Robust Principal Component Analysis with Complex Noise

Qian Zhao[†] TIMMY.ZHAOQIAN@GMAIL.COM
Deyu Meng[†] DYMENG@MAIL.XJTU.EDU.CN
Zongben Xu[†] ZBXU@MAIL.XJTU.EDU.CN
Wangmeng Zuo[§] CSWMZUO@GMAIL.COM
Lei Zhang[‡] CSLZHANG@COMP.POLYU.EDU.HK

[†]School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China
[§]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[‡]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

Abstract

The research on robust principal component analysis (RPCA) has been attracting much attention recently. The original RPCA model assumes sparse noise, and use the L_1-norm to characterize the error term. In practice, however, the noise is much more complex and it is not appropriate to simply use a certain L_p-norm for noise modeling. We propose a generative RPCA model under the Bayesian framework by modeling data noise as a mixture of Gaussians (MoG). The MoG is a universal approximator to continuous distributions and thus our model is able to fit a wide range of noises such as Laplacian, Gaussian, sparse noises and any combinations of them. A variational Bayes algorithm is presented to infer the posterior of the proposed model. All involved parameters can be recursively updated in closed form. The advantage of our method is demonstrated by extensive experiments on synthetic data, face modeling and background subtraction.

1. Introduction

As a classical and popular tool for data analysis, principal component analysis (PCA) has a wide range of applications in science and engineering (Jolliffe, 2002). Essentially, PCA seeks the best L_2-norm low-rank approximation of the given data matrix. However, L_2-norm is sensitive to gross noises and outliers, which are often introduced in data acquisition. Therefore, how to make PCA robust has been attracting much attention in the last decade (De la Torre & Black, 2003; Ke & Kanade, 2005; Ding et al., 2006; Kwak, 2008).

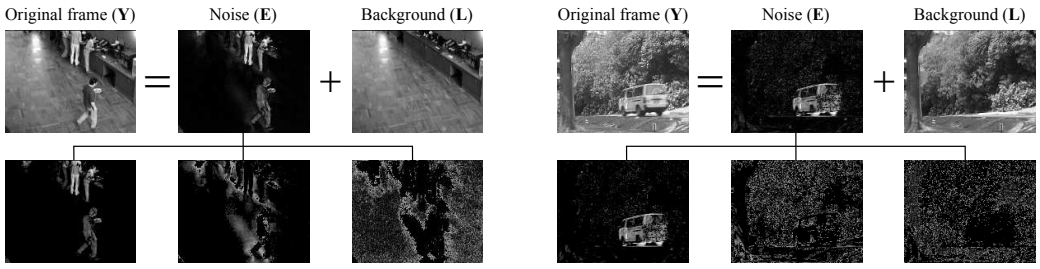
Motivated by the recent advances in low-rank matrix analysis (Candès & Recht, 2009; Candès & Tao, 2010; Recht et al., 2010), the so-called *robust principal component analysis* (RPCA) (Wright et al., 2009) has been proposed to decompose a given data matrix into a low-rank matrix and a sparse matrix. Denote by $\mathbf{Y} \in \mathbb{R}^{m\times n}$ the original data matrix, by $\mathbf{L} \in \mathbb{R}^{m\times n}$ the low-rank component and by $\mathbf{E} \in \mathbb{R}^{m\times n}$ the sparse component, RPCA can be mathematically described as the following convex optimization problem:

$$\min_{\mathbf{L},\mathbf{E}} \ \|\mathbf{L}\|_* + \lambda\|\mathbf{E}\|_1 \quad s.t. \ \mathbf{Y} = \mathbf{L} + \mathbf{E}, \qquad (1)$$

where $\|\mathbf{L}\|_* = \sum_r \sigma_r(\mathbf{L})$ denotes the nuclear norm of $\mathbf{L}$, $\sigma_r(\mathbf{L})$ $(r = 1, 2, \ldots, \min(m, n))$ is the r^{th} singular value of $\mathbf{L}$, $\|\mathbf{E}\|_1 = \sum_{ij} |e_{ij}|$ denotes the L_1-norm of $\mathbf{E}$ and e_{ij} is the element in the i^{th} row and j^{th} column of $\mathbf{E}$. Under certain noise sparsity and rank upper-bound assumptions, it has been proved that one can exactly recover $\mathbf{L}$ and $\mathbf{E}$ from $\mathbf{Y}$ with high probability (Candès et al., 2011). RPCA has been successfully applied to many machine learning and computer vision problems, such as video surveillance (Wright et al., 2009), face modeling (Peng et al., 2010) and subspace clustering (Liu et al., 2010).

RPCA, however, still has clear limitations. As shown in Eq. (1), it utilizes L_1-norm to characterize $\mathbf{E}$, which is only optimal for Laplacian noise. Although L_1-norm can better fit sparse noise than L_2-norm, the real noise is often neither Gaussian nor Laplacian, but has much more complex statistical structures. For example, the *Bootstrap* and *Campus* sequences (Li et al., 2004) shown in Figure 1 can be reasonably modeled as the sum of a low-rank part (background) and a noise part (foreground). However, such noise has a rather complex structure. As can be seen from Figure 1, the noise in the *Bootstrap* sequence can be decomposed into:

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.



(a) *Bootstrap* sequence (b) *Campus* sequence

Figure 1. Background subtraction by the proposed MoG-RPCA method: (a) *Bootstrap* sequence; (b) *Campus* sequence. First row (from left to right): original frame; noise component; low-rank component. Second row: three noise components obtained by the proposed MoG-RPCA method.

moving objects (people) in the foreground, shadows alongside objects, and background noise. The noise in the *Campus* sequence can also be separated into three layers with different degrees of variations: moving object (bus), variations of tree leaves and shadows, and background noise. Clearly, in these real scenarios, it is not appropriate to simply use L_1-norm or L_2-norm to model the noise, which assumes Laplacian or Gaussian noise, respectively.

This paper presents a new RPCA approach, which can fit more complex noise. We formulate the problem as a generative model under the Bayesian framework, and model data noise as a mixture of Gaussians (MoG). We then employ the variational inference method to infer the posterior. Since MoG is a universal approximator to any continuous probability distribution (Bishop, 2006; Meng & De la Torre, 2013), the proposed MoG-RPCA approach is capable of adapting a much wider range of real noises than the current RPCA methods.

2. Related Work

Early attempts to solve the RPCA problem replace the L_2-norm error by some robust losses. De la Torre & Black (2003) utilized the Geman-McClure function in robust statistics to improve the robustness of PCA; Ding et al. (2006) used a smoothed R_1-norm to this end; Kwak (2008) introduced the L_1-norm variance and designed an efficient algorithm to optimize it. These methods, however, are sensitive to initialization, and only perform well on Laplacian-like noise.

In recent years, low-rank matrix analysis methods have been rapidly developed. Wright et al. (2009) initially formulated the RPCA model as shown in Eq. (1). Some variants have also been proposed, e.g., Xu et al. (2010) used the $L_{1,2}$-norm to handle data corrupted by column. The iterative thresholding method (Candès et al., 2011) was proposed to solve the RPCA model. This method, however, converges very slow. To speed up the computation, Lin et al. proposed the accelerated proximal gradient (APG) (Lin et al., 2009) and the augmented Lagrangian multiplier (ALM) (Lin et al., 2010) methods. ALM leads to state-of-the-art performance in terms of both speed and accuracy.

Bayesian approaches to RPCA have also been investigated. Ding et al. (2011) modeled the singular values of $\mathbf{L}$ and the entries of $\mathbf{E}$ with beta-Bernoulli priors, and used a Markov chain Monte Carlo (MCMC) sampling scheme to perform inference. This method needs many sampling iterations, always hampering its practical use. Babacan et al. (2012) adopted the automatic relevance determination (ARD) approach to model both $\mathbf{L}$ and $\mathbf{E}$, and utilized the variational Bayes (VB) method to do inference. This method is more computationally efficient. These Bayesian methods, however, assume a certain noise prior (a sparse noise plus a dense noise), which cannot always effectively model the diverse types of noises occurring in practice.

One problem closely related to RPCA is L_1-norm low-rank matrix factorization (LRMF), which aims to factorize a matrix into the product of two smaller matrices under the L_1 measure. Ke & Kanade (2005) proposed to solve it via alternative linear/quadratic programming (ALP/AQP). Eriksson & van den Hengel (2010) designed an L_1-Wiberg approach by extending the classical Wiberg method to L_1 minimization. Zheng et al. (2012) proposed a RegL1ALM method by using convex trace-norm regularization to improve convergence. Wang et al. (2012) considered the problem in a probabilistic framework, and solved it via conditional EM algorithm. Meng et al. (2013) proposed a novel cyclic weighted median method to efficiently solve the problem. The L_1-norm LRMF problem can be viewed as a fixed-rank variant of RPCA. However, the L_1-norm LRMF model is not convex and it can be trapped to local optima. Moreover, the rank needs to be pre-specified in this line of research, which is often unavailable in practice.

3. RPCA with MoG Noise

3.1. Model Formulation

Let's consider RPCA as a generative model:

$$\mathbf{Y} = \mathbf{L} + \mathbf{E}. \tag{2}$$

If we assume that the entries of $\mathbf{E}$ are drawn independently from a Laplacian distribution, and the singular values of $\mathbf{L}$

are drawn from another Laplacian distribution, we can obtain the RPCA model (1) by performing maximum a posteriori (MAP) estimation on $\mathbf{L}$. Clearly, we can interpret RPCA as a MAP estimation problem with Laplacian noise. However, real noises are more complicated. To improve RPCA, a natural idea is to use MoG to model noise since MoG is a universal approximator to any continuous distributions (Bishop, 2006). For example, a Gaussian is a special case of MoG and a Laplacian can be expressed as a scaled MoG (Andrews & Mallows, 1974). Similar noise modeling strategy was also adopted by Meng & De la Torre (2013) for LRMF problem.

Noise Component Modeling. We assume that each e_{ij} in $\mathbf{E}$ follows a MoG distribution:

$$e_{ij} \sim \sum\nolimits_{k=1}^{K} \pi_k \mathcal{N}(e_{ij}|\mu_k, \tau_k^{-1}), \tag{3}$$

where π_k is the mixing proportion with $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$, K is the Gaussian components number and $\mathcal{N}(e|\mu, \tau^{-1})$ denotes the Gaussian distribution with mean μ and precision τ. Eq. (3) can be equivalently expressed as a two-level generative model by introducing the indicator variables z_{ijk}s (Bishop, 2006):

$$\begin{aligned} e_{ij} &\sim \prod\nolimits_{k=1}^{K} \mathcal{N}(e_{ij}|\mu_k, \tau_k^{-1})^{z_{ijk}}, \\ \mathbf{z}_{ij} &\sim \text{Multinomial}(\mathbf{z}_{ij}|\boldsymbol{\pi}), \end{aligned} \tag{4}$$

where $\mathbf{z}_{ij} = (z_{ij1}, \ldots, z_{ijK}) \in \{0,1\}^K$, $\sum_{k=1}^{K} z_{ijk} = 1$ and $\mathbf{z}_{ij}$ follows a multinomial distribution parameterized by $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$. To complete the Bayesian model, we introduce conjugate priors over the parameters of Gaussian components, μ_ks, τ_ks, and the mixing proportions, $\boldsymbol{\pi}$, as:

$$\begin{aligned} \mu_k, \tau_k &\sim \mathcal{N}(\mu_k|\mu_0, (\beta_0\tau_k)^{-1})\text{Gam}(\tau_k|c_0, d_0), \\ \boldsymbol{\pi} &\sim \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0), \end{aligned} \tag{5}$$

where $\text{Gam}(\tau|c_0, d_0)$ is the Gamma distribution with parameters c_0 and d_0, and $\text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0)$ denotes the Dirichlet distribution parameterized by $\boldsymbol{\alpha}_0 = (\alpha_{01}, \ldots, \alpha_{0K})$.

Low-rank Component Modeling. One simple way to model the low-rank component $\mathbf{L}$ is to impose a Laplacian prior over the singular values of $\mathbf{L}$. Another way is to incorporate the beta-Bernoulli priors on the singular values, resulting in exact zeros on most singular values (Ding et al., 2011). In this paper, we adopt the ARD for low-rank component modeling (Babacan et al., 2012) due to its fast speed and good scalability.

We formulate $\mathbf{L} \in \mathbb{R}^{m\times n}$ with rank $l \leq \min(m, n)$ as the product of $\mathbf{U} \in \mathbb{R}^{m\times R}$ and $\mathbf{V} \in \mathbb{R}^{n\times R}$:

$$\mathbf{L} = \mathbf{U}\mathbf{V}^T = \sum\nolimits_{r=1}^{R} \mathbf{u}_{\cdot r}\mathbf{v}_{\cdot r}^T, \tag{6}$$

where $R > l$, and $\mathbf{u}_{\cdot r}$ ($\mathbf{v}_{\cdot r}$) is the r^{th} column of $\mathbf{U}$ ($\mathbf{V}$). Our goal is to achieve column sparsity in $\mathbf{U}$ and $\mathbf{V}$, such that some columns in $\mathbf{U}$ and $\mathbf{V}$ will approach zeros. The low-rank nature of $\mathbf{L}$ can then be guaranteed. This goal can be achieved by imposing the following priors on $\mathbf{U}$ and $\mathbf{V}$:

$$\mathbf{u}_{\cdot r} \sim \mathcal{N}(\mathbf{u}_{\cdot r}|\mathbf{0}, \gamma_r^{-1}\mathbf{I}_m), \quad \mathbf{v}_{\cdot r} \sim \mathcal{N}(\mathbf{v}_{\cdot r}|\mathbf{0}, \gamma_r^{-1}\mathbf{I}_n), \tag{7}$$

where $\mathbf{I}_m$ denotes the $m\times m$ identity matrix. The conjugate prior on each precision variable γ_r is:

$$\gamma_r \sim \text{Gam}(\gamma_r|a_0, b_0). \tag{8}$$

Note that each column pair $\mathbf{u}_{\cdot r}$, $\mathbf{v}_{\cdot r}$ of $\mathbf{U}$, $\mathbf{V}$ has the same sparsity profile characterized by the common precision variable γ_r. It has been validated that such a modeling could lead to large precision values of some γ_rs, and hence result in a good low-rank estimate of $\mathbf{L}$ (Babacan et al., 2012).

Combining Eqs. (2), (4)-(8) together, we can construct the full Bayesian model of RPCA with MoG noise, denoted by MoG-RPCA. The goal turns to infer the posterior of all involved variables:

$$p(\mathbf{U}, \mathbf{V}, \boldsymbol{\mathcal{Z}}, \boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\gamma}|\mathbf{Y}), \tag{9}$$

where $\boldsymbol{\mathcal{Z}} = \{\mathbf{z}_{ij}\}$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$, $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_K)$, and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_R)$.

3.2. Remarks

Capability of MoG-RPCA in Fitting Sparse Noise: Since the original RPCA model was designed to deal with sparse noise, we need to evaluate if the proposed MoG-RPCA can also handle sparse noise. Consider a MoG distribution with two components:

$$p(x) = \pi\mathcal{N}(x|\mu_1, \tau_1^{-1}) + (1-\pi)\mathcal{N}(x|\mu_2, \tau_2^{-1}). \tag{10}$$

Let $\mu_1 = 0$ and $\tau_1^{-1} = 0$. Eq. (10) degenerates to:

$$p(x) = \pi\delta(0) + (1-\pi)\mathcal{N}(x|\mu_2, \tau_2^{-1}), \tag{11}$$

where $\delta(0)$ is the Dirac delta distribution concentrated at 0. This distribution can be equivalently described by the generative model as:

$$\begin{aligned} x &\sim z\delta(0) + (1-z)\mathcal{N}(x|\mu_2, \tau_2^{-1}), \\ z &\sim \text{Bernoulli}(\pi), \end{aligned} \tag{12}$$

where $\text{Bernoulli}(\pi)$ is the Bernoulli distribution with parameter π, implying that the variable x is zero with probability π. This distribution is actually the spike-and-slab prior, from which sparsity can be naturally confirmed (Ishwaran & Rao, 2005). The spike-and-slab prior can thus be viewed as a special case of MoG, that is, the MoG-RPCA is also capable of fitting sparse noise, as can be easily observed from Figure 2.

Merits of MoG-RPCA: The first merit of MoG-RPCA is that it is capable of fitting a much wider range of noises than the traditional RPCA models. Besides Laplacian,

Gaussian, spike-and-slab distributions or any combinations of them, our model can handle more complicated noises. The second merit is that all the parameters involved in the proposed model, including $\mathbf{U}$, $\mathbf{V}$ and the rank of them, γ_rs, z_{ijk}s, π_ks, μ_ks and τ_ks, can be automatically inferred from the observed data under easy non-informative settings of hyperparameters. Another merit of our model is that instead of assuming zero-mean data noise in traditional RPCA methods, we leave μ_ks, the means of all noise components, as to-be-estimated parameters, which further enhances the adaptability of our model to real asymmetric noise. All these merits will be extensively substantiated by our experiments.

3.3. Variational Inference

We use the variational Bayes (VB) (Bishop, 2006) method to infer the posterior of MoG-RPCA. VB seeks an approximation distribution $q(\mathbf{x})$ to the true posterior $p(\mathbf{x}|\mathcal{D})$ ($\mathcal{D}$ denotes the observed data) by solving the following variational optimization:

$$\min_{q\in\mathcal{C}} \mathrm{KL}(q\|p) = -\int q(\mathbf{x}) \ln\left\{\frac{p(\mathbf{x}|\mathcal{D})}{q(\mathbf{x})}\right\} \mathrm{d}\mathbf{x}, \tag{13}$$

where $\mathrm{KL}(q\|p)$ denotes the KL divergence between $q(\mathbf{x})$ and $p(\mathbf{x}|\mathcal{D})$, and $\mathcal{C}$ denotes the set of probability densities with certain restrictions to make the minimization tractable. Taking $q(\mathbf{x}) = \prod_i q_i(\mathbf{x}_i)$, the closed-form solution to $q_j(\mathbf{x}_j)$, with other factors fixed, can be attained by:

$$q_j^*(\mathbf{x}_j) = \frac{\exp\left\{\langle \ln p(\mathbf{x},\mathcal{D})\rangle_{\mathbf{x}\setminus\mathbf{x}_j}\right\}}{\int \exp\left\{\langle \ln p(\mathbf{x},\mathcal{D})\rangle_{\mathbf{x}\setminus\mathbf{x}_j}\right\} \mathrm{d}\mathbf{x}_j}, \tag{14}$$

where $\langle\cdot\rangle$ denotes the expectation, and $\mathbf{x}\setminus\mathbf{x}_j$ denotes the set of $\mathbf{x}$ with $\mathbf{x}_j$ removed. Eq. (13) can be solved by alternatively calculating (14).

