SPARSE BAYESIAN LEARNING FOR ROBUST PCA

Jing Liu, Yacong Ding, and Bhaskar Rao

Department of Electrical and Computer Engineering,
University of California at San Diego, La Jolla, CA 92093, USA

ABSTRACT

In this paper, we propose a new Bayesian model to solve the Robust PCA problem - recovering the underlying low-rank matrix and sparse matrix from their noisy compositions. We first derive and analyze a new objective function, which is proven to be equivalent to the fundamental minimizing "rank+sparsity" objective. To solve this objective, we develop a concise Sparse Bayesian Learning (SBL) method that has minimum assumptions and effectively deals with the crux of the problem. The concise modeling allows simple and effective Empirical Bayesian inference via MAP-EM. Simulation studies demonstrate the superiority of the proposed method over the existing state-of-the-art methods. The efficacy of the method is further verified through a text extraction image processing task.

Index Terms— Robust PCA, Sparse Bayesian Learning, low-rank matrix, sparse matrix

1. INTRODUCTION

Recovering the low-rank matrix $\boldsymbol{L}$ and sparse matrix $\boldsymbol{E}$ from their composition $\boldsymbol{M}$ (usually with additional noise) has received a lot of interest in the past decade. This problem is known as Robust PCA, and was first studied in the noiseless case [1][2][3]. The underlying optimization problem is [2]:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \text{rank}(\boldsymbol{L}) + \lambda\|\boldsymbol{E}\|_0 \quad s.t.\ \|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}\|_F \le \delta, \quad (1)$$

When $\delta = 0$, the problem reduces to the noiseless case. Using the SVD of $\boldsymbol{L}$, i.e., $\boldsymbol{L} = \boldsymbol{U}\text{diag}(\boldsymbol{s})\boldsymbol{V}^T$, the problem in (1) is equivalent to the following:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{s}\succeq 0,\boldsymbol{E}} \|\boldsymbol{s}\|_0 + \lambda\|\boldsymbol{E}\|_0 \quad (2)$$
$$s.t.\ \|\boldsymbol{M}-\boldsymbol{U}\text{diag}(\boldsymbol{s})\boldsymbol{V}^T-\boldsymbol{E}\|_F \le \delta,\ \boldsymbol{U},\boldsymbol{V}\text{orthonormal}.$$

Denote $\boldsymbol{m} = \text{vec}(\boldsymbol{M})$, $\boldsymbol{e} = \text{vec}(\boldsymbol{E})$, and $\boldsymbol{A}_i = \text{vec}(\boldsymbol{U}_i\boldsymbol{V}_i^T)$, where $\boldsymbol{A}_i, \boldsymbol{U}_i$ and $\boldsymbol{V}_i$ denote the i-th column of $\boldsymbol{A}, \boldsymbol{U}$ and $\boldsymbol{V}$ respectively, (2) can be written in the following vector form:

$$\min_{\boldsymbol{A},\boldsymbol{s}\succeq 0,\boldsymbol{e}} \|\boldsymbol{s}\|_0 + \lambda\|\boldsymbol{e}\|_0 \quad s.t.\ \|\boldsymbol{m}-\boldsymbol{A}\boldsymbol{s}-\boldsymbol{e}\|_2 \le \delta,$$
$$\boldsymbol{A}_i = \text{vec}(\boldsymbol{U}_i\boldsymbol{V}_i^T),\ \forall i,\ \boldsymbol{U},\boldsymbol{V}\text{orthonormal}. \quad (3)$$

This research was supported in part by the Ericsson endowed chair funds.

It is known that the optimization problem in (1) is NP-hard. To make the problem computationally viable, [1][2][3][4] suggest relaxing the rank minimization to nuclear norm minimization and the ℓ_0-'norm' penalty to an ℓ_1-norm penalty. This is known as Principal Component Pursuit (PCP) [3] in the noiseless case, and Stable Principal Component Pursuit (SPCP)[4] in the noisy case:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{E}\|_1 \quad s.t.\ \|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}\|_F \le \delta, \quad (4)$$

which is equivalent to

$$\min_{\boldsymbol{A},\boldsymbol{s}\succeq 0,\boldsymbol{e}} \|\boldsymbol{s}\|_1 + \lambda\|\boldsymbol{e}\|_1 \quad s.t.\ \|\boldsymbol{m}-\boldsymbol{A}\boldsymbol{s}-\boldsymbol{e}\|_2 \le \delta,$$
$$\boldsymbol{A}_i = \text{vec}(\boldsymbol{U}_i\boldsymbol{V}_i^T),\ \forall i,\ \boldsymbol{U},\boldsymbol{V}\text{orthonormal}. \quad (5)$$

Interestingly, one can recover both low-rank matrix and sparse matrix exactly (or stably) under certain conditions by solving (4). However, from a robust linear regression viewpoint (dealing with sