# Bayesian Adaptive Matrix Factorization with Automatic Model Selection

Peixian Chen pchenac@cse.ust.hk
Naiyan Wang winsty@gmail.com
Nevin L. Zhang lzhang@cse.ust.hk
Dit-Yan Yeung dyyeung@cse.ust.hk
The Hong Kong University of Science and Technology

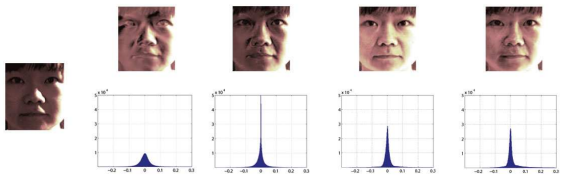
## Abstract

*Low-rank matrix factorization has long been recognized as a fundamental problem in many computer vision applications. Nevertheless, the reliability of existing matrix factorization methods is often hard to guarantee due to challenges brought by such model selection issues as selecting the noise model and determining the model capacity. We address these two issues simultaneously in this paper by proposing a robust non-parametric Bayesian adaptive matrix factorization (AMF) model. AMF proposes a new noise model built on the Dirichlet process Gaussian mixture model (DP-GMM) by taking advantage of its high flexibility on component number selection and capability of fitting a wide range of unknown noise. AMF also imposes an automatic relevance determination (ARD) prior on the low-rank factor matrices so that the rank can be determined automatically without the need for enforcing any hard constraint. An efficient variational method is then devised for model inference. We compare AMF with state-of-the-art matrix factorization methods based on data sets ranging from synthetic data to real-world application data. From the results, AMF consistently achieves better or comparable performance.*

## 1. Introduction

Matrix factorization is a crucial component in many computer vision applications, such as face recognition [35], motion segmentation [7], and structure from motion (SfM) [27]. Briefly speaking, it approximates a given data matrix by the product of a basis matrix and a coefficient matrix under some criteria. If the underlying rank of the two factor matrices is lower than that of the original data matrix, matrix factorization is an effective way to reveal the low-dimensional structure of the data.

The success of matrix factorization depends very much on proper model selection. Two model selection problems are involved. The first one is selection of the noise model which affects how well each entry of the matrix can be represented by the model, and the second one is the selection of the capacity of the two factor matrices related to the expressive power of the model. In Fig. 1, we empirically show the effect of these two problems in a face shadow removal application. As we can see, the noise model indeed affects the performance greatly while the rank controls the degree of abstraction.



(a) Original (b) Rank 8, 1 Gaussian (c) Rank 8, 1 Laplace (d) Rank 4, 32 Gaussian(Ours) (e) Rank 8, 32 Gaussian(Ours)

Figure 1. A face shadow removal example demonstrating the effect of some model selection issues. The original image is shown in (a) and the recovery results of different models are shown in (b)-(e). In the results, the first row shows the recovered images and the second row shows the corresponding noise distributions of the matrices.

For the first model selection problem, various attempts have been made to find a better description for the noise underlying the data, ranging from the traditional $\ell_2$ norm, the robust $\ell_1$ norm [5] to non-convex norms [30]. These methods often make overly strong assumptions about the noise distribution which unfortunately do not hold in many real applications. Recently, Meng and De la Torre proposed to use a *Gaussian mixture model* (GMM) to fit the noise [18], and further extended it into a full Bayesian model [32]. Although it is more flexible than the methods above, the number of Gaussian components in the GMM still has to be specified in advance. This is a real limitation since the number of Gaussians weighs heavily in the generalizability of a GMM noise model. In this paper we propose to use a *Dirichlet process Gaussian mixture model* (DP-GMM) [22] as the noise model. On one hand, we can take advantage of the fact that GMM is a universal approximator for any continuous distribution [17] and thus able to fit various types of noise. On the other hand, we can infer the number of Gaussian components needed from data, instead of doing heuristic pruning or trying ungrounded guesses.

For the second problem, one method is to limit the rank

of the factor matrices directly while another is to impose regularization terms on them. Except for some applications such as SfM in which strong knowledge of the true rank is available, it is generally difficult to estimate the underlying rank accurately. Some heuristic methods [26, 19] have been developed for rank selection but their performance is not stable across different applications. For the regularization approach, some regularizers have been proposed to fulfill the requirements of specific applications, such as the basis orthogonality constraints in SfM and non-negativity constraints in image analysis. Some others have been chosen simply to avoid overfitting, e.g., imposing $\ell_2$ regularization [24, 28], which is a generalization of the nuclear norm and is effective at reducing the rank of the resulting matrix. Here we choose an *automatic relevance determination* (ARD) [15, 20] prior for the factor matrices. ARD has long been recognized as an effective technique for detecting the relevant components of the input, so that we can automatically infer the optimal rank by pruning other irrelevant ones.

In this paper, we propose our novel non-parametric full Bayesian model for *adaptive matrix factorization* (AMF). AMF for the first time makes full use of the flexibility and adaptiveness of DP-GMM as noise model, and is completed by ARD for automatic rank selection. It is also designed to be capable of handling input with missing data. For model inference, we devise an efficient variational method based on the stick-breaking representation of DP. For experimental validation, we use the text removal and face shadow removal tasks to demonstrate the effectiveness of the automatic model selection capabilities of AMF.