Let's approximate the posterior distribution (9) with the following factorized form:

$$\begin{aligned} q(\mathbf{U},\mathbf{V},\boldsymbol{\mathcal{Z}},\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{\pi},\boldsymbol{\gamma}) = \prod\nolimits_i q(\mathbf{u}_{i\cdot}) \prod\nolimits_j q(\mathbf{v}_{j\cdot}) \\ \prod\nolimits_{ij} q(\mathbf{z}_{ij}) \prod\nolimits_k q(\mu_k,\tau_k) q(\boldsymbol{\pi}) \prod\nolimits_r q(\gamma_r), \end{aligned} \tag{15}$$

where $\mathbf{u}_{i\cdot}$ ($\mathbf{v}_{j\cdot}$) is the i^{th} (j^{th}) row of $\mathbf{U}$ ($\mathbf{V}$). Then we can analytically infer all the factorized distributions involved in Eq. (15) as below. The computational details are given in the supplementary material.

Estimation of Noise Component: The parameters involved in the noise component are $\boldsymbol{\mu}$, $\boldsymbol{\tau}$, $\boldsymbol{\mathcal{Z}}$ and $\boldsymbol{\pi}$. Based on the prior imposed in Eq. (5) and its conjugate property, we can get the following update equation for each μ_k, τ_k ($k = 1,\ldots,K$):

$$q(\mu_k,\tau_k) = \mathcal{N}(\mu_k|m_k,(\beta_k\tau_k)^{-1})\mathrm{Gam}(\tau_k|c_k,d_k), \tag{16}$$

where

$$\begin{aligned} \beta_k &= \beta_0 + \textstyle\sum_{ij}\langle z_{ijk}\rangle, \\ m_k &= \frac{1}{\beta_k}(\beta_0\mu_0 + \textstyle\sum_{ij}\langle z_{ijk}\rangle(y_{ij} - \langle \mathbf{u}_{i\cdot}\rangle\langle \mathbf{v}_{j\cdot}\rangle^T)), \\ c_k &= c_0 + \frac{1}{2}\textstyle\sum_{ij}\langle z_{ijk}\rangle, \\ d_k &= d_0 + \frac{1}{2}\{\textstyle\sum_{ij}\langle z_{ijk}\rangle\langle (y_{ij} - \mathbf{u}_{i\cdot}\mathbf{v}_{j\cdot}^T)^2\rangle + \beta_0\mu_0^2 \\ &\quad - \frac{1}{\beta_k}(\textstyle\sum_{ij}\langle z_{ijk}\rangle(y_{ij} - \langle \mathbf{u}_{i\cdot}\rangle\langle \mathbf{v}_{j\cdot}\rangle^T) + \beta_0\mu_0)^2\}. \end{aligned}$$

Similarly, it is easy to obtain the update equation for mixing proportions $\boldsymbol{\pi}$:

$$q(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}), \tag{17}$$

where $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_K)$, $\alpha_k = \alpha_{0k} + \sum_{ij}\langle z_{ijk}\rangle$.

The variational posterior for the indicators $\boldsymbol{\mathcal{Z}}$ can also be derived in closed form:

$$q(\mathbf{z}_{ij}) = \prod\nolimits_k {r_{ijk}}^{z_{ijk}}, \tag{18}$$

where

$$\begin{aligned} r_{ijk} &= \frac{\rho_{ijk}}{\sum_k \rho_{ijk}}, \\ \rho_{ijk} &= \frac{1}{2}\langle \ln\tau_k\rangle - \frac{1}{2}\ln 2\pi - \frac{1}{2}\langle\tau_k\rangle\langle (y_{ij} - \mathbf{u}_{i\cdot}\mathbf{v}_{j\cdot}^T - \mu_k)^2\rangle \\ &\quad + \langle \ln\pi_k\rangle. \end{aligned}$$

Estimation of Low-rank Component: The parameters involved in the low-rank component are $\mathbf{U}$, $\mathbf{V}$ and $\boldsymbol{\gamma}$. For each row $\mathbf{u}_{i\cdot}$ of $\mathbf{U}$, using the factorization (15), we can get

$$q(\mathbf{u}_{i\cdot}) = \mathcal{N}(\mathbf{u}_{i\cdot}|\boldsymbol{\mu}_{\mathbf{u}_{i\cdot}},\boldsymbol{\Sigma}_{\mathbf{u}_{i\cdot}}), \tag{19}$$

with mean $\boldsymbol{\mu}_{\mathbf{u}_{i\cdot}}$ and covariance $\boldsymbol{\Sigma}_{\mathbf{u}_{i\cdot}}$ given by

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{u}_{i\cdot}}^T &= \boldsymbol{\Sigma}_{\mathbf{u}_{i\cdot}}\left\{\textstyle\sum_k\langle\tau_k\rangle\sum_j\langle z_{ijk}\rangle(y_{ij} - \langle\mu_k\rangle)\langle\mathbf{v}_{j\cdot}\rangle\right\}^T, \\ \boldsymbol{\Sigma}_{\mathbf{u}_{i\cdot}} &= \left\{\textstyle\sum_k\langle\tau_k\rangle\sum_j\langle z_{ijk}\rangle\langle\mathbf{v}_{j\cdot}^T\mathbf{v}_{j\cdot}\rangle + \boldsymbol{\Gamma}\right\}^{-1}, \end{aligned}$$

where $\boldsymbol{\Gamma} = \mathrm{diag}(\langle\boldsymbol{\gamma}\rangle)$. Similarly, for each row $\mathbf{v}_{j\cdot}$ of $\mathbf{V}$, we have

$$q(\mathbf{v}_{j\cdot}) = \mathcal{N}(\mathbf{v}_{j\cdot}|\boldsymbol{\mu}_{\mathbf{v}_{j\cdot}},\boldsymbol{\Sigma}_{\mathbf{v}_{j\cdot}}), \tag{20}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{v}_{j\cdot}}^T &= \boldsymbol{\Sigma}_{\mathbf{v}_{j\cdot}}\left\{\textstyle\sum_k\langle\tau_k\rangle\sum_i\langle z_{ijk}\rangle(y_{ij} - \langle\mu_k\rangle)\langle\mathbf{u}_{i\cdot}\rangle\right\}^T, \\ \boldsymbol{\Sigma}_{\mathbf{v}_{j\cdot}} &= \left\{\textstyle\sum_k\langle\tau_k\rangle\sum_i\langle z_{ijk}\rangle\langle\mathbf{u}_{i\cdot}^T\mathbf{u}_{i\cdot}\rangle + \boldsymbol{\Gamma}\right\}^{-1}. \end{aligned}$$

