &= \frac{\sqrt{\beta}K_{\gamma+1}(\sqrt{\alpha\beta})}{\sqrt{\alpha}K_\gamma(\sqrt{\alpha\beta})}\\ E(s_k^{-1}) &= \frac{\sqrt{\alpha}K_{\gamma+1}(\sqrt{\alpha\beta})}{\sqrt{\beta}K_\gamma(\sqrt{\alpha\beta})}-\frac{2\gamma}{\beta}\\ E(\log(s_k)) &= \log\frac{\sqrt{\beta}}{\sqrt{\alpha}}+\frac{\partial}{\partial\gamma}\log K_\gamma(\sqrt{\alpha\beta})\end{aligned} \tag{17}$$

where $K_p(x)$ is a modified Bessel function of the second kind.

Update λ_r, $U_{m\cdot}$ and $V_{n\cdot}$

- For λ_r

$$q(\lambda_r) \propto \lambda_r^{-\eta_{r,1}-1}exp(-\frac{\eta_{r,2}}{\lambda_r}) \tag{18}$$

It is easy to see that $q(\lambda_r)$ follows inverse-gamma distribution, where $\eta_{r,1}, \eta_{r,2}$ are the positive real shape parameters.

$$\begin{aligned}\eta_{r,1} &= a_1+\frac{M+N}{2}\\ \eta_{r,2} &= b_1+\frac{1}{2}(a_{\cdot r}^Ta_{\cdot r}\sum_{m-1}^{M}(\Sigma_m^u)_{rr}+b_{\cdot r}^Tb_{\cdot r}\sum_{n-1}^{N}(\Sigma_n^v)_{rr})\end{aligned} \tag{19}$$

- For $U_{m\cdot}, V_{n\cdot}$

Algorithm 1 Variational Inference for ALAMF

Input: Data $Y=\{y_{ij}\}_{mn}$, compotent number K, maximum rank R and asymmetric parameter τ
Repeat:
  for each cluster k of noise **do**
    Update β_k, σ_k, ϕ_{mnk} and s_k by (10)(13)(14)(17)
    Remove insignificant mixture components
  end for
  Update λ_r by (19) for each rank r
  Update $U_{m\cdot}$ and $V_{n\cdot}$ by(21)(22)
  Shrink rank
until converse
Output: $U_{m\cdot}$ and $V_{n\cdot}$

$$q(U_{m\cdot}) \propto exp(-\frac{1}{2}(U_{m\cdot}-a_{m\cdot})^T(\Sigma_m^u)^{-1}(U_{m\cdot}-a_{m\cdot}) \tag{20}$$

Each row of U follows a Gaussian distribution with mean $a_{m\cdot}^T$and covariance Σ_m^u, where,

$$\begin{aligned}a_{m\cdot}^T = \Sigma_m^u\cdot(&\sum_{n:(m,n)\in\Omega}\sum_{k=1}^{K}[\tau(1-\tau)E(\sigma_k^{-2})E(s_k)y_{mn}\phi_{mnk}\\&-(\frac{1}{2}-\tau)E(\sigma_k^{-1})\phi_{mnk}]b_{n\cdot}^T)\end{aligned} \tag{21}$$

$$\begin{aligned}\Sigma_m^u = [\tau(1-\tau)&\sum_{n:(m,n)\in\Omega}\sum_{k=1}^{K}E(\sigma_k^{-2})E(s_k)\phi_{mnk}(b_{n\cdot}^Tb_{n\cdot}\\&+\Sigma_n^v)+\Lambda]^{-1}\end{aligned} \tag{22}$$

where, $\Lambda = diag(\lambda)^{-1}$

The update for $b_{n\cdot}$ and Σ_n^v is similar.

By repeating the update steps above, we finally get the estimation of U and V corresponding to $a_{m\cdot}$ and $b_{n\cdot}$. The algorithm of ALAMF is summarized in Algorithm 1. The details of insignificant mixture components removing and rank shrinking in Algorithm 1 can be found in [Chen *et al.*, 2015].

