Bayesian Robust Matrix Factorization for Image and Video Processing

Naiyan Wang **Dit-Yan Yeung**
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
winsty@gmail.com dyyeung@cse.ust.hk

Abstract

Matrix factorization is a fundamental problem that is often encountered in many computer vision and machine learning tasks. In recent years, enhancing the robustness of matrix factorization methods has attracted much attention in the research community. To benefit from the strengths of full Bayesian treatment over point estimation, we propose here a full Bayesian approach to robust matrix factorization. For the generative process, the model parameters have conjugate priors and the likelihood (or noise model) takes the form of a Laplace mixture. For Bayesian inference, we devise an efficient sampling algorithm by exploiting a hierarchical view of the Laplace distribution. Besides the basic model, we also propose an extension which assumes that the outliers exhibit spatial or temporal proximity as encountered in many computer vision applications. The proposed methods give competitive experimental results when compared with several state-of-the-art methods on some benchmark image and video processing tasks.

1. Introduction

Finding a low-rank approximation to some given data matrix is a fundamental problem in many computer vision and machine learning applications, e.g., *structure from motion* (SfM) and collaborative filtering. Its objective is to approximate the data matrix by a low-rank representation according to some chosen criteria. Depending on the application, additional constraints may also be incorporated, e.g., basis orthogonality constraints for SfM and non-negativity constraints for image analysis.

The conventional approach to matrix factorization is based on *singular value decomposition* (SVD). Optimality of the solution is guaranteed when the loss function is defined in terms of the l_2 norm. However, conventional SVD can only deal with complete data in the data matrix. To address this problem, some methods [3] have been proposed to relax this restriction. As another issue, when the data matrix is contaminated with outliers, the estimated solution may deviate significantly from the ground truth due to sensitivity of the l_2 norm to outliers.

To overcome this shortcoming due to the lack of robustness, one natural way is to use the non-zero counting function (a.k.a. l_0 "pseudo-norm", because strictly speaking it is not a norm) in place of the l_2 norm. It totally ignores the scale of the outliers and hence reduces its sensitivity to them. However, the corresponding optimization problem is intractable due to the discontinuous property of the l_0 "pseudo-norm". A common solution is to resort to convex relaxation by using the l_1 norm instead. Based on this approach, many interesting applications have emerged in recent years, e.g., [19, 14].

In this paper, we propose a full Bayesian formulation for robust matrix factorization which turns out to be related to a recent work called *probabilistic robust matrix factorization* (PRMF) [18]. We note that the drawback of PRMF lies in three aspects which we try to address here:

1. PRMF assumes that the basis and coefficient matrices are generated i.i.d. from zero-mean fixed-variance Gaussian distributions. This simplistic modeling assumption may be too restrictive, limiting the model flexibility needed for many real-world applications. Besides, PRMF adopts a point estimation approach. Full Bayesian treatment can take advantage of full posterior density estimation instead of only using the mode of it, and thus improve the predictive performance. An example is [15]. We present our full Bayesian model and its inference algorithm in sections 5.1 and 5.2, respectively.

2. For many computer vision applications in which PRMF can be applied, the outliers which correspond to moving objects in the foreground usually form groups with high within-group spatial or temporal proximity. However, PRMF treats each pixel independently with no clustering effect. Recent work [22] shows great improvement when such clustering result is introduced based on an optimization method. We address this issue in section 5.3.

3. When the loss function is defined based on the l_1 norm, the resulting method may not be robust enough when

This research has been supported by General Research Fund 621310 from the Research Grants Council of Hong Kong.

the number of outliers is large. Our approach goes beyond simply using the l_1 norm based on point estimation. On one hand, it involves a non-convex loss with more expressive modeling power. On the other hand, as a full Bayesian method, it greatly alleviates the instability problem suffered by other methods. We elaborate this aspect and relate our approach to some existing methods in section 6.

2. Related Work

Enhancing model robustness in matrix factorization is by no means a new topic in the computer vision and machine learning communities. An early work [6] proposed solving the problem using an alternating weighted robust regression method. Along the same line, Ke and Kanade [11] developed a method based on linear programming. More recently, Eriksson and Hengel [7] extended the traditional Wiberg algorithm with an l_1 loss. However, all these methods are computationally demanding making them unappealing for large-scale applications. Moreover, since they are not based on the regularization or Bayesian framework, overfitting is a potential risk that has to be dealt with.

A recent breakthrough in this research topic can be attributed to *principal component pursuit* (PCP) [4] and *stable principal component pursuit* (SPCP) [23]. Besides using the l_1 loss, PCP and SPCP utilize the nuclear norm for normalization. Convexity of the l_1 and nuclear norms enables the application of efficient convex program solvers [13]. By using the nuclear norm, many extensions have been proposed subsequently, e.g., l_1-ALP [21] enforces the basis to be orthogonal while DECOLOR [22] makes the outliers contiguous using a graph cut algorithm.

As for probabilistic methods, PMF [16] and BPMF [15] are two representative models for (non-robust) matrix factorization. Lakshminarayanan *et al.* [12] proposed a robust extension of BPMF for collaborative filtering based on Student's t-distribution. Another recent attempt is PRMF [18] which uses the *expectation-maximization* (EM) algorithm. The highlights of PRMF are its high efficiency and online extension. Other methods include *Bayesian robust PCA* (BRPCA) [5] and *variational Bayesian low-rank factorization* (VBLR) [1]. We review and compare with these two methods in detail in section 6.2.