For $\boldsymbol{\gamma}$ which controls the rank of $\mathbf{L}$, we have

$$q(\gamma_r) = \mathrm{Gam}(\gamma_r|a_r,b_r), \tag{21}$$

where

$$a_r = a_0 + \frac{m+n}{2}, \quad b_r = b_0 + \frac{1}{2}\left(\langle \mathbf{u}_{\cdot r}^T \mathbf{u}_{\cdot r}\rangle + \langle \mathbf{v}_{\cdot r}^T \mathbf{v}_{\cdot r}\rangle\right).$$

As discussed in Babacan et al. (2012), some γ_rs tend to be very large during the inference process and the corresponding $\mathbf{u}_{\cdot r}$ and $\mathbf{v}_{\cdot r}$ will be removed from $\mathbf{U}$ and $\mathbf{V}$. The low-rank property of $\mathbf{L}$ can thus be achieved.

Setting of the Hyperparameters: We set all the hyperparameters involved in our model in a *non-informative* manner to make them influence as less as possible the inference of posterior distributions (Bishop, 2006). Throughout our experiments, we set $\mu_0 = 0$, and $\alpha_{01}, \ldots, \alpha_{0K}$, β_0, a_0, b_0, c_0, d_0 a small value 10^{-6}. Our method performs stably well on all experiments with these easy settings.

Tuning of the Number of Gaussians: We use a simple but effective method to automatically tune the number of Gaussians K. We first run the proposed MoG-RPCA method with a relatively large K. Then we check if there exist two analogous noise components (with means μ_i, μ_j and variances τ_i^{-1}, τ_j^{-1}, respectively) which satisfy that both $|\mu_i - \mu_j|/(|\mu_i| + |\mu_j|)$ and $|\tau_i^{-1} - \tau_j^{-1}|/(\tau_i^{-1} + \tau_j^{-1})$ are smaller than a preset threshold. If yes, we set K to $K-1$ and re-run our method by taking current parameters as initializations and combining the two analogous Gaussian components into one. Otherwise we terminate the iteration and output the result. Such a strategy is efficient since the information obtained in previous step is fully utilized as initialization to the next step. In all our experiments, we simply set the maximum K as 6, and the experimental results showed that it is flexible enough to fit the noises in all synthetic and real data we used.

Complexity: It is easy to see that only simple computations are involved in the variational inference of parameters, except that inferring each of $\mathbf{u}_{i\cdot}$s and $\mathbf{v}_{j\cdot}$s needs to invert a $R \times R$ matrix, leading to $\mathcal{O}((m+n)R^3)$ costs in total. Altogether, the complexity of MoG-RPCA is $\mathcal{O}((m+n)R^3 + KmnR + mnR^2)$ per iteration, where m, n, K, R are the dimensionality and size of the input data, the MoG number, and the rank presetting, respectively. The cost of our method is thus linear in both data dimensionality and size, which is comparable to the existing RPCA algorithms.

4. Experiments

We evaluate the performance of the proposed MoG-RPCA method on synthetic, face and video data. The competing methods include classic PCA and representative RPCA and LRMF methods: RPCA (Wright et al., 2009)[1], BRPCA (Ding et al., 2011)[2], VBRPCA (Babacan et al., 2012)[3], ALP (Ke & Kanade, 2005), RegL1ALM (Zheng et al., 2012)[4] and PRMF (Wang et al., 2012)[5]. We wrote the code for ALP and utilized the "svd" function in Matlab for PCA. All experiments were implemented in Matlab on a PC with 2.60GHz CPU and 8GB RAM.

4.1. Synthetic Simulations

Ten sets of synthetic data were generated to evaluate the performance of MoG-RPCA with different types of noises. In the first five sets of simulations, we randomly generated 20 matrices with size 100×100 and rank $r = 5$. Each of these matrices was generated by the product of two smaller matrices as $\mathbf{U}\mathbf{V}^T$. Both $\mathbf{U}$ and $\mathbf{V}$ are of sizes $100 \times r$, and their entries were independently drawn from $\mathcal{N}(0,1)$. We further added certain types of noise to the ground truth matrix as follows. (1) *No noise* added. (2) *Sparse noise*: 10% entries mixed with uniform noise within $[-25, 25]$. (3) *Gaussian noise*: all entries mixed with Gaussian noise $\mathcal{N}(0, 0.05)$. (4) *Mixture noise with zero mean*: 10% entries mixed with uniform noise between $[-25, 25]$, 20% with Gaussian noise $\mathcal{N}(0,1)$, and the other 70% with Gaussian noise $\mathcal{N}(0, 0.01)$. (5) *Mixture noise with nonzero mean*: 10% entries mixed with uniform noise within $[-15, 35]$, 30% with Gaussian noise $\mathcal{N}(0.1, 1)$, and the other 60% with Gaussian noise $\mathcal{N}(-0.1, 0.01)$. The other five sets of simulations were similarly constructed except for setting rank $r = 10$.

Two criteria were utilized for performance assessment. (1) *Relative reconstruction error* (RRE): $\|\hat{\mathbf{L}} - \mathbf{L}\|_F / \|\mathbf{L}\|_F$, where $\mathbf{L}$ and $\hat{\mathbf{L}}$ denote the ground truth and reconstructed low-rank matrices, respectively. (2) *Estimated rank* (ER): the rank of $\hat{\mathbf{L}}$[6]. The performance of each competing method on each simulation was evaluated as the average over the 20 matrices in terms of RRE and ER, as listed in Table 1.