## 2. Related Work

Matrix factorization has its root in numerical analysis. One of the most commonly used methods is *singular value decomposition* (SVD), which can give the optimal solution if the true rank is known and additive Gaussian noise is assumed. A drawback of SVD is that it cannot cope with missing data. To remedy it, Buchanan and Fitzgibbon proposed a damped Newton's method in [4]. To enhance the robustness of SVD, a method based on iteratively reweighted least squares estimation was first proposed in [7]. Following it, some other early methods cast the matrix factorization problem as several small linear programming problems in each step, e.g., [13, 8]. However, all these methods have high computation cost and hence are not suitable for large-scale applications. A recent breakthrough in matrix factorization has been brought by *principal component pursuit* (PCP) [5]. PCP reformulates the problem as a convex optimization problem. It uses the nuclear norm to regularize the rank of the resulting matrix and the $\ell_1$ norm as the noise model to accommodate outliers in the matrix. To solve the optimization problem, it advocates using the *alternating direction method of multipliers* (ADMM) [3] which yields considerable speedup when compared to the methods above. Inspired by this pioneering work, [33, 34, 26] exploited different settings for matrix factorization and demonstrated great improvement over traditional methods in several applications. Another emerging trend is to apply gradient descent on the Grassmannian manifold, e.g.,[11, 31]. The advantage of these methods lies in their ability of learning the low-rank matrix factorization online. In [24], Salakhutdinov and Mnih first formulated the matrix factorization problem in a probabilistic framework. Then it was extended to a full Bayesian model [23] and a nonlinear model [14]. All the three models are well suited for collaborative filtering applications. Along this line, Wang *et al.* proposed two robust Bayesian matrix factorization methods from the point estimation perspective [28] and the full Bayesian perspective [29], yielding good results in some computer vision applications. Some other related methods include the Bayesian robust principal component analysis [6] which can be seen as a probabilistic version of PCP, and the variational Bayesian low-rank matrix estimation [1], which also relies on ARD and uses a fast variational inference algorithm for Bayesian matrix factorization. The previous works that are most closely related to ours are [18, 32]. They both use a GMM to model the possibly complex and unknown noise in the data. Though they add priors on the factor matrices in the latter work [32], the need for setting the number of Gaussian components beforehand to a great extent limits the generalizability of this model. And this new algorithm lacks the ability of tackling inputs with missing data. We will give empirical comparison with these related models and show AMF consistently achieves better or comparable results.

## 3. Notations

We introduce some notations to be used in the sequel. For a matrix $\mathbf{X}$, $\mathbf{X}^T$ denotes its transpose and $\mathrm{tr}(\mathbf{X})$ its trace. We also use $\mathbf{x}_{i\cdot}$ and $\mathbf{x}_{\cdot j}$ to denote the $i$th row and $j$th column, respectively, of $\mathbf{X}$. Let $\mathbf{I}$ denote the identity matrix with proper size. For probability distributions, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\mathcal{N}(\mu, \sigma)$ the univariate normal distribution with mean $\mu$ and variance $\sigma$, $\mathcal{B}(\alpha, \beta)$ the beta distribution with parameters $\alpha$ and $\beta$, $\mathbf{IG}(\alpha, \beta)$ the inverse-gamma distribution with shape parameter $\alpha$ and scale parameter $\beta$, and $\mathbf{Mult}(\pi)$ the multinomial distribution.

## 4. Background

In this section, we review some background knowledge to set the stage for presenting our model in the next section. We first introduce the general form of low-rank matrix factorization and then review the Dirichlet process.

### 4.1. Low-Rank Matrix Factorization

As with most common low-rank matrix factorization models, the input data matrix $\mathbf{Y}$ can be expressed as

$$\mathbf{Y} = \mathbf{U}\mathbf{V}^T + \mathbf{E}, \tag{1}$$

where the data matrix $\mathbf{Y} = [y_{mn}] \in \mathbb{R}^{M\times N}$ is assumed to be of low rank, $\mathbf{U} \in \mathbb{R}^{M\times R}$ and $\mathbf{V} \in \mathbb{R}^{N\times R}$ are the factor matrices, and $\mathbf{E} = [\epsilon_{mn}] \in \mathbb{R}^{M\times N}$ denotes the additive noise. The representation essentially decomposes $\mathbf{Y}$ into two low-rank factor matrices with rank $R \ll \min(M, N)$.

For example, if the data is just a single image which is assumed to be of low rank, then the matrix $\mathbf{Y}$ can simply be set as the image. If the data is a video or a collection of highly correlated images, then we need to first vectorize each frame of the video or each image of the collection to form one column of the matrix $\mathbf{Y}$.

### 4.2. Dirichlet Process and Stick-Breaking Construction

Since we will define our noise model based on the DP-GMM, we first review here some basic definitions and applications of the Dirichlet process.

#### 4.2.1 Dirichlet Process

The concept of *Dirichlet process* (DP) was proposed by Ferguson [9]. We may consider a DP as an extension of the ordinary Dirichlet distribution by taking the number of components $K$ to infinity. Let us take DP-GMM as an example for illustration. The DP is a distribution over an infinite number of clusters, each of which has a set of parameters $\chi_k$ for a Gaussian distribution, namely the mean and covariance. A draw from this DP will choose some cluster $k$ according to the Dirichlet distribution and return the parameters specific to the cluster. Consequently, we obtain a Gaussian distribution with the parameter set $\chi_k$. Hence a DP is a "distribution over distributions".

DP offers full flexibility in selecting the number of clusters. Any model built on a DP is thus able to accommodate multinomial distributions with arbitrarily many categories or possible outcomes. Within the context of mixture models, DP provides a non-parametric Bayesian choice that is capable of automatically determining the number of mixture components required to model the target distribution.

#### 4.2.2 Stick-Breaking Construction

Stick-breaking construction refers to a stochastic process for constructing a DP [25]. Consider a unit-length stick (0,1). We first draw a value $\beta_1$ from the beta distribution $\mathcal{B}(1, \alpha)$. Then we let $\theta_1 = \beta_1$, and pick the fraction $1-\beta_1$ as the remainder of the stick. Then we draw $\beta_2$ from $\mathcal{B}(1, \alpha)$, and make $\theta_2$ equal to $\beta_2(1 - \beta_1)$. Repeating this procedure, we have a sequence of sticks with lengths

$$\theta_k = \beta_k \prod_{l=1}^{k-1}(1 - \beta_l), \tag{2}$$

where $\beta_k$ are independent draws from the distribution $\mathcal{B}(1, \alpha)$. It is easy to show that $\sum_{k=1}^{\infty} \theta_k = 1$ with probability one. Following the construction above, the stick-breaking distribution over $\boldsymbol{\theta}$ is written as $\boldsymbol{\theta} \sim \mathbf{GEM}(\alpha)$ [21].

## 5. Our Model

In this section, we first present the key methods and motivation underlying our *adaptive matrix factorization* (AMF) model. This is then followed by the detailed generative process of AMF. After that, we discuss the relationship between AMF and existing methods based on the $\ell_2$ or $\ell_1$ loss.

### 5.1. Adaptive Matrix Factorization

The success of matrix factorization relies heavily on model selection, which, as discussed before, includes determining the capacity of the factor matrices and selecting the noise model.

In most real-world applications, the actual rank $R$ needed for modeling the data is initially unknown. For instance, in a background subtraction task with a static background, ideally it is adequate to set the rank to one. However, if the background is multimodal (e.g., due to periodically changing objects such as a traffic light), the rank needed is typically much higher. To determine the appropriate rank, a common approach is to try different values of $R$ by performing multiple runs and then choose the one that yields the best performance. To avoid inefficient and groundless attempts, we adopt the *automatic relevance determination* (ARD) method [16] by imposing a prior on each dimension (column) of $\mathbf{U}$ and $\mathbf{V}$ to curtail the irrelevant columns from impairing the performance. Specifically, we impose Gaussian priors with variance $\lambda_r$ on the $r$th columns of $\mathbf{U}$ and $\mathbf{V}$:

$$\begin{aligned} \mathbf{p}(\mathbf{U} \mid \boldsymbol{\lambda}) &= \textstyle\prod_{r=1}^{R} \mathcal{N}(\mathbf{u}_{\cdot r} \mid 0, \lambda_r \mathbf{I}_M), \\ \mathbf{p}(\mathbf{V} \mid \boldsymbol{\lambda}) &= \textstyle\prod_{r=1}^{R} \mathcal{N}(\mathbf{v}_{\cdot r} \mid 0, \lambda_r \mathbf{I}_N), \end{aligned} \tag{3}$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_R)^T$.

In terms of modeling the noise term $\mathbf{E}$, although it has been common to use a single Gaussian or Laplace distribution, this approach is often inadequate for many real-world applications in which the noise may be of different types or from different sources. Consequently, simply using an $\ell_2$ or $\ell_1$ loss cannot give satisfactory results. Considering

that GMM has proved to be a universal approximator for any continuous density function, it could be exploited to better characterize the noise from unknown sources using possibly multimodal probability distributions. For example, when modeling images, we may use a Gaussian distribution with large variance to handle large deviations due to shadow or occlusion while using another Gaussian distribution with smaller variance to fit the sensor noise. This provides greater flexibility in real applications.

Indeed, this idea has recently been pursued by [18, 32]. Nevertheless, the number of Gaussian components $K$ of the GMM used in [18, 32] has to be specified *a priori*. Its value often affects the performance of the model significantly. If $K$ is too small, the clusters may not be able to model well the complicated noise from wide-ranging sources. If $K$ is too large, however, it will be time-consuming to take every Gaussian component into consideration when most of them contribute little to modeling the noise. Moreover, without a proper prior on the mixing coefficients of the GMM, the model is unstable and easy to overfit.

To remedy these problems, we use a DP-GMM here for modeling the noise. On one hand, it retains the expressive power of GMM. On the other hand, it takes advantage of the non-parametric Bayesian approach by using a DP to determine the number of Gaussians from data automatically. An appealing advantage of the non-parametric Bayesian approach is that, instead of imposing assumptions that might be wrong, it "lets the data speak for itself".

Based on the stick-breaking construction, the GMM noise model extricates itself from the dilemma of selecting an appropriate number of components. After relaxing $K$ to infinity, the noise $\epsilon_{mn}$ can be expressed as:

$$\mathbf{p}(\epsilon_{mn}) = \sum_{k=1}^{\infty} \theta_k \mathcal{N}(\epsilon_{mn} \mid 0, \sigma_k), \tag{4}$$

where the mixing proportion of each Gaussian component is obtained from the stick-breaking process, i.e., $\boldsymbol{\theta} \sim \mathbf{GEM}(\alpha)$, with $\sum_{k=1}^{\infty} \theta_k = 1$. As a consequence, the noise entries will cluster themselves into $K$ groups without the need for a complicated model selection procedure.

### 5.2. Generative Process

We now combine the desirable features of ARD and DP-GMM to define the AMF model. We also place proper conjugate priors on the parameters if applicable. The graphical model of AMF is depicted in Fig. 2 and the generative process is given as follows:

1. Draw component mixing proportions $\boldsymbol{\theta} \sim \mathbf{GEM}(\alpha)$.
2. For each cluster $k$ of noise:
   - Draw variance $\sigma_k \sim \mathbf{IG}(a_0, b_0)$.

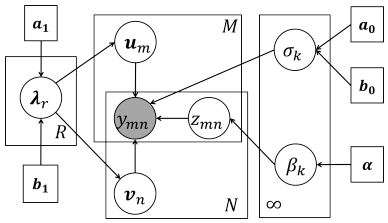


Figure 2. Graphical model of AMF

3. For each dimension $r$ of $\mathbf{U}$ and $\mathbf{V}$ (i.e., each column of $\mathbf{U}$ and $\mathbf{V}$):
   - Draw variance $\lambda_r \sim \mathbf{IG}(a_1, b_1)$.
4. For each element in $\mathbf{U}$ and $\mathbf{V}$:
   - Draw $u_{mr}, v_{nr} \sim \mathcal{N}(0, \lambda_r)$.
5. For each data element $y_{mn}$:
   - Draw noise cluster label $z_{mn} \sim \mathbf{Mult}(\boldsymbol{\theta})$;
   - Draw observation $y_{mn} \sim \mathcal{N}(\mathbf{u}_{m\cdot}\mathbf{v}_{n\cdot}^T, \sigma_{z_{mn}})$.