5 Experiments

In this section, we empirically compare the proposed ALAMF model with seven state-of-the-art methods. There are non-Bayesian methods (PMoEP [Cao *et al.*, 2015], CWM [Meng *et al.*, 2013], PCP [Candès *et al.*, 2011]) and Bayesian methods (AMF [Chen *et al.*, 2015], MBRMF [Wang and Yeung, 2013], VBLR [Babacan *et al.*, 2012], MRPCA [Zhao *et al.*, 2014]). Note that PCP, MBRMF and MRPCA are not designed to handle missing data, thus we replace MRPCA with its previous version MoG [Meng and De La Torre, 2013]. For all the experiments we have conducted, the hyperparameters of ALAMF are fixed without further tuning: $a_0=b_0=10^{-4}, a_1=b_1=0.1, \alpha=1$.

5.1 Synthetic Experiments

In this section, we follow [Chen *et al.*, 2015; Cao *et al.*, 2015] and design three sets of synthetic experiments to compare the

performance of all the above low-rank matrix factorization methods. For each of experiment, we generate 30 ground-truth low-rank matrices. Each of them was generated by the multiplication of two randomly generated low-rank matrices $U_{gt} \in R^{50\times4}$ and $V_{gt} \in R^{50\times4}$, i.e.$Y_{gt} = U_{gt}V_{gt}^T$. Each element of U and V follows Gaussion distribution $\mathcal{N}(0,1)$.

We add different types of noises in the non-missing entries as follows: (1) Gaussian Noise: all of the entries were corrupted with $\mathcal{N}(0,0.5^2)$ (2) Rayleigh Noise: all of the entries were corrupted with $Ray(-2,2)$ (3) Mixture Noise: 15% of the entries were corrupted with Gaussian Noise $\mathcal{N}(0.05,0.5^2)$, 20% are contaminated with uniform distribution noise $U[-5,5]$ and 20% were contaminated with uniform distribution noise $U[-2,2]$. Further, we randomly specify 20% of entries in Y_{gt} as missing data.

We denoted the noisy matrix as Y_{no}. Six measures were utilized for performance assessment:
$E1 = ||W \odot (Y_{no} - \tilde{U}\tilde{V}^T)||_1, E2 = ||W \odot (Y_{no} - \tilde{U}\tilde{V}^T)||_2,$
$E3 = ||(Y_{gt} - \tilde{U}\tilde{V}^T)||_1, E4 = ||(Y_{gt} - \tilde{U}\tilde{V}^T)||_2,$
$E5 = subspace(U_{gt}, \tilde{U}), E6 = subspace(V_{gt}, \tilde{V})$

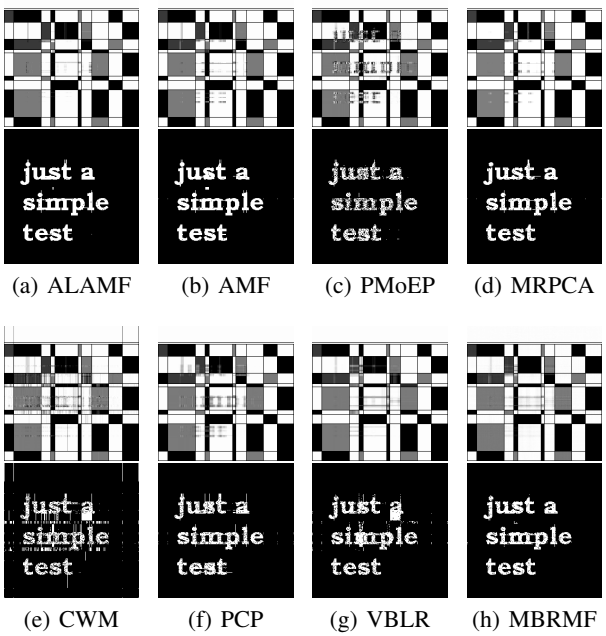
where $\tilde{U}, \tilde{V}$ are the recovered low-rank matrices, and $subspace(U_1, U_2)$ denotes the angle between subspaces spanned by the columns of U_1 and U_2. Note that E1 and E2 are the optimization objective function for L_1 and L_2 norm LRMF problems, while the latter four measures are more faithful to evaluate whether the method recovers the correct subspaces.