There are also other related methods. The first one is [9]. The authors utilized stochastic gradient descent on the Grassmannian manifold to develop an online robust low-rank approximation method. The second one is to improve PCP for video processing [8]. The authors proposed a framework which uses a two-pass PCP refined by optical flow to improve the results. We note that the method proposed in our paper may also be applied under this framework to achieve further performance boost.

3. Notations

In this paper, $\mathbf{I}$ denotes an identity matrix of the proper size and $\mathbf{A}^T$ denotes the transpose of matrix $\mathbf{A}$. For matrix norm, the general l_p norm is defined as $\|\mathbf{A}\|_p = (\sum_{ij} |a_{ij}|^p)^{1/p}$. Some special cases include the ℓ_1 norm ($p = 1$) defined as $\|\mathbf{A}\|_1 = \sum_{ij} |a_{ij}|$ and the Frobenius norm ($p = 2$) defined as $\|\mathbf{A}\|_F = (\sum_{ij} a_{ij}^2)^{1/2}$. Another useful norm is the nuclear norm (or called trace norm) $\|\mathbf{A}\|_*$, which is defined as the sum of the singular values of $\mathbf{A}$. As for vectors, we assume that all are in the form of column vectors. Several probability distributions appear in this paper: $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ (the same notation is also used for the univariate normal distribution with $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ replaced by the mean and variance, respectively); $\mathcal{L}(\mu, \alpha)$ denotes the Laplace distribution with mean μ and scale α; $\mathrm{Exp}(\alpha)$ denotes the exponential distribution with parameter α; and $\mathcal{W}(\mathbf{V}, p)$ denotes the Wishart distribution with scale matrix $\mathbf{V}$ and degrees of freedom p.

4. Brief Review of PRMF

In what follows, we assume $\mathbf{Y} = [y_{ij}] \in \mathbb{R}^{m \times n}$ denotes the data matrix with the exact data representation depending on the application. For example, the entire matrix $\mathbf{Y}$ may correspond to an $m \times n$ image which is assumed to be of low rank. In the case of video, each column of $\mathbf{Y}$ corresponds to the features from one frame stacked together to form a column vector and hence different columns are highly correlated, implying the low-rank property. The notation $\mathbf{U} \in \mathbb{R}^{m \times r}$ is used to denote the basis matrix, $\mathbf{V} \in \mathbb{R}^{n \times r}$ the coefficient matrix, and $\mathbf{u}_i$ and $\mathbf{v}_j$ the ith and jth rows of $\mathbf{U}$ and $\mathbf{V}$, respectively. Each row of $\mathbf{U}$ and $\mathbf{V}$ is assumed to be mutually independent, and $\mathbf{Y}$ is determined by the low-rank matrix $\mathbf{U}\mathbf{V}^T$ (with $r \ll m, n$) plus some additive noise.

Wang *et al.* [18] proposed the PRMF model for robust matrix factorization. Due to space limitations, we only provide a brief review of the key steps here. A longer review of PRMF can be found in the supplemental material. PRMF places a Laplace distribution on the residual error as likelihood and Gaussian distributions on the factors as priors. Its probabilistic model formulation can be related to optimization-based algorithms by adopting the *maximum a posteriori* (MAP) approach so that the loss function is expressed in terms of the l_1 norm. For computational efficiency, a hierarchical view of the Laplace distribution is exploited by expressing the Laplace distribution as an infinite Gaussian mixture with the exponential distribution as mixing distribution:

$$\mathcal{L}(z \mid u, \alpha) = \int_0^\infty \mathcal{N}(z \mid u, \tau)\, \mathrm{Exp}(\tau \mid \frac{\alpha}{2})\, d\tau. \quad (1)$$

This hierarchical view is crucial to the development of an

efficient and potentially online EM algorithm for model inference in PRMF. Figure 1(a) shows its graphical model.

5. Bayesian Robust Matrix Factorization

In this section, we first present the graphical model and the generative process of our basic *Bayesian robust matrix factorization* (BRMF) model. We then present details of the model inference. Finally, we introduce clustering effect into the basic BRMF model by presenting a Markov extension.

5.1. Full Bayesian Model

We extend PRMF by introducing several new characteristics, with the major ones summarized below to facilitate understanding the differences between PRMF and BRMF:

1. We assume that the mean vectors and precision matrices (a.k.a. inverse covariance matrices) of the rows of $\mathbf{U}$ and $\mathbf{V}$ have conjugate priors (multivariate normal distribution and Wishart distribution, respectively). Learning the mean vectors can offer further flexibility to the generation of $\mathbf{Y}$ and learning the precision matrices can capture the correlation between different features.

2. We use a Laplace mixture with the *generalized inverse Gaussian* (GIG) distribution[1] as the noise model to further enhance model robustness. It has been demonstrated in [20] that using a Laplace mixture with GIG outperforms the l_1 norm when it is used to define a regularizer for variable selection. We expect it to be equally superior when it plays the role of a loss function instead of a regularizer.