Here $\theta_k \triangleq \beta_k \prod_{l=1}^{k-1}(1-\beta_l)$ and $\beta_k$ is drawn independently from $\mathcal{B}(1, \alpha)$ according to the stick-breaking construction. In the model, $a_0, b_0, a_1, b_1, \alpha$ are hyperparameters. Based on the generative process, the joint distribution can be expressed as:

$$\begin{aligned} &\mathbf{p}(\mathbf{U}, \mathbf{V}, \mathbf{Y}, \mathbf{z}, \boldsymbol{\sigma}, \boldsymbol{\lambda}, \boldsymbol{\beta} \mid a_0, b_0, a_1, b_1, \alpha) \\ =&\mathbf{p}(\mathbf{Y} \mid \mathbf{U}, \mathbf{V}, \mathbf{z}, \boldsymbol{\sigma})\, \mathbf{p}(\mathbf{U} \mid \boldsymbol{\lambda})\, \mathbf{p}(\mathbf{V} \mid \boldsymbol{\lambda})\, \mathbf{p}(\boldsymbol{\lambda} \mid a_1, b_1) \\ &\mathbf{p}(\boldsymbol{\sigma} \mid a_0, b_0)\, \mathbf{p}(\mathbf{z} \mid \boldsymbol{\beta})\, \mathbf{p}(\boldsymbol{\beta} \mid \alpha). \end{aligned} \tag{5}$$

### 5.3. Relationship with Existing Methods

Let us try to investigate how the proposed method is related to existing methods based on the $\ell_2$ or $\ell_1$ loss. First, it is easy to see that using the $\ell_2$ loss is equivalent to using a single Gaussian to fit the noise, which is obviously less flexible. Second, as noted in [28], the $\ell_1$ norm or the corresponding Laplace distribution can be expressed as an integrated Gaussian mixture with mixing distribution equal to the exponential distribution. This means that for each entry in the matrix, it has its own GMM to represent the noise. This scheme offers too much flexibility for noise modeling since it is rarely the case for any noise to affect only a single data entry. Our approach may be seen as a tradeoff between the two approaches by allowing different entries to share the same Gaussian distribution.

## 6. Variational Inference

The key problem in the parameter estimation of Bayesian models is to compute the posterior distribution of the latent

variables given the observed data. Like in many Bayesian models, exact inference of the posterior distribution in our model is intractable and hence approximate inference is needed. Although sampling methods such as *Markov chain Monte Carlo* (MCMC) algorithms can provide very accurate asymptotic approximation to the posterior, they are often prohibitively slow for high-dimensional data. Moreover, convergence is not easy to detect. As a more efficient and deterministic alternative to MCMC, we adopt a mean-field variational method for AMF in this paper.

Variational methods approximate the posterior distribution of the latent variables by a factorized form consisting of new variational distributions $\mathbf{q}$ for the latent variables with free variational parameters. The approximation is made as close to the target posterior distribution $\mathbf{p}$ as possible by minimizing the *Kullback-Leibler* (KL) divergence of the two distributions.

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

$$\begin{aligned}\mathbf{q}(\mathbf{U},\mathbf{V},\mathbf{z},\boldsymbol{\sigma},\boldsymbol{\lambda},\boldsymbol{\beta}) = &\prod_{m=1}^{M}\mathbf{q}_{\mathbf{u}_{m\cdot}}(\mathbf{a}_{m\cdot},\boldsymbol{\Sigma}_m^u)\prod_{n=1}^{N}\mathbf{q}_{\mathbf{v}_{n\cdot}}(\mathbf{b}_{n\cdot},\boldsymbol{\Sigma}_n^v)\\ &\prod_{r=1}^{R}\mathbf{q}_{\lambda_r}(\eta_r)\prod_{k=1}^{K}\mathbf{q}_{\beta_k}(\gamma_k)\prod_{k=1}^{K}\mathbf{q}_{\sigma_k}(\tau_k)\prod_{m=1}^{M}\prod_{n=1}^{N}\mathbf{q}_{z_{mn}}(\phi_{mn}),\end{aligned} \tag{6}$$

where each row of $\mathbf{U}$ follows a Gaussian distribution with mean $\mathbf{a}_{m\cdot}$ and covariance $\boldsymbol{\Sigma}_m^u$ and it is similar for $\mathbf{V}$, $\lambda_r$ and $\sigma_k$ follow inverse-gamma distributions parametrized by $\eta_{r,1}, \eta_{r,2}$, and $\tau_{k,1}, \tau_{k,2}$, respectively, $\mathbf{q}_{\beta_k}(\gamma_k)$ is a beta distribution, and $\mathbf{q}_{z_{mn}}(\phi_{mn})$ is a multinomial distribution. Following the work in [2], the approximation is built on truncated stick-breaking construction at $K$, which has been proved to closely approximate a true DP as long as $K$ is chosen to be large enough [12]. Empirically $K$ may be initialized to some value from tens to hundreds based on the model complexity. The useless dimensions will gradually be pruned automatically.

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

$$\mathcal{L} = \mathbb{E}_q[\log\ \mathbf{p}(\mathbf{U},\mathbf{V},\mathbf{Y},\mathbf{z},\boldsymbol{\sigma},\boldsymbol{\lambda},\boldsymbol{\beta})] - \mathbb{E}_q\left[\log \mathbf{q}(\mathbf{U},\mathbf{V},\mathbf{z},\boldsymbol{\sigma},\boldsymbol{\lambda},\boldsymbol{\beta})\right], \tag{7}$$

where $\mathbb{E}_q[\cdot]$ denotes the expectation with respect to $\mathbf{q}$. Maximization is performed via iteratively updating each parameter by setting the derivative of $\mathcal{L}$ with respect to the parameter to zero while keeping other parameters fixed.