We alleviate the local optimum issue by means of the multiple random initialization strategy. For all the methods, we first run with 20 random initializations for each generated matrix and then select the best result with respect to the objective value. Finally, we select the initialization with the largest likelihood value as the result for this generated matrix. The performance of each method on each simulation is evaluated as the average results over the 30 random generated matrices in terms of the six measures, and the results are summarized in Table 1. For all the methods, we set the rank of the low-rank component to 8 and apply the random initialization strategy to U and V. For ALAMF, we choose asymmetric parameter τ uniformly from 0.2 to 0.8 and record the corresponding results on different datasets. Specifically, ALAMF degenerates as symmetric form when τ = 0.5, and then we denote it as LAMF. Finally, we use 6 measures to show the performance of ALAMF whose τ makes the best general performance.

From Table 1, when given a simple noise type (Gaussian), ALAMF achieves the best when τ = 0.5, and it shares the same result with LAMF. In this case, ALAMF(LAMF) has a little improvement from previous methods. When given asymmetric noise or complex noise, ALAMF and LAMF perform much better in most measures than other methods, which means our model is extremely adept at dealing with asymmetric distribution. What's more, ALMF can still have a good result even when 20% of the input entries are corrupted.

5.2 Text Removal

In this part, we follow [Wang and Yeung, 2013] and [Chen *et al.*, 2015] to conduct a text removal simulation experiment. This task is to remove some text embedded in an image which has a certain pattern as background. The size of the clean image is set to 256×256 with the corresponding data matrix of rank 10. As the true rank is often unknown beforehand in real-world data, we set the initialized rank of the initial U and V to be twice of the true rank for all the algorithms. For all the methods, we first run with 20 random initializations and then select the best result with respect to the objective value. For our ALAMF, τ is uniformly chosen from 0.2 to 0.8, and the best result is reported.



(a) ALAMF (b) AMF (c) PMoEP (d) MRPCA

(e) CWM (f) PCP (g) VBLR (h) MBRMF

Figure 2: Background and foreground masks recovered by different algorithms.

From foreground (mask) it is obviously that ALAMF, AMF, MRPCA and MBRMF are better than other methods. Furthermore, the results on the background show that ALAMF does the best job in denoising. According to Table 2, which is the quantitative analysis, ALAMF has the highest RE record and has a competitive result on AUC.

5.3 Face Reconstruction Experiments

Similar to [Meng *et al.*, 2013], we study a real application using face images captured under varying illumination. We generate some relatively large datasets and some relatively small datasets in the experiments.

Firstly, a larger dataset was built by using the first and fifth subsets of Extended Yale B datasets(Georghiades, Belhumeur and Kriegman 2001;Basri and Jacobs 2003). For each person, we use all the 64 images in the datasets. We use the original face as input and compare all the eight methods.

Further, for a smaller dataset, we downsample the images to 48×42 and set (0%,10%), (10%,10%), (20%,10%), (30%,10%) of the randomly selected pixels of each image in the first and fifth subsets as (missing entries and salt-pepper noise), respectively. We random sample 16 faces of each subset and then two data matrices of dimension 2016×16 are