The overall generative process is summarized as follows:[2]

1. Draw precision matrix $\mathbf{\Lambda}_u$ for $\mathbf{u}_i$ from $\mathcal{W}(\mathbf{W}_0, \nu_0)$.
2. Draw mean $\boldsymbol{\mu}_u$ for $\mathbf{u}_i$ from $\mathcal{N}(\boldsymbol{\mu}_0, (\beta_0 \mathbf{\Lambda}_u)^{-1})$.
3. Draw each row of basis $\mathbf{u}_i$ from $\mathcal{N}(\boldsymbol{\mu}_u, \mathbf{\Lambda}_u^{-1})$.
4. Draw scale η_{ij} for each observation from $\text{GIG}(p, a, b)$.
5. Draw each observation y_{ij} from $\mathcal{L}(\mathbf{u}_i^T \mathbf{v}_j, \eta_{ij})$.

Here $\mathbf{W}_0, \boldsymbol{\mu}_0, \nu_0, \beta_0, p, a, b$ denote the hyperparameters. To facilitate efficient sampling, we further express the Laplace distribution as an infinite Gaussian mixture as in equation (1). As such, we split step 5 into two substeps:

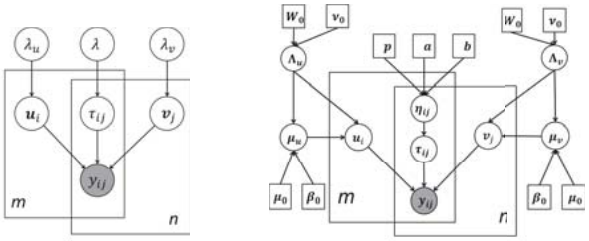
5.1 Draw $\tau_{ij} \sim \text{Exp}(\eta_{ij}/2)$.

5.2 Draw each observation y_{ij} from $\mathcal{N}(\mathbf{u}_i^T \mathbf{v}_j, \tau_{ij})$.

The graphical model for BRMF is shown in Figure 1(b).

[1] Readers are referred to the supplemental material for details of GIG.

[2] For brevity, we only show the process of generating $\mathbf{U}$. We can draw $\mathbf{\Lambda}_v, \boldsymbol{\mu}_v$ and $\mathbf{V}$ similarly by following steps 1 to 3.



(a) PRMF (b) BRMF

Figure 1. Plate notations for PRMF [18] and BRMF (proposed).

5.2. Model Inference

Since all the distributions belong to the (conjugate) exponential family, we can take advantage of the efficient Gibbs sampling to infer the posterior distributions. For more details, please refer to the supplemental material.

Sample $\boldsymbol{\mu}_u$ and $\mathbf{\Lambda}_u$: Based on the conjugate prior property, it is easy to show that the joint posterior distribution is a Gaussian-Wishart distribution and hence we can sample $\boldsymbol{\mu}_u, \mathbf{\Lambda}_u$ as follows:

$$\boldsymbol{\mu}_u, \mathbf{\Lambda}_u \mid \mathbf{U}, \mathbf{W}_0, \boldsymbol{\mu}_0, \nu_0, \beta_0 \sim \mathcal{N}(\boldsymbol{\mu}_0', (\beta_0' \mathbf{\Lambda}_u)^{-1}) \, \mathcal{W}(\mathbf{W}_0', \nu_0'), \tag{2}$$

where

$$\begin{aligned} \boldsymbol{\mu}_0' &= \frac{\beta_0 \boldsymbol{\mu}_0 + m\bar{\mathbf{u}}}{\beta_0 + m}, \qquad \beta_0' = \beta_0 + m, \qquad \nu_0' = \nu_0 + m \\ \mathbf{W}_0' &= \left(\mathbf{W}_0^{-1} + \bar{\mathbf{\Sigma}} + \frac{\beta_0 m}{\beta_0 + m} (\bar{\mathbf{u}} - \boldsymbol{\mu}_0)(\bar{\mathbf{u}} - \boldsymbol{\mu}_0)^T \right)^{-1} \\ \bar{\mathbf{u}} &= \frac{1}{m} \sum_{i=1}^{m} \mathbf{u}_i, \qquad \bar{\mathbf{\Sigma}} = \sum_{i=1}^{m} (\mathbf{u}_i - \bar{\mathbf{u}})(\mathbf{u}_i - \bar{\mathbf{u}})^T. \end{aligned} \tag{3}$$

The sampling scheme for $\boldsymbol{\mu}_v, \mathbf{\Lambda}_v$ is similar.