### 6.1. Update $\boldsymbol{\gamma}$, $\boldsymbol{\tau}$ and $\boldsymbol{\phi}$:

We follow the stick-breaking procedure by setting a large enough value of $K$ for truncated approximation. Let $\Omega$ denote the set of indices of the observed data. The parameters are updated as follows:

$$\begin{aligned}\gamma_{k,1} &= 1 + \sum\nolimits_{(m,n)\in\Omega}\phi_{mnk},\\ \gamma_{k,2} &= \alpha + \sum\nolimits_{(m,n)\in\Omega}\sum_{t=k+1}^{K}\phi_{mnt};\\ \tau_{k,1} &= a_0 + \frac{1}{2}\sum\nolimits_{(m,n)\in\Omega}\phi_{mnk},\\ \tau_{k,2} &= b_0 + \frac{1}{2}\sum\nolimits_{(m,n)\in\Omega}\phi_{mnk}\mathbb{E}_q[(y_{mn} - \mathbf{u}_{m\cdot}\mathbf{v}_{n\cdot}{}^T)^2]\end{aligned} \tag{8}$$

for $k = 1, \ldots, K$ and $(m, n) \in \Omega$, where

$$\begin{aligned}\phi_{mnk} \propto \exp\Big\{&\mathbb{E}_q[\log\beta_k] + \sum_{t=1}^{k-1}\mathbb{E}_q[\log(1-\beta_t)]\\ &- \frac{1}{2}\frac{\tau_{k,1}}{\tau_{k,2}}\mathbb{E}_q[(y_{mn} - \mathbf{u}_{m\cdot}\mathbf{v}_{n\cdot}{}^T)^2]\\ &- \frac{1}{2}[\log\tau_{k,2} - \psi(\tau_{k,1})]\Big\},\end{aligned} \tag{9}$$

$$\begin{aligned}\mathbb{E}_q[(y_{mn} - \mathbf{u}_{m\cdot}\mathbf{v}_{n\cdot}^T)^2] &= y_{mn}(y_{mn} - 2\mathbf{a}_{m\cdot}\mathbf{b}_{n\cdot}^T)\\ &\quad + \operatorname{tr}\left((\mathbf{a}_{m\cdot}^T\mathbf{a}_{m\cdot} + \boldsymbol{\Sigma}_m^u)(\mathbf{b}_{n\cdot}^T\mathbf{b}_{n\cdot} + \boldsymbol{\Sigma}_n^v)\right),\\ \mathbb{E}_q[\log\beta_k] &= \psi(\gamma_{k,1}) - \psi\left(\gamma_{k,1} + \gamma_{k,2}\right),\\ \mathbb{E}_q[\log(1-\beta_k)] &= \psi(\gamma_{k,2}) - \psi\left(\gamma_{k,1} + \gamma_{k,2}\right).\end{aligned} \tag{10}$$

Here $\psi$ denotes the digamma function.

**Remarks:** Note that after updating $\boldsymbol{\tau}$, we can easily obtain the mixing proportions $\boldsymbol{\theta}$ of the $K$ clusters. After normalization, we will prune those clusters with probabilities less than a predefined threshold, since a very small $\theta_k$ indicates that it is very unlikely for some entries to belong to the corresponding cluster. As such, components that do not play any significant role can be removed.

### 6.2. Update $\mathbf{a}_{m\cdot}$, $\mathbf{b}_{n\cdot}$, $\boldsymbol{\Sigma}_m^u$, $\boldsymbol{\Sigma}_n^v$ and $\boldsymbol{\eta}$:

We next update the parameters related to $\mathbf{U}$ and $\mathbf{V}$:

$$\begin{aligned}\mathbf{a}_{m\cdot}{}^T &= \boldsymbol{\Sigma}_m^u\cdot\left(\sum_{n:(m,n)\in\Omega}\sum_{k=1}^{K}\frac{\tau_{k,1}}{\tau_{k,2}}\left(y_{mn}\phi_{mnk}\mathbf{b}_{n\cdot}^T\right)\right)\\ \boldsymbol{\Sigma}_m^u &= \left[\sum_{n:(m,n)\in\Omega}\sum_{k=1}^{K}\frac{\tau_{k,1}}{\tau_{k,2}}\phi_{mnk}\left(\mathbf{b}_{n\cdot}^T\mathbf{b}_{n\cdot} + \boldsymbol{\Sigma}_n^v\right) + \boldsymbol{\Lambda}\right]^{-1}\end{aligned} \tag{11}$$

where $\boldsymbol{\Lambda} = \operatorname{diag}(\boldsymbol{\lambda})^{-1}$. The update for $\mathbf{b}_{n\cdot}$ is similar. We omit it due to space constraint. For $\boldsymbol{\eta}$, we know $\mathbb{E}_q[\lambda_r] = \frac{\eta_{r,2}}{\eta_{r,1}}$:

$$\mathbb{E}_q[\lambda_r] = \frac{2b_1 + \mathbf{a}_{\cdot r}^T\mathbf{a}_{\cdot r} + \sum_{m=1}^{M}(\boldsymbol{\Sigma}_m^u)_{rr} + \mathbf{b}_{\cdot r}^T\mathbf{b}_{\cdot r} + \sum_{n=1}^{N}(\boldsymbol{\Sigma}_n^v)_{rr}}{2a_1 + M + N}. \tag{12}$$

**Remarks:** If we find that $\mathbb{E}_q[\lambda_r]$ is less than some predefined threshold, we will delete the corresponding dimension $r$ and decrement $R$ by 1 before proceeding. The rationale behind this is that, with a zero mean in the prior for this column, a very small variance indicates that this column will shrink to zero and hence will not contribute to explaining the data.

By repeating the update steps above, we discard the scarcely used Gaussian components and the matrix dimensions while adjusting the free variational parameters to approximate the original distribution until convergence.

## 7. Experiments

In this section, we empirically compare the proposed AMF[1] model with seven state-of-the-art methods. There are non-Bayesian methods (CWM [19] and PCP [5]) and Bayesian methods (VBLR [1], BRPCA [6], PRMF [28], BRMF [29] and MoG-RPCA (MRPCA) [32]). For all the experiments we have conducted, the hyperparameters of AMF are fixed without further tuning: $a_0 = b_0 = 10^{-4}$, $a_1 = b_1 = 0.1, \alpha = 1$.