	ALAMF	LAMF	AMF	PMoEP	MRPCA/MoG	CWM	VBLR	PCP	MBRMF
Gaussian Nosie									
E1	-	4.58e+2/3.58e+2	4.61e+2/3.58e+2	**3.98e+2**/2.87e+2	4.59e+2/**2.82e+2**	4.15e+2/3.14e+2	4.57e+2/3.59e+2	4.23e+2	4.01e+2
E2	-	1.32e+2/1.02e+2	1.34e+2/1.01e+2	**9.99e+1**/6.59e+1	1.33e+2/**6.36e+1**	1.38e+2/1.05e+2	1.74e+2/1.01e+2	1.42e+2	1.23e+2
E3	-	**2.13e+1/3.04e+1**	2.47e+1/3.21e+1	6.70e+1/2.10e+2	2.52e+1/9.72e+1	5.46e+1/8.10e+1	6.44e+1/3.28e+1	7.56e+1	5.09e+1
E4	-	**1.79e+2/2.11e+2**	1.92e+2/2.17e+2	3.18e+2/4.07e+2	1.95e+2/3.71e+2	2.86e+2/3.38e+2	3.10e+2/2.19e+2	3.38e+2	2.79e+2
E5	-	**4.45e-2/5.54e-2**	5.22e-2/5.99e-2	5.08e-2/6.33e-2	5.25e-2/5.82e-2	5.86e-2/6.23e-2	8.47e-2/5.82e-2	-	5.89e-2
E6	-	**4.23e-2**/6.15e-2	4.89e-2/**5.80e-2**	4.78e-2/6.24e-2	4.90e-2/5.86e-2	5.86e-2/6.51e-2	8.09e-2/5.87e-2	-	5.71e-2
Rayleigh Noise									
E1	2.53e+3/1.02e+3	2.36e+3/**7.40e+2**	2.38e+3/9.37e+2	**1.98e+3**/9.15e+2	2.55e+3/7.49e+2	2.04e+3/7.97e+2	2.53e+3/9.88e+2	2.16e+3	1.99e+3
E2	4.31e+3/8.62e+2	3.61e+3/6.71e+2	3.63e+3/6.99e+2	**2.90e+3**/5.07e+2	4.48e+3/**4.48e+2**	3.65e+3/7.27e+2	5.46e+3/8.02e+2	3.96e+3	3.66e+3
E3	**5.93e+2/2.04e+2**	1.12e+3/2.62e+2	1.06e+3/3.37e+2	1.70e+3/7.51e+2	6.36e+2/8.64e+2	2.01e+3/5.85e+2	1.84e+3/2.54e+2	1.55e+3	2.20e+3
E4	**9.44e+2/5.41e+2**	1.09e+3/6.24e+2	1.29e+3/7.10e+2	1.60e+3/1.06e+3	9.73e+2/1.08e+3	1.75e+3/9.21e+2	1.65e+3/6.15e+2	1.54e+3	1.84e+3
E5	2.48e-1/**1.46e-1**	**2.43e-1**/1.53e-1	2.67e-1/1.50e-1	2.46e-1/1.49e-1	2.47e-1/1.51e-1	3.23e-1/1.57e-1	4.90e-1/1.60e-1	-	2.99e-1
E6	**2.41e-1**/1.77e-1	2.59e-1/1.51e-1	2.66e-1/1.80e-1	2.56e-1/**1.49e-1**	2.92e-1/1.52e-1	3.31e-1/1.71e-1	5.07e-1/1.65e-1	-	2.88e-1
Mixture Noise									
E1	1.83e+3/1.49e+3	1.83e+3/1.49e+3	1.82e+3/1.51e+3	1.79e+3/1.48e+3	1.83e+3/1.42e+3	**1.77e+3/1.38e+3**	1.83e+3/1.74e+3	1.83e+3	1.72e+3
E2	4.85e+3/**9.27e+2**	**8.50e+2**/2.87e+3	4.82e+3/3.96e+3	4.78e+3/3.60e+3	4.82e+3/1.92e+3	4.09e+3/3.02e+3	4.80e+3/3.26e+3	4.53e+3	4.27e+3
E3	**9.01e-1/2.09e+1**	9.55e-1/2.54e+1	6.24e+0/2.52e+1	3.80e+2/9.50e+2	6.43e+0/6.89e+3	4.59e+2/9.87e+2	1.68e+1/9.80e+2	2.71e+2	1.66e+2
E4	**7.16e+0/3.45e+1**	7.60e+0/4.47e+1	1.51e+1/4.64e+1	5.08e+2/9.51e+2	2.07e+1/2.15e+3	6.56e+2/9.96e+2	5.08e+1/1.21e+3	5.11e+2	3.73e+2
E5	**1.65e-2**/5.97e-2	1.70e-2/1.43e-2	2.39e-2/**1.42e-2**	9.15e-2/1.49e-1	2.35e-2/3.329e-1	1.42e-1/1.94e-1	5.33e-2/3.28e-1	-	8.01e-2
E6	**9.07e-3/3.02e-2**	1.19e-2/7.28e-2	2.33e-2/9.22e-2	8.38e-2/1.53e-1	2.35e-2/3.249e-1	1.52e-1/1.96e-1	4.16e-2/3.12e-1	-	8.48e-2