Sample $\mathbf{u}_i$: We extract all terms related to $\mathbf{u}_i$ and then apply Bayes' rule:

$$p(\mathbf{u}_i \mid \mathbf{Y}, \mathbf{V}, \boldsymbol{\mu}_u, \mathbf{\Lambda}_u, \mathbf{T}) \propto \prod_{j=1}^{n} \mathcal{N}(y_{ij} \mid \mathbf{u}_i^T \mathbf{v}_j, \tau_{ij}) \mathcal{N}(\mathbf{u}_i \mid \boldsymbol{\mu}_u, \mathbf{\Lambda}_u^{-1}). \tag{4}$$

Here $\mathbf{T} = [\tau_{ij}] \in \mathbb{R}^{m \times n}$. Then we can show that:

$$\mathbf{u}_i \mid \mathbf{Y}, \mathbf{V}, \boldsymbol{\mu}_u, \mathbf{\Lambda}_u, \mathbf{T} \sim \mathcal{N}(\mathbf{u}_i \mid \mathbf{u}_i', (\mathbf{\Lambda}_i')^{-1}), \tag{5}$$

where

$$\begin{aligned} \mathbf{\Lambda}_i' &= \mathbf{\Lambda}_u + \sum_{j=1}^{n} \frac{\mathbf{v}_j \mathbf{v}_j^T}{\tau_{ij}} \\ \mathbf{u}_i' &= (\mathbf{\Lambda}_i')^{-1} \left(\sum_{j=1}^{n} \frac{y_{ij} \mathbf{v}_j}{\tau_{ij}} + \mathbf{\Lambda}_u \boldsymbol{\mu}_u \right). \end{aligned} \tag{6}$$

Note that since the rows of $\mathbf{U}$ are independent, they may be sampled in parallel. We note that this potential for parallelization is important for large-scale applications. The sampling scheme for $\mathbf{v}_j$ is similar.

Sample τ_{ij}:

$$\frac{1}{\tau_{ij}} \mid y_{ij}, \mathbf{u}_i, \mathbf{v}_j, \eta_{ij} \sim \text{IG}\left(\frac{\sqrt{\eta_{ij}}}{|r_{ij}|}, \eta_{ij} \right), \tag{7}$$

where $r_{ij} = y_{ij} - \mathbf{u}_i^T \mathbf{v}_j$ and IG denotes the *inverse Gaussian* distribution.

Sample η_{ij}:

$$\eta_{ij} \mid \tau_{ij}, p, a, b \sim \text{GIG}(p+1, \tau_{ij}+a, b). \tag{8}$$

Note that sampling directly from GIG may be inefficient, so we convert the posterior distribution to some other form which allows more efficient sampling. To achieve this, we choose one of the special cases of GIG [10]. Specifically, we take $p = -1/2$ and thus the posterior distribution becomes $\frac{1}{\eta_{ij}} \sim \text{IG}(\sqrt{\frac{\tau_{ij}+a}{b}}, \tau_{ij}+a)$.

5.3. Markov Extension

In order to extend the basic model by assuming that the outliers form clusters (which correspond to moving objects in the foreground in the case of background modeling), we propose an extension via placing a first-order *Markov random field* (MRF) in the generation of $\mathbf{T}$. We refer to this extended form as *Markov BRMF* (MBRMF). Then for each pair of entries (i,j) and (p,q) which are neighbors, we define a potential function as follows:

$$\boldsymbol{\psi}(\tau_{ij}, \tau_{pq}) = \exp\{-\alpha \,|\log \tau_{ij} - \log \tau_{pq}|\}, \tag{9}$$

where α controls the strength of the prior. Consequently, step 5.1 of the generative process is changed to

$$\tau_{ij} \sim \frac{1}{Z} \text{Exp}(\frac{\eta_{ij}}{2}) \prod_{(p,q)\in N(i,j)} \boldsymbol{\psi}(\tau_{ij}, \tau_{pq}), \tag{10}$$

where Z is a normalization constant and $N(i,j)$ is the set of all the neighbors of entry (i,j). The sampling for τ_{ij} (Equation (7)) should also be changed accordingly:

$$p(\frac{1}{\tau_{ij}} \mid \eta_{ij}, r_{ij}, \mathbf{T}_{-ij}) \propto \text{IG}(\frac{\sqrt{\eta_{ij}}}{|r_{ij}|}, \eta_{ij}) \prod_{(p,q)\in N(i,j)} \boldsymbol{\psi}(\tau_{ij}, \tau_{pq}), \tag{11}$$

where $\mathbf{T}_{-ij}$ denotes $\mathbf{T}$ except τ_{ij}. The definition of neighborhood depends on the target application. For example, if each entry in the data matrix represents a raw pixel in an image, it can simply be defined as its 4-connected or 8-connected neighbors. For video processing, we may define the neighborhood based on both the inter-frame and intra-frame relationships: if $F_t(x,y)$ denotes the pixel (x,y) in frame t, then its neighbors may include $F_{t-1}(x,y)$ and $F_{t+1}(x,y)$ in addition to its 4-connected or 8-connected neighbors in frame t.

This Markov extension results in a more flexible model but also brings about some technical difficulties in inference. Specifically, we can no longer rely entirely on Gibbs sampling but have to resort to the Metropolis-Hastings algorithm which typically is less efficient than Gibbs sampling. For the proposal distribution, we choose

$$q(\tau_{ij} \mid \eta_{ij}, r_{ij}) = \text{IG}\Big(\frac{\sqrt{\eta_{ij}}}{|r_{ij}|}, \eta_{ij}\Big). \tag{12}$$

The advantage of this choice is that we can simplify the calculation of the acceptance ratio and accept it with the following probability:

$$\begin{aligned} &\min\left(1, \frac{p(\tau_{ij} \mid \eta_{ij}, r_{ij}, \mathbf{T}_{-ij}) q(\hat{\tau}_{ij} \mid \eta_{ij}, r_{ij})}{p(\hat{\tau}_{ij} \mid \eta_{ij}, r_{ij}, \mathbf{T}_{-ij}) q(\tau_{ij} \mid \eta_{ij}, r_{ij})}\right) \\ =&\min\left(1, \frac{\prod_{(p,q)\in N(i,j)} \boldsymbol{\psi}(\tau_{ij}, \tau_{pq})}{\prod_{(p,q)\in N(i,j)} \boldsymbol{\psi}(\hat{\tau}_{ij}, \tau_{pq})}\right). \end{aligned} \tag{13}$$

Here $\hat{\tau}_{ij}$ denotes the newly sampled τ_{ij}. We find that this scheme works well in practice.