### 7.1. Synthetic Experiments

In this part, we first follow [18] and design three sets of synthetic experiments to compare the performance of all the above low-rank matrix factorization methods. For these three sets of experiments, we first add three different types of noise to the corresponding sets of ground-truth matrices. The noise details are shown in Table 1. And then we drop $20\%$ of the each input to test the robustness of these methods. For each experiment we generate 10 ground-truth low-rank matrices, each of which is denoted by $\mathbf{Y}_0 \in \mathbb{R}^{50\times 50}$, the product of two randomly generated low-rank matrices $\mathbf{U} \in \mathbb{R}^{50\times 4}$ and $\mathbf{V} \in \mathbb{R}^{50\times 4}$. We assume that each element in $\mathbf{U}$ and $\mathbf{V}$ follows a normal distribution $\mathcal{N}(0,1)$ and the ground-truth rank is $r = 4$ for all $\mathbf{Y}_0$.

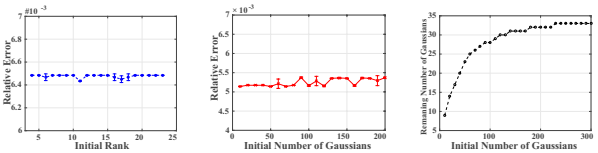
| | $\mathcal{N}(0, 0.5^2)$ | $U[-5, 5]$ | $U[-2, 2]$ |
|---|---|---|---|
| Gaussian Noise | 100% | 0 | 0 |
| Sparse Noise | 0 | 30% | 0 |
| Mixture Noise | 15% | 20% | 20% |

Table 1. Three types of noise. $U$ denotes the uniform noise followed by its range.

The recovered low-rank matrices are denoted by $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$. In each set of experiment, we first compare these eight methods under two different initial ranks. Considering that some methods are incapable of tuning the rank automatically, we initialize $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$ to have the ground-truth rank $R = 4$, and then we set the initial rank to be twice the ground-truth rank, $R = 8$, since the ground-truth rank is generally unknown *a priori* in real-world applications. For the number of Gaussian components $K$ in AMF and MRPCA, we initially set it to 64 which is large enough to accommodate various types of noise. With the same settings, we later drop $20\%$ data entries of each input and repeat the experiments. Note that MRPCA, PCP, BRPCA and BRMF are not designed to handle missing data, so we replace MRPCA with its previous version MoG [18] which is claimed to be able to deal with missing data. We only compare AMF, MoG, CWM, VBLR and PRMF in this part. For performance comparison, we use the relative error of the Frobenius norm with respect to the ground truth, defined as $\frac{\|\mathbf{Y}_0 - \bar{\mathbf{U}}\bar{\mathbf{V}}^T\|_F}{\|\mathbf{Y}_0\|_F}$. For each noise setting, we run 10 times with different input $\mathbf{Y}_0$ and record their performance. The mean values of the relative error are reported in Table 2.

From Table 2, when given complete input, we can see that AMF outperforms all other methods by giving the smallest relative error under three different types of noise, even though no prior knowledge is available about the number of components and the correct rank. Even when $20\%$ of the input entries are corrupted, AMF shows distinguishing robustness with small relative errors. As a full Bayesian model, AMF achieves comparable speed, saving the cost for hyper-parameter tuning and multiple trials for determining a proper number of clusters or the matrix rank. When the noise type is simple, Gaussian for example, MRPCA, BRPCA and VBLR with corrupted input achieve comparable results as AMF's. However, their performance fades next to AMF when the noise gets more complex. This attributes to AMF's high flexibility to model unknown complex noise and to correctly detect the rank.

We now shift our focus to AMF only and empirically investigate its intrinsic properties by varying the rank $R$ and the initial number of Gaussians $K$. We decouple the effect of the two parameters by varying only one of them at a time. We first fix $R = 4$ which is exactly the ground-truth value and vary $K$ from 10 to 200. Then we fix $K = 64$ but change $R$ from 4 to 23. Each set of experiments is conducted based on the same ground-truth matrix as input. For each distinct setting of $R$ and $K$, we run AMF five times with random initialization. The mean and standard deviation for each setting are recorded in Fig. 3.




(a) Vary $R$ with $K = 64$ (b) Vary $K$ with $R = 4$ (c) Remaining clusters

Figure 3. Intrinsic Analysis of AMF

In Fig. 3(a) and 3(b), the mean relative error of AMF almost stays unchanged with nearly indiscernible error bars when either one of the parameters is varied. While in