We can see from Table 1 that, unsurprisingly, PCA has the best performance in the no noise and Gaussian noise cases, and the L_1-norm based methods perform better in the case of sparse noise. While our method always has comparable performance in these situations, its advantage tends to be significant in the case of more complex noise, which is demonstrated by the following Mann-Whitney-Wilcoxon test (Mann & Whitney, 1947). In Gaussian noise experi-

[1] http://perception.csl.illinois.edu/matrix-rank/sample_code.html

[2] http://people.ee.duke.edu/~lcarin/BCS.html

[3] http://www.dbabacan.info/publications.html

[4] https://sites.google.com/site/yinqiangzheng/

[5] http://winsty.net/prmf.html

[6] Since in the LRMF methods, PCA, ALP, RegL1ALM and PRMF, the rank is fixed as a pre-specified parameter, this criterion is not applicable to this line of methods.

Table 1. Performance evaluation on synthetic data. The best results in terms of RRE are highlighted in bold.

$\mathrm{rank}(\mathbf{L}) = 5$		PCA	RPCA	BRPCA	VBRPCA	ALP	RegL1ALM	PRMF	MoG-RPCA
No Noise	RRE	**1.04e-15**	3.33e-8	0.232	5.11e-4	1.96e-4	7.21e-9	1.50e-5	5.03e-5
	ER	-	5	5	5	-	-	-	5
	Time(s)	0.0019	0.0728	24.89	0.0113	3.27	0.117	0.272	0.101
Sparse Noise	RRE	0.768	1.33e-7	6.60e-2	0.117	4.21e-4	**4.01e-8**	6.12e-5	8.17e-5
	ER	-	5	5	5	-	-	-	5
	Time(s)	0.0045	0.135	23.42	0.0337	3.71	0.316	0.315	0.264
Gaussian Noise	RRE	**3.11e-2**	5.95e-2	**3.11e-2**	4.98e-2	3.83e-2	7.07e-2	3.88e-2	**3.11e-2**
	ER	-	57	5	5	-	-	-	5
	Time(s)	0.0040	0.178	52.02	0.0767	5.75	0.496	0.561	0.258
Mixture Noise (zero mean)	RRE	7.72e-2	4.96e-2	3.27e-2	8.65e-2	2.69e-2	6.36e-2	4.73e-2	**1.90e-2**
	ER	-	58	5	5	-	-	-	5
	Time(s)	0.0038	0.173	18.11	0.0679	5.99	0.513	0.563	0.417
Mixture Noise (nonzero mean)	RRE	1.11	8.63e-2	6.64e-2	0.762	4.80e-2	8.54e-2	5.16e-2	**2.41e-2**
	ER	-	58	5	2	-	-	-	5
	Time(s)	0.0036	0.173	17.96	0.0675	6.47	0.500	0.542	0.538
$\mathrm{rank}(\mathbf{L}) = 10$		PCA	RPCA	BRPCA	VBRPCA	ALP	RegL1ALM	PRMF	MoG-RPCA
No Noise	RRE	**1.82e-15**	1.73e-8	0.193	1.20e-3	3.42e-4	9.06e-9	1.57e-5	1.52e-4
	ER	-	10	10	10	-	-	-	10
	Time(s)	0.0021	0.0917	42.64	0.0220	3.80	0.143	0.351	0.162
Sparse Noise	RRE	0.778	3.33e-3	7.81e-2	0.984	5.20e-4	**4.76e-8**	7.12e-5	8.41e-5
	ER	-	11	10	1	-	-	-	10
	Time(s)	0.0045	0.190	41.76	0.119	5.19	0.394	0.690	0.550
Gaussian Noise	RRE	**3.12e-2**	4.99e-2	3.13e-2	4.78e-2	3.80e-2	9.07e-2	3.93e-2	3.13e-2
	ER	-	57	10	10	-	-	-	10
	Time(s)	0.0039	0.176	89.27	0.110	7.35	0.562	0.620	0.313
Mixture Noise (zero mean)	RRE	0.775	6.25e-2	2.55e-2	0.959	3.30e-2	7.02e-2	4.60e-2	**2.08e-2**
	ER	-	58	10	1	-	-	-	10
	Time(s)	0.0037	0.172	29.67	0.0838	8.83	0.563	0.620	1.20
Mixture Noise (nonzero mean)	RRE	1.04	0.101	7.58e-2	1	5.74e-2	0.125	7.65e-2	**2.65e-2**
	ER	-	57	10	0	-	-	-	10
	Time(s)	0.039	0.170	29.61	0.0842	9.30	0.568	0.619	1.37

Table 2. Quantitative comparison of the ground truth (denote by "True") noise probability density functions and those estimated (denote by "Est.") by the MoG-RPCA method in the synthetic experiments.

$\mathrm{rank}(\mathbf{L}) = 5$		No Noise	Sparse		Gaussian	Mixture (zero mean)			Mixture (nonzero mean)		
		Comp. 1	Comp. 1	Comp. 2	Comp. 1	Comp. 1	Comp. 2	Comp. 3	Comp. 1	Comp. 2	Comp. 3
π_k	True	-	0.1	0.9	-	0.1	0.2	0.7	0.1	0.3	0.6
	Est.	-	0.10	0.90	-	0.107	0.183	0.710	0.10	0.28	0.62
μ_k	True	0	0	0	0	0	0	0	10	0.1	-0.1
	Est.	-4.56e-5	-0.48	3.68e-6	-0.016	-0.24	0.017	-2.33e-3	10.25	0.094	-0.10
τ_k^{-1}	True	0	208.3	0	0.05	208.3	1	0.01	208.3	1	0.01
	Est.	5.05e-4	209.6	1.27e-4	0.049	199.5	1.02	0.011	200.1	1.07	0.012
$\mathrm{rank}(\mathbf{L}) = 10$		No Noise	Sparse		Gaussian	Mixture (zero mean)			Mixture (nonzero mean)		
		Comp. 1	Comp. 1	Comp. 2	Comp. 1	Comp. 1	Comp. 2	Comp. 3	Comp. 1	Comp. 2	Comp. 3
π_k	True	-	0.1	0.9	-	0.1	0.2	0.7	0.1	0.3	0.6
	Est.	-	0.10	0.90	-	0.107	0.178	0.715	0.10	0.27	0.63
μ_k	True	0	0	0	0	0	0	0	10	0.1	-0.1
	Est.	2.66e-5	-0.16	-3.21e-8	-0.013	-0.088	0.022	1.75e-3	10.32	0.10	-0.098
τ_k^{-1}	True	0	208.3	0	0.05	208.3	1	0.01	208.3	1	0.01
	Est.	2.77e-4	206.9	1.56e-4	0.050	197.8	1.01	0.012	202.8	1.13	0.013