6. Relationship with Existing Methods

To understand better the anticipated superiority of BRMF, we compare and relate BRMF with several popular robust matrix recovery algorithms in this section.

6.1. Optimization-Based Methods

In this subsection, we compare BRMF with several popular optimization-based methods, including PCP [4], SPCP [23], and DECOLOR [22]. In particular, we focus on how the error residue contributes to the objective function near the optimal solution because it affects the robustness of an algorithm most directly.

Because all the methods considered regard the signal as consisting of the low-rank ($\mathbf{B}$) and sparse ($\mathbf{E}$) components, we denote the optimal solution by $\mathbf{B}^*$ and $\mathbf{E}^*$ which correspond to the two components, respectively. Let $\mathbf{R} = \mathbf{Y} - \mathbf{B}^*$ denote the residue. By the optimality condition for each method, we can eliminate $\mathbf{E}^*$ and express the loss function in terms of $\mathbf{R}$ only. The loss functions are shown in Table 1 together with the corresponding objective functions. We drop the Markov dependency term of DECOLOR for brevity since it will not affect its robustness.

For BRMF, we need to consider a simplified MAP version since all the other methods are MAP based. Our full Bayesian treatment is actually more flexible and powerful because the MAP version only uses the posterior mode. We first compute the *probability density function* (pdf) of the residue as

$$p(r_{ij} \,|\, p, a, b) = \iint p(r_{ij} \,|\, \tau_{ij})\, p(\tau_{ij} \,|\, \eta_{ij}) p(\eta_{ij} \,|\, p, a, b)\, d\tau_{ij}\, d\eta_{ij}. \tag{14}$$

From [20], the integration can be computed as

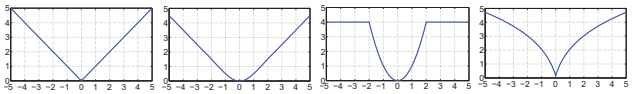
$$\begin{aligned} p(r_{ij} \mid p, a, b) = &\frac{b^{1/2} \exp(\sqrt{ab})}{4(b+|r_{ij}|)^{3/2}} \left(1 + \sqrt{a(b+|r_{ij}|)}\right) \cdot \\ &\exp\Big(-\sqrt{a(b+|r_{ij}|)}\Big), \end{aligned} \tag{15}$$

when $p = -1/2$. The negative logarithm of equation (15) can be treated as the loss function of the MAP version of BRMF. We compare these four loss functions in Figure 2. As we can see, the loss functions of DECOLOR and BRMF are non-convex while those of PCP and SPCP are convex.

	Original objective function	Contribution of residue to objective	
PCP	$\min_{\mathbf{B}} \|\mathbf{Y}-\mathbf{B}\|_1+\alpha\|\mathbf{B}\|_*$	$\|r_{ij}\|$	
SPCP	$\min_{\mathbf{B},\mathbf{E}} \frac{1}{2}\|\mathbf{Y}-\mathbf{B}-\mathbf{E}\|_F^2+\alpha\|\mathbf{B}\|_*+\beta\|\mathbf{E}\|_1$	$r_{ij}^2/2$	if $\|r_{ij}\|<\beta$
		$\beta\|r_{ij}\|-\beta^2/2$	if $\|r_{ij}\|\geq\beta$
DECOLOR	$\min_{\mathbf{B},\mathbf{E}} \frac{1}{2}\|\mathbf{Y}-\mathbf{B}-\mathbf{E}\|_F^2+\alpha\|\mathbf{B}\|_*+\beta\|\mathbf{E}\|_0$	$r_{ij}^2/2$	if $\|r_{ij}\|<\sqrt{2\beta}$
		β	if $\|r_{ij}\|\geq\sqrt{2\beta}$

Table 1. Comparison of several popular robust matrix recovery methods.

Non-convex penalties tend to have better performance when the irrepresentable condition is not met [17]. From the practical aspect, they are better when the deviation and number of outliers are high, which are typical conditions encountered in our applications. One problem with using non-convex loss functions in optimization-based methods is that it may lead to unstable results especially when the data contain noise. This is validated in our experiments later for DECOLOR, which uses a non-convex loss function. In addition, the non-convex loss function of DECOLOR is not smooth at two non-zero points. However, these two issues do not arise in BRMF model: 1) BRMF is only non-smooth at zero, which means BRMF does not explicitly distinguish outliers from noise. 2) The result of BRMF is based on the expectation of samples. These two properties allow BRMF to greatly alleviate the instability problem and tend to produce more stable results even when the noise level is high.



(a) PCP (b) SPCP (c) DECOLOR (d) BRMF

Figure 2. Comparison of loss functions of four different methods.