[1] AMF codes are available at http://peixianc.me/research.html

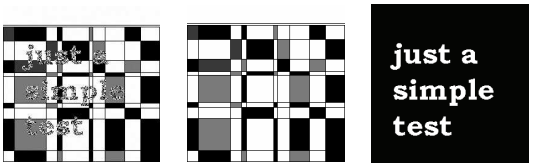
| Noise | R | AMF | MRPCA/MoG | CWM | PCP | VBLR | BRPCA | BRMF | PRMF |
|---|---|---|---|---|---|---|---|---|---|
| Gaussian | 4 | **0.0451**(0.33) | **0.0451**(0.26) | 0.0576(0.11) | 0.0751(0.57) | 0.0758(0.08) | **0.0451**(1.54) | 0.0545(1.25) | 0.0563(0.86) |
| | 8 | **0.0500**(0.37) | 0.0519(1.56) | 0.0777(0.15) | 0.0914(1.82) | 0.0875(0.08) | 0.0514(1.90) | 0.0747(1.24) | 0.0888(0.93) |
| | 4(w/m) | **0.0468**(0.45) | 0.0517(5.55) | 0.0572(0.27) | - | **0.0468**(0.15) | - | - | 0.0604(0.48) |
| | 8(w/m) | **0.0485**(0.49) | 0.0847(7.08) | 0.0738(0.71) | - | **0.0485**(0.16) | - | - | 0.0853(0.51) |
| Sparse | 4 | **6.9E-7**(1.31) | 1.9E-6 (0.48) | 0.0920(0.23) | 0.1342(0.52) | 0.0645(0.07) | 0.0354(1.31) | 0.0320(1.16) | 0.0519(0.77) |
| | 8 | **7.9E-7**(1.51) | 3.1E-6(0.61) | 0.1880(0.64) | 0.1471(1.48) | 0.0433(0.07) | 0.0797(1.80) | 0.1856(1.24) | 0.2359(0.89) |
| | 4(w/m) | **1.9E-5**(0.87) | 0.0513(1.24) | 0.0024(0.28) | - | 0.3016(0.13) | - | - | 0.0273(0.31) |
| | 8(w/m) | **2.2E-5**(0.89) | 0.2821(4.11) | 0.1722(0.87) | - | 0.3085(0.17) | - | - | 0.2219(0.45) |
| Mixture | 4 | **0.0050**(1.17) | 0.0052(1.13) | 0.0754(0.29) | 0.1318(0.76) | 0.2837(0.11) | 0.0096(1.92) | 0.0276(1.28) | 0.1025(0.23) |
| | 8 | **0.0062**(1.27) | 0.0150(1.60) | 0.2152(0.74) | 0.1653(1.54) | 0.2848(0.14) | 0.0442(2.56) | 0.1994(1.36) | 0.2468(0.24) |
| | 4(w/m) | **0.0272**(1.21) | 0.0730(1.61) | 0.1245(0.63) | - | 0.3246(0.18) | - | - | 0.1636(0.26) |
| | 8(w/m) | **0.0311**(2.11) | 0.4427(3.41) | 0.3265(0.38) | - | 0.3053(0.17) | - | - | 0.3413(0.25) |

Table 2. Mean relative error of eight methods under three types of noise with different initial ranks. Numbers in brackets are the corresponding time records (in second) of these methods. (w/m) indicates the situation with missing input. Best results are shown in bold.

Fig. 3(c), as the initial number of Gaussians $K$ is varied, the number of components remaining always stays around 33. Both these two experiments show that AMF is quite stable no matter how the initialization and parameters are set.

### 7.2. Text Removal

We next follow [29] to conduct a text removal simulation experiment. The 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 $256 \times 256$ with the corresponding data matrix of rank 10. Fig. 4 shows the input image (Fig. 4(a)) which is formed from the (background) clean image (Fig. 4(b)) and the (foreground) outlier mask (Fig. 4(c)).




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

Figure 4. Image for text removal simulation

As the true rank is often unknown beforehand in real-world data, we set the maximum rank of the initial $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$ to be twice the true rank for all the algorithms. We replace BRMF by MBRMF since, as demonstrated in [29], the latter performs better in dealing with contiguous outliers. Fig. 5 demonstrates the text recovered and the reconstructed background images using different algorithms.

By visually examining the recovered masks in Fig. 5, we note that both masks by AMF and MBRMF appear to be most sharp. The masks recovered by MRPCA , PRMF and PCP come close. The one by BRPCA is recognizable but the results of CWM and VBLR are quite fuzzy. With regard to reconstruction of the low-rank matrices, AMF gives the cleanest recovered background, followed by MBRMF, PCP and MRPCA. PRMF and BRPCA leave behind too many outliers although they do a fair job in outlier detection. To some extent CWM fails both tasks while VBLR performs slightly better in recovering the background.

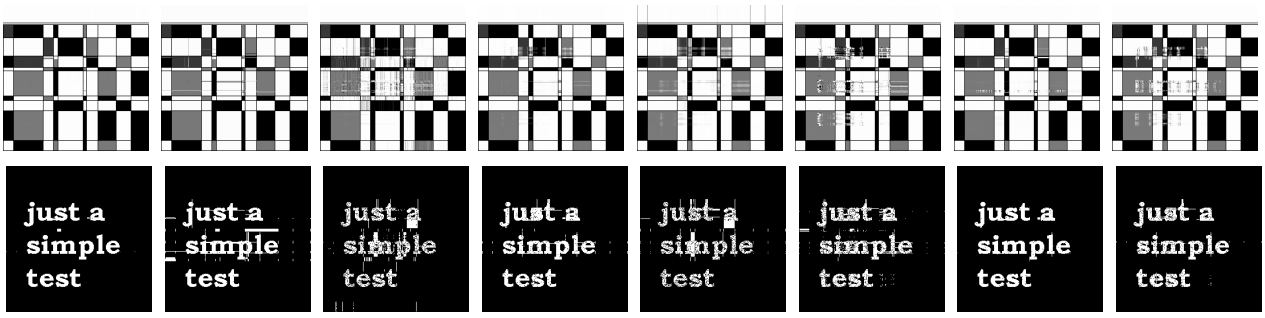
Table 3 shows the quantitative results. On detecting outliers, although MBRMF yields highest AUC values, AMF comes very close. With regard to background recovery, AMF obviously gains lower relative error than the other methods, showing that the quantitative results are in line with the qualitative results above.

| | AMF | MRPCA | CWM | PCP | VBLR | BRPCA | MBRMF | PRMF |
|---|---|---|---|---|---|---|---|---|
| AUC | 0.991 | 0.954 | 0.867 | 0.976 | 0.899 | 0.921 | **0.993** | 0.960 |
| RE | **0.068** | 0.101 | 0.185 | 0.103 | 0.163 | 0.205 | 0.102 | 0.144 |

Table 3. Comparison of different methods. We use the *area under curve* (AUC) to measure outlier mask detection and the relative error of the Frobenius norm for background recovery. Best results are shown in bold.