ments, the performance of MoG-RPCA is not significantly different from BRPCA (two-sided test: p-value $= 0.9705$) and PCA (two-sided test: p-value $= 0.9705$), which is already optimal for Gaussian noise. These three methods, however, perform significantly better than the other competing methods (p-value $< 10^{-5}$). In the case of sparse noise, our method significantly outperforms the competing methods (p-value < 0.005) except for PRMF and RegL1ALM, which are specifically designed for sparse noise. However, in the case of mixture noise, which is more realistic in practice, it is statistically significant that the proposed MoG-RPCA performs better than all other methods (p-value $< 10^{-5}$). As for rank estimation, the proposed method can accurately estimate the underground rank of the ground truth matrix in all cases. This further verifies the effectiveness of the proposed method.

The good performance of the proposed MoG-RPCA method in complex noise cases can be easily explained by Table 2 and Figure 2, which compare the ground truth noises and those estimated by our method. It is easy to see that the estimated noise distributions comply with the real ones very well. Especially, in the cases of complicated mixture noise (the 3^{rd} and 4^{th} columns of Figure 2), our method very faithfully resolves the real noise configurations.

We also compared the average CPU times for all methods in Table 1. As can be seen, our method costs comparable CPU time with other methods in most situations, which complies with the complexity analysis in Section 3.3. Considering its superiority in complex noise adaption and automatic rank estimation, it is reasonable to say that our method is efficient.

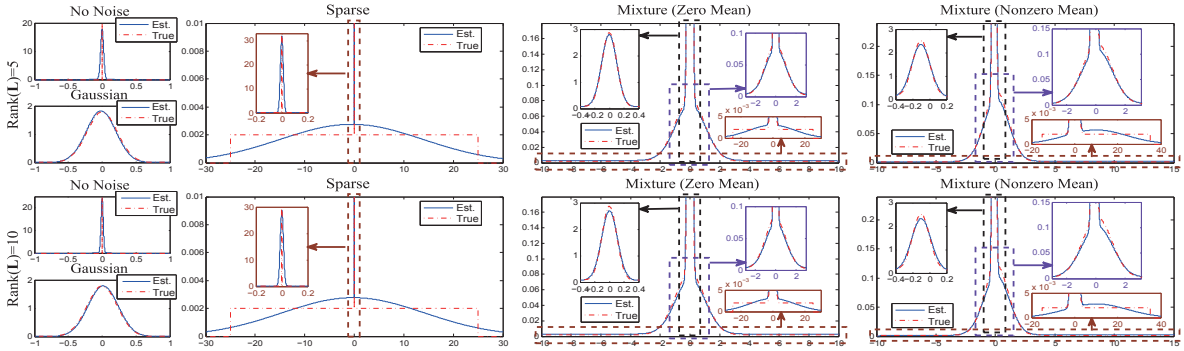


Figure 2. Visual comparison of the ground truth (denote by "True") noise probability density functions and those estimated (denote by "Est.") by the MoG-RPCA method in the synthetic experiments. The embedded sub-figures depict the zoom-in of the indicated portions.

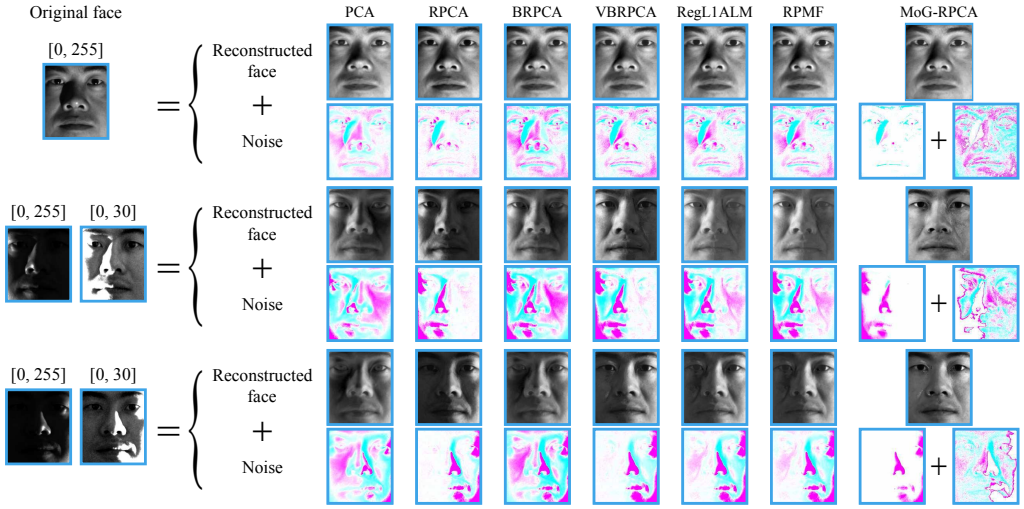


Figure 3. From left to right: original faces, reconstructed faces and extracted noise by PCA, RPCA, BRPCA, VBRPCA, RegL1ALM, PRMF and MoG-RPCA. The noises with positive and negative values are depicted in purple and blue, respectively. The 2nd and 3rd faces are also shown in range $[0, 30]$ to better visualize the camera noise in the dark region of faces. This figure should be viewed in color and the details are better seen by zooming on a computer screen.

4.2. Face Modeling

This experiment aims to test the effectiveness of MoG-RPCA in face modeling applications. The second subset of the Extended Yale B database (Georghiades et al., 2001), consisting of 64 faces of one subject with size 192×168, was used to generate the data matrix with size 32256×64. Typical images are depicted in Figure 3. All competing methods were implemented, except for ALP which encounters the "out of memory" problem. We set the rank as 4 (Basri & Jacobs, 2003) for the factorization-based methods, including PCA, RegL1ALM and PRMF. For the other competing methods, the rank was automatically learned from data. The reconstructed faces and the extracted noises by all methods are compared in Figure 3.