Table 1: Performance evaluation on synthetic data. For each item, results with full matrices are showed on the left of slash and results with missing data are showed on the right. The best results in terms of the six criteria are highlighted in bold. The best result of ALAMF under Gaussian noise is observed when τ is set to 0.5. We omit the result of ALAMF in this situation, because it shares the same result with LAMF. Notice that the six measures on ALAMF share the same τ for each kind of noise.

	ALAMF	AMF	PMoEP	MRPCA	CWM	VBLR	PCP	MBRMF
AUC	0.9956	**0.9958**	0.9840	0.9829	0.9759	0.9867	0.9823	0.9949
RE	**0.0671**	0.0688	0.1717	0.0886	0.1569	0.1448	0.1102	0.0740

Table 2: Comparison of different methods on text removal simulation experiment.

Figure 3: Face shadow removal results. Each of the nine groups from left to right shows the original face and those recovered by ALAMF, AMF, PMoEP, MRPAC, CWM, VBLR, PCP and MBRMF.

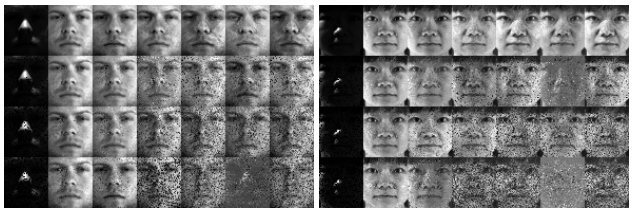


Figure 4: Face shadow removal results with input corrupted. Each of the seven groups from left to right shows the original face and those recovered by ALAMF, AMF, PMoEP, MoG, CWM and VBLR. The number of missing entities is increasing from top to bottom.

formed. Since only ALAMF, AMF, PMoEP, MoG, CWM and VBLR are claimed to be able to handle missing data, we give the result of the six algorithms.

In Figure 3 and Figure 4, we show the result of all the compared method on large face datasets and small datasets. We set the rank to 8 and adopt random initialization strategy for all methods. For our ALAMF, τ is uniformly chosen from 0.2 to 0.8, and the best result is reported.

According to figure 3, MRPCA, CWM, PCP and MBRMF still have some residual shadow and VBLR suffers from losing detail information of faces. It is hard to distinguish ALAMF, AMF, PMoEP by naked eyes, and all of them can not only remove shadow, but also keep the details of faces.

Moreover, on more challenging small datasets, we can observe from figure 4 that from top to bottom, PMoEP, MoG, CWM and VBLR can not achieve satisfactory results on recovering images with the increasing of missing data. However, ALAMF and AMF perform much better in most cases and ALAMF can be in the lead when the noise is stronger.

6 Conclusion

In this paper, we propose a novel robust non-parametric Bayesian method for matrix factorization. Firstly, due to the use of Laplace distribution assumption on the residual error, our model is more effective. Secondly, asymmetric distribution is a better fit to noise in real-world, which makes our method more reasonable and adaptable. What's more, the decomposability of Asymmetric Laplace distribution into two simple distributions brings solvable model inference.

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program) under Grant 2013CB336500 and National Natural Science Foundation of China under Grant 61233011.

References

[Babacan *et al.*, 2012] S. Derin Babacan, Martin Luessi, Rafael Molina, and Aggelos K. Katsaggelos. Sparse bayesian methods for low-rank matrix estimation. *IEEE Transactions on Signal Processing*, 60(8):3964–3977, 2012.

[Candès *et al.*, 2011] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[Cao *et al.*, 2015] Xiangyong Cao, Yang Chen, Qian Zhao, Deyu Meng, Yao Wang, Dong Wang, and Zongben Xu. Low-rank matrix factorization under general mixture noise distributions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1493–1501. IEEE, 2015.