6.2. Bayesian Methods

In this subsection, we compare our method with two full Bayesian methods, namely, VBLR [1] and BRPCA [5].

VBLR differs from BRMF in several aspects. First, it uses the Jeffreys prior to model both noise and outliers. Second, it assumes that each basis is independent. Also, it adopts variational approximation for model inference. Although the Jeffreys prior has shown successes in variable selection, in outlier modeling, it may lead to unstable results with non-convergent properties when the number of outliers is large and there exists noise in data matrix, which is verified in our synthetic experiment in the supplemental material. Besides, it is difficult to incorporate a Markov extension when the variational approximation approach is used.

As for BRPCA, it adopts quite a different approach. This method is originated from factor analysis. It utilizes the beta-Bernoulli conjugate distribution, which is sparse (binary) in nature, to model the outliers and the low-rank component. It is regarded as a "Bayesian l_0" model because of this property. The incorporation of noise, outliers and the low-rank component makes the model quite complicated. Among other things, there is strong coupling between the variables. For example, the low-rank component involves the multiplication of three variables with one of them being binary. As a consequence, it generally needs more steps to converge to a stationary distribution, if at all, or may even get stuck at local optima.

7. Experiments

In this section, we empirically compare BRMF and MBRMF with some state-of-the-art algorithms on several tasks. The algorithms compared include PCP [4], BRPCA [5], DECOLOR [22], PRMF [18], and VBLR [1]. The codes and other supplemental materials can be found in http://winsty.net/brmf.html.

7.1. Parameter Setting

In our experiments to be presented below, we fix most of the hyperparameters: $\mathbf{W}_0=\mathbf{I}$, $\boldsymbol{\mu}_0=\mathbf{0}$, $\nu_0=r$, $\beta_0=\alpha=2$, $p=-1/2$, $a=10^{-4}$. The only hyperparameter that needs to be "tuned" is b, which is set in two different ways for the noisy case ($b=10$) and noiseless case ($b=1$). Although our Bayesian methods are slower than optimization-based methods, the fact that very minimal effort is needed for tuning their hyperparameters makes them superior to optimization-based methods. For the *Markov chain Monte Carlo* (MCMC) steps, we empirically find that the proposed algorithm converges very fast. As such, we only need 50 steps for burn-in. Afterwards, we collect 50 samples for approximating the posterior distribution by sampling once every two steps.[3] This scheme works well in all our experiments.

7.2. Text Removal

We first conduct a proof-of-concept experiment by considering a simulated image processing task based on artificially generated data. The goal of this task is to remove some generated text from an image with somewhat regular pattern. This experiment has been inspired by a previous work [19] which studied a more advanced version of this task. The clean, ground-truth image is of size 256×256 with rank equal to 10 for the data matrix. First, we add ran-

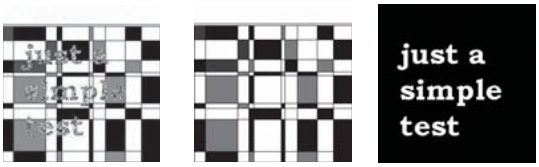
[3] Independence between samples can be further enhanced by sampling less frequently but it will need to run for more steps.

dom noise to the image to make the *peak signal-to-noise ratio* (PSNR) of the resulting image equal to 100. We then add to the image a short phrase in text form which plays the role of outliers. Figure 3 shows the image together with the clean image and outlier mask. For fairness, we set the maximum rank of all the algorithms to 20, which is two times the true rank of the underlying matrix.

The results obtained by different methods are visually shown in Figure 4. As far as outlier detection is concerned, DECOLOR and MBRMF obtain the best masks, not only visually but also quantitatively, and all other algorithms except BRPCA give reasonable results. BRPCA totally fails in this task. On the aspect of low-rank matrix recovery, it can be seen that MBRMF gives the best result, which is followed by DECOLOR and BRMF. Although PCP and PRMF do a fair job in detecting the outliers, they fail to recover the low-rank matrix well. On the other hand, although BRPCA performs poorly in detecting outliers, it outperforms PCP and PRMF slightly in recovering the low-rank matrix. Quantitative results are also shown in Table 2. We can see that MBRMF has a clear advantage over the other algorithms for both outlier detection and low-rank matrix recovery. Specifically, it exceeds the PSNR of the nearest competitor DECOLOR by 3.63dB, which can be considered a significant improvement.

	PCP	BRPCA	DECOLOR	PRMF	BRMF	VBLR	MBRMF
AUC	0.972	0.946	0.998	0.975	0.992	0.978	**0.999**
PSNR	25.66	26.40	29.38	24.05	28.66	23.93	**33.01**

Table 2. Comparison of different methods for text removal. The first row shows the outlier detection results based on the area under curve (AUC) measure and the second row shows the image recovery results based on PSNR.



(a) Input image (b) Clean image (c) Outlier mask

Figure 3. Image used in text removal experiment.