The proposed method, as well as the other competing methods, is able to remove the cast shadows and saturations in faces. Our method, however, performs better on faces with a large dark region. Such face images contain both significant cast shadow and saturation noises, which correspond to the highly dark and bright areas in face, and camera/read noise (Nakamura, 2005) which is much amplified in the dark areas. It is very interesting that the proposed method is capable of accurately extracting these two kinds of noises, as clearly depicted in Figure 3. The better noise fitting capability of the proposed method thus leads to better face reconstruction performance.

4.3. Background Subtraction

Background subtraction from video sequences captured by a static camera can be modeled as a low-rank matrix analysis problem (Wright et al., 2009; Candès et al., 2011). Four commonly utilized video sequences, including two indoor scenes (*Bootstrap* and *Hall*) and two outdoor scenes (*Fountain* and *Campus*), provided by Li et al. (2004)[7], were adopted in our experiments. We extracted 400 frames from the *Fountain* sequence and 600 frames from each of the oth-

[7] http://perception.i2r.a-star.edu.sg/bk_model/bk_index

Figure 4. From left to right: original video frames, background extracted by PCA, RPCA, BRPCA, VBRPCA, RegL1ALM, PRMF and MoG-RPCA, together with their extracted noise images.

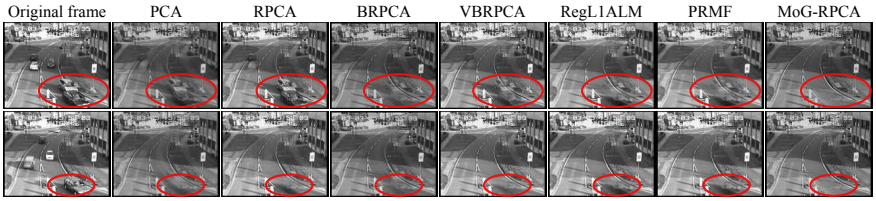


Figure 5. From left to right: original video frames, background extracted by PCA, RPCA, BRPCA, VBRPCA, RegL1ALM, PRMF and MoG-RPCA. The foreground areas are demarcated for easy comparison.

er three sequences. All the competing methods except ALP were implemented for comparison. For the factorization-based methods, including PCA, RegL1ALM and PRMF, several rank parameters were tried and the best one was recorded. For the other methods, the rank was learned from data. The results for typical sample frames are shown in Figures 1 and 4.

It can be seen that all the competing methods can extract the background from videos with slight differences in visualization. Our method, however, can extract more elaborate foreground (noise) information. More specifically, our method can discover three levels of foreground information with different variations from the *Hall* and *Bootstrap* videos: moving people, shadows alongside the foreground objects and background noise, and from the *Fountain* and *Campus* videos: moving people/bus, variations of fountain/tree leaves, and background noise.

We also tested our method on a real traffic video sequence acquired by a static surveillance camera[8], which is more challenging due to diverse variations of its foreground cars. Typical background frames extracted by competing methods are shown in Figure 5. It is easy to see that our method more clearly removes foreground information and attains a better subtraction of background. This further substantiates the effectiveness of the proposed method.

[8] http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html

5. Conclusion

We proposed a new RPCA method by modeling noise as a MoG distribution under the Bayesian framework. Compared with the current RPCA methods, which assume certain noise distribution (e.g., Gaussian or sparse noise) on data, our method can perform the RPCA task under more complex noises. The effectiveness of our method was demonstrated by synthetic data with artificial noises and by face modeling and background subtraction problems with real noises. The proposed method shows clear advantages over previous methods on its capability in accurately recovering the low-rank structure and elaborately extracting the multimodal noise configuration from observed data.

Acknowledgments

This research was supported by 973 Program of China with No. 3202013CB329404, the NSFC projects with No. 61373114, 11131006, 61075054, 61271093, and HK RGC GRF grant (PolyU 5313/13E).

References

Andrews, D. F. and Mallows, C. L. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B*, 36(1):99–102, 1974.

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

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

Bishop, C. M. *Pattern recognition and machine learning*. Springer New York, 2006.

Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9 (6):717–772, 2009.

Candès, E. J. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of the ACM*, 58(3):11:1–11:37, 2011.

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

Ding, C., Zhou, D., He, X., and Zha, H. R_1-PCA: Rotational invariant L_1-norm principal component analysis for robust subspace factorization. In *ICML*, 2006.

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

Eriksson, A. and van den Hengel, A. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the L_1 norm. In *CVPR*, 2010.

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

Ishwaran, H. and Rao, J. S. Spike and slab variable selection: Frequentist and Bayesian strategies. *Annals of Statistics*, 33 (2):730–773, 2005.

Jolliffe, I. T. *Principal component analysis*. Springer series in statistics. Springer, New York, 2nd edition, 2002.

Ke, Q. and Kanade, T. Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *CVPR*, 2005.

Kwak, N. Principal component analysis based on L_1-norm maximization. *IEEE Transactions on PAMI*, 30(9):1672–1680, 2008.

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

Lin, Z., Ganesh, A., Wright, J., Wu, L., Chen, M., and Ma, Y. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. Technical Report UILU-ENG-09-2214, UIUC, 2009.

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

Liu, G., Lin, Z., and Yu, Y. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.

Mann, H. B. and Whitney, D. R. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18(1):50–60, 1947.

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

Meng, D., Xu, Z., Zhang, L., and Zhao, J. A cyclic weighted median method for L_1 low-rank matrix factorization with missing entries. In *AAAI*, 2013.

Nakamura, J. *Image Sensors and Signal Processing for Digital Still Cameras*. CRC Press, Boca Raton, 2005.

Peng, Y., Ganesh, A., Wright, J., Xu, W., and Ma, Y. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *CVPR*, 2010.

Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

Wang, N., Yao, T., Wang, J., and Yeung, D. A probabilistic approach to robust matrix factorization. In *ECCV*, 2012.

Wright, J., Peng, Y., Ma, Y., Ganesh, A., and Rao, S. Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. In *NIPS*, 2009.

Xu, H., Caramanis, C., and Sanghavi, S. Robust PCA via outlier pursuit. In *NIPS*, 2010.

Zheng, Y., Liu, G., Sugimoto, S., Yan, S., and Okutomi, M. Practical low-rank matrix approximation under robust L_1-norm. In *CVPR*, 2012.