[Chen *et al.*, 2015] Peixian Chen, Naiyan Wang, Nevin L. Zhang, and Dit-Yan Yeung. Bayesian adaptive matrix factorization with automatic model selection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1284–1292. IEEE, 2015.

[Croux and Filzmoser, 1998] Christophe Croux and Peter Filzmoser. Robust factorization of a data matrix. In *COMPSTAT*, pages 245–250. Springer, 1998.

[Ding *et al.*, 2011] Xinghao Ding, Lihan He, and Lawrence Carin. Bayesian robust principal component analysis. *IEEE Transactions on Image Processing*, 20(12):3419–3430, 2011.

[Ji *et al.*, 2010] Hui Ji, Chaoqiang Liu, Zuowei Shen, and Yuhong Xu. Robust video denoising using low rank matrix completion. In *2010 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1791–1798. IEEE, 2010.

[Ke and Kanade, 2005] Qifa Ke and Takeo Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *2005 IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 739–746. IEEE, 2005.

[Lakshminarayanan *et al.*, 2011] Balaji Lakshminarayanan, Guillaume Bouchard, and Cedric Archambeau. Robust bayesian matrix factorisation. In *AISTATS*, pages 425–433, 2011.

[Lin *et al.*, 2010] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[Meng and De La Torre, 2013] Deyu Meng and Fernando De La Torre. Robust matrix factorization with unknown noise. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1337–1344. IEEE, 2013.

[Meng *et al.*, 2013] Deyu Meng, Zongben Xu, Lei Zhang, and Ji Zhao. A cyclic weighted median method for l1 low-rank matrix factorization with missing entries. In *AAAI*, volume 4, page 6, 2013.

[Mitra *et al.*, 2010] Kaushik Mitra, Sameer Sheorey, and Rama Chellappa. Large-scale matrix factorization with missing data under additional constraints. In *Advances in Neural Information Processing Systems*, pages 1651–1659, 2010.

[Okatani *et al.*, 2011] Takayuki Okatani, Takahiro Yoshida, and Koichiro Deguchi. Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 842–849. IEEE, 2011.

[Qian *et al.*, 2016] Wei Qian, Bin Hong, Deng Cai, Xiaofei He, Xuelong Li, et al. Non-negative matrix factorization with sinkhorn distance. In *International Joint Conferences on Artificial Intelligence*, pages 1960–1966, 2016.

[Salakhutdinov and Mnih, 2007] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[Salakhutdinov and Mnih, 2008] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, pages 880–887. ACM, 2008.

[Shu *et al.*, 2014] Xianbiao Shu, Fatih Porikli, and Narendra Ahuja. Robust orthonormal subspace learning: Efficient recovery of corrupted low-rank matrices. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3874–3881, 2014.

[Smith *et al.*, 1980] J.A. Smith, Tzeu Lie Lin, and K.J. Ranson. The lambertian assumption and landsat data. *Photogrammetric Engineering and Remote Sensing*, 46(9):1183–1189, 1980.

[Wand *et al.*, 2011] Matthew P. Wand, John T. Ormerod, Simone A. Padoan, and Rudolf Fuhrwirth. Mean field variational bayes for elaborate distributions. *Bayesian Analysis*, 6(4):847–900, 2011.

[Wang and Yeung, 2013] Naiyan Wang and Dit-Yan Yeung. Bayesian robust matrix factorization for image and video processing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1785–1792. IEEE, 2013.

[Wang *et al.*, 2012] Naiyan Wang, Tiansheng Yao, Jingdong Wang, and Dit-Yan Yeung. A probabilistic approach to robust matrix factorization. In *European Conference on Computer Vision*, pages 126–139. Springer, 2012.

[Zhao *et al.*, 2014] Qian Zhao, Deyu Meng, Zongben Xu, Wangmeng Zuo, and Lei Zhang. Robust principal component analysis with complex noise. In *Proceedings of the 31st International Conference on Machine Learning*, pages 55–63, 2014.

[Zhou *et al.*, 2010] Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candes, and Yi Ma. Stable principal component pursuit. In *2010 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 1518–1522. IEEE, 2010.