7.3. Background Modeling

In this subsection, we compare different methods for background modeling which is a real-world application. Specifically, given a video sequence, we want to separate the foreground objects from the background. We consider the scenario with a stationary camera. Consequently, the background components in different frames are highly correlated (and hence the data matrix is of low rank) and the moving objects in the foreground can be regarded as outliers. The challenge lies in selecting the right model to use. For example, while an underfitting model incorrectly recognizes the dynamic background such as some waving trees as the foreground, an overfitting one incorrectly ignores the foreground we are interested in. To provide more details about the results, entire video sequences are available in the supplemental material. Because we use the results of BRMF to initialize MBRMF, and MBRMF has already shown better results in the previous experiment, BRMF alone is not reported in the results. We set the maximum rank of all the methods to $\min(20, \sqrt{n})$, where n is the number of frames in a video sequence. Based on our (unoptimized) MATLAB implementation, each MCMC step takes around 1 to 10 seconds depending on the video resolution and the number of frames.

7.3.1 SABS Dataset

SABS [2] is a synthetic dataset suitable for background modeling research. An appealing advantage of this dataset is that ground-truth information is available for every frame making controlled experiments feasible. More details about the dataset can be found in [2]. Figure 5 shows one sample frame and the detection result using MBRMF.

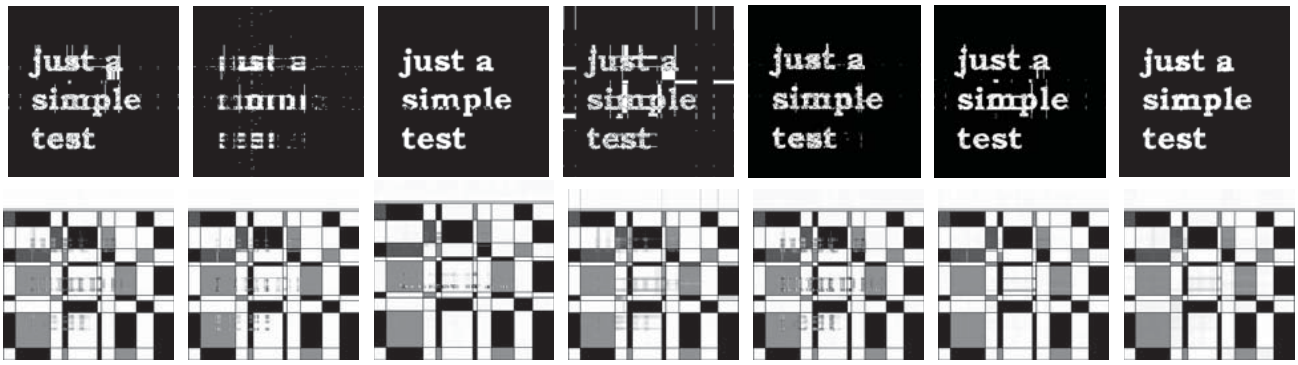
(a) Frame 200 of *Basic* (b) Foreground

Figure 5. Illustration of SABS [2] dataset.

Since all the methods included in this study are computationally demanding if applied to the original resolution, we first downsample the video frames to 150×200 in resolution and then convert them from color to grayscale. We note that downsampling will reduce the noise level in the *NoisyNight* case, so we add back the noise to make the noise level comparable to the original video data. For the evaluation criterion, we use the F-measure which is the harmonic mean of precision and recall. It is a preferred choice in the case that the class distribution is skewed. We exclude the *bootstrap* case because all methods compared do not need training and thus every case is already a bootstrap. The length of these 8 evaluations varies from 600 to 1400 frames. We do not compare with other methods such as mixture of Gaussians because it has been shown in [8] that robust matrix factorization methods significantly outperform the other methods. Quantitative results are summarized in Table 3. We note that all methods except PCP give reasonable results. MBRMF gets the best results in 7 out of 8 evaluations, which agree with findings from the previous experiments.

7.3.2 Real Dataset

We now conduct experiments on some real-world video sequences commonly used in background modeling research. More details about these video sequences are summarized



(a) PCP (b) BRPCA (c) DECOLOR (d) VBLR (e) PRMF (f) BRMF (g) MBRMF

Figure 4. Text removal results. The first row shows the foreground masks and the second row shows the restored background images.

	Basic	Dynamic	Darkening	LightSwitch	NoisyNight	Camouflage	NoCamouflage	MPEG-40Kbps	Average
PCP	0.698	0.619	0.639	0.620	0.672	0.701	0.706	0.693	0.669
BRPCA	0.739	0.662	0.769	0.731	0.719	0.809	0.821	0.750	0.750
DECOLOR	0.768	0.721	0.772	0.749	0.750	0.793	0.793	0.765	0.764
PRMF	0.775	0.657	0.757	0.740	0.678	0.821	0.823	0.768	0.752
VBLR	0.764	0.688	**0.776**	**0.766**	0.795	0.809	0.816	0.763	0.772
MBRMF	**0.824**	**0.729**	**0.776**	0.738	**0.800**	**0.830**	**0.838**	**0.820**	**0.794**

Table 3. Comparison of different methods on SABS dataset based on F-measure.

in the supplemental material. Due to the lack of full annotation, quantitative comparison for foreground detection cannot be made. Instead, we only visually examine the quality of the reconstructed background.