### 7.3. Face Shadow Removal

In this section, we study a real application using face images captured under varying illumination. Such a face shadow removal task is often applied as an important preprocessing step by many face recognition systems. Sources of the noise, such as low illumination, self-shadowing and specularity, pose great challenges to this task. We use the images of two persons from the Extended Yale B database [10] for illustration. For each person, we use all the 64 images in the database. Since well-aligned face images of the same person under varying illumination lie very close to a 4-dimensional linear subspace, the rank is set to 8 for all the methods with the exception of PCP which is allowed to choose the rank automatically. First we use the original faces as input and compare all the eight methods. Then for each face, we randomly drop half of the pixels and




(a) AMF (b) MRPCA (c) CWM (d) PCP (e) VBLR (f) BRPCA (g) MBRMF (h) PRMF

Figure 5. Background and foreground masks recovered by different algorithms

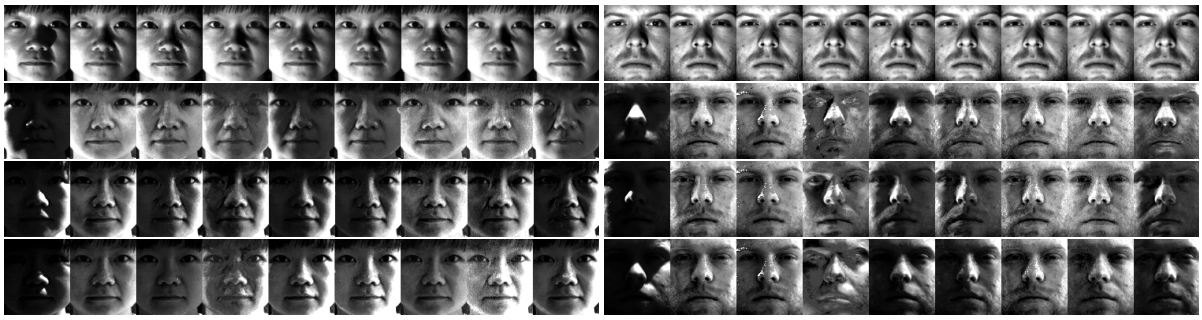


Figure 6. Face shadow removal results. Each of the nine groups from left to right shows the original face and those recovered by AMF, MRPCA, CWM, PCP, VBLR, BRPCA, MBRMF, and PRMF.

use these corrupted images as input. Since only AMF, MRPCA, CWN, VBLR and PRMF are claimed to be able to handle missing data, we give the results of these five algorithms. Fig. 6 7 shows some of the results. By observing

Figure 7. Face shadow removal results with input corrupted. Each of the six groups from left to right shows the original face and those recovered by AMF, MoG, CWM, VBLR and PRMF.

the first row of Fig. 6, we can see that all the algorithms show good performance under normal situations in eliminating the shadow on the lady's face and removing the eye glint from the man. However, in extreme cases when a large portion of the face is shadowed, AMF outperforms the other methods by recovering the face as much as possible while preserving the features of the original face. Although MRPCA and BRPCA can sometimes obtain comparable results, their effects are often influenced by artifacts especially for images of the man. Most results from MBRMF are noisy and unable to preserve such features as the beard of the man. Also in Fig. 7, AMF shows impressive robustness though half of the pixel entries are corrupted. Compared to the other four methods, AMF is capable of recovering the original face with much less noise. These results again demonstrate the necessity for decomposing the noise sources and AMF has great potential for applications in such situations.

## 8. Conclusion and Future Work

We have proposed a novel non-parametric Bayesian method for matrix factorization. It addresses two crucial model selection issues by placing an ARD prior and a DP prior on the factor matrices and the noise model, respectively. Based on the stick-breaking representation of DP, we have devised an efficient variational inference algorithm. From the experimental results, we have demonstrated that the proposed method has high potential to handle a wide range of applications with automatic model selection.

To take this work further, as in [34, 29], we also want to exploit the fact that contiguous outliers often occur in many computer vision applications. This implies some clustering structure of the noise pattern in the matrix. Representing this property explicitly in the model may lead to further performance improvement. We will explore this research direction in our future work.

## 9. Acknowledgement

This research has been partially supported by research grant FSGRF14EG36.

## 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] M. Blei and I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

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

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

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

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

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

[9] T. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, pages 209–230, 1973.

[10] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[11] 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.

[12] H. Ishwaran and L. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453), 2001.

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

[14] N. D. Lawrence and R. Urtasun. Non-linear matrix factorization with Gaussian processes. In *ICML*, pages 601–608, 2009.

[15] D. J. MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

[16] D. J. MacKay. Probable networks and plausible predictions–a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995.

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

[18] D. Meng and F. De la Torre. Robust matrix factorization with unknown noise. In *ICCV*, 2013.

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

[20] R. M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

[21] J. Pitman. Combinatorial stochastic processes. Technical report, Springer, 2002.

[22] C. Rasmussen. The infinite Gaussian mixture model. In *NIPS*, volume 12, pages 554–560, 1999.

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

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

[25] J. Sethuraman. A constructive definition of Dirichlet priors. Technical report, DTIC Document, 1991.

[26] Y. Shen, Z. Wen, and Y. Zhang. Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optimization Methods and Software*, pages 1–25, 2012.

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

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

[29] N. Wang and D.-Y. Yeung. Bayesian robust matrix factorization for image and video processing. In *ICCV*, 2013.

[30] S. Wang, D. Liu, and Z. Zhang. Nonconvex relaxation approaches to robust matrix recovery. In *IJCAI*, pages 1764–1770, 2013.

[31] J. Xu, V. Ithapu, L. Mukherjee, J. Rehg, and V. Singh. GOSUS: Grassmannian online subspace updates with structured-sparsity. In *ICCV*, 2013.

[32] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and L. Zhang. Robust principal component analysis with complex noise. 2014.

[33] 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.

[34] 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.

[35] Z. Zhou, A. Wagner, H. Mobahi, J. Wright, and Y. Ma. Face recognition with contiguous occlusion using Markov random fields. In *ICCV*, pages 1050–1057, 2009.