As shown in Figure 6, PCP is clearly inferior to other methods compared. Obvious ghosting effect can be observed in all the video sequences except *wavingTrees*. Although it is also observed in the results obtained by PRMF and VBLR, the effect is significantly less obvious. As for BRPCA, it only fails for *hall* which is extremely short. DECOLOR and MBRMF are successful in almost all sequences. There is another interesting observation in the *wavingTrees* sequence: reconstruction of the trees in the background by MBRMF is clearer than those by other methods. This is in line with the synthetic experiment above in showing that MBRMF is more accurate in recovering the low-rank matrix.

7.3.3 Robustness to Noise

We further conduct some experiments to study the robustness of different methods to various types of noise (Gaussian, Salt, Poisson, Speckle). Due to space limitations, we only show the results of *fountain* with additive Gaussian noise in Figure 7 and leave the rest to the supplemental material. Even though all methods except PCP perform well under the noiseless setting, only MBRMF and VBLR survive when noise is added. We observe serious ghosting effect in the results of PCP, BRPCA and DECOLOR. The effect for PRMF is also visible though less severe. On the other hand, MBRMF is still very robust due mainly to its generic noise model which does not distinguish outliers from noise, as explained earlier in section 6.

8. Conclusion and Future Work

In this paper, we have proposed a novel full Bayesian model for robust matrix factorization together with a Markov extension which incorporates spatial or temporal proximity. Due to the use of conjugate priors for the model parameters, an efficient sampling algorithm can be devised for Bayesian inference. Using both synthetic and real datasets, our experiments show that the proposed methods, particularly MBRMF, outperform other state-of-the-art robust matrix factorization methods. Moreover, MBRMF remains robust even under high noise level.

Currently the inference algorithms for our proposed models are batch algorithms. In applications where data arrive sequentially, online algorithms will be more useful. Some challenges have to be overcome when extending the current algorithms to online algorithms. Moreover, in order to apply the proposed algorithms to high-resolution video and achieve real-time performance, we will also investigate random sampling techniques and GPU implementations of the algorithms in our future work.

References

[1] S. Babacan, M. Luessi, R. Molina, and A. Katsaggelos. Sparse Bayesian methods for low-rank matrix estimation. *IEEE Transactions on Signal Processing*, 60(8):3964–3977, 2012. 2, 5

[2] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *CVPR*, pages 1937–1944, 2011. 6

(a) Original (b) PCP (c) BRPCA (d) DECOLOR (e) PRMF (f) VBLR (g) MBRMF

Figure 6. Results of real background modeling.

(a) Noisy (b) PCP (c) BRPCA (d) DECOLOR (e) PRMF (f) VBLR (g) MBRMF

Figure 7. Robustness of different methods to additive Gaussian noise.

[3] A. M. Buchanan and A. W. Fitzgibbon. Damped Newton algorithms for matrix factorization with missing data. In *CVPR*, pages 316–322, 2005. 1

[4] E. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the Association for Computing Machinery*, 58(3), 2011. 2, 4, 5

[5] L. Carin, X. Ding, and L. He. Bayesian robust principal component analysis. *IEEE Transactions on Image Processing*, 20(12):3419–3430, 2011. 2, 5

[6] C. Croux and P. Filzmoser. Robust factorization of a data matrix. In *COMPSTAT*, pages 245–249, 1998. 2

[7] A. Eriksson and A. Van Den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l_1 norm. In *CVPR*, pages 771–778, 2010. 2

[8] Z. Gao, L.-F. Cheong, and M. Shan. Block-sparse RPCA for consistent foreground detection. In *ECCV*, pages 690–703, 2012. 2, 6

[9] J. He, L. Balzano, and A. Szlam. Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video. In *CVPR*, pages 1568–1575, 2012. 2

[10] B. Jørgensen. *Statistical Properties of the Generalized Inverse Gaussian Distribution*, volume 21. Springer, New York, 1982. 4

[11] Q. Ke and T. Kanade. Robust l_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *CVPR*, pages 739–746, 2005. 2

[12] B. Lakshminarayanan, G. Bouchard, and C. Archambeau. Robust Bayesian matrix factorisation. In *AISTATS*, 2011. 2

[13] Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Arxiv preprint arXiv:1009.5055*, 2010. 2

[14] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *CVPR*, pages 763–770, 2010. 1

[15] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *ICML*, pages 880–887, 2008. 1, 2

[16] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. *NIPS*, 20:1257–1264, 2008. 2

[17] Y. She and A. Owen. Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*, 106(494):626–639, 2011. 5

[18] N. Wang, T. Yao, J. Wang, and D.-Y. Yeung. A probabilistic approach to robust matrix factorization. In *ECCV*, pages 126–139, 2012. 1, 2, 3, 5

[19] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma. TILT: Transform invariant low-rank textures. *International Journal of Computer Vision*, 99(1):1–24, 2012. 1, 5

[20] Z. Zhang, S. Wang, D. Liu, and M. Jordan. EP-GIG priors and applications in Bayesian sparse learning. *Journal of Machine Learning Research*, 13:2031–2061, 2012. 3, 4

[21] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi. Practical low-rank matrix approximation under robust l_1 norm. In *CVPR*, pages 771–778, 2012. 2

[22] X. Zhou, C. Yang, and W. Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):597–610, 2013. 1, 2, 4, 5

[23] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma. Stable principal component pursuit. In *International Symposium on Information Theory*, pages 1518–1522, 2010. 2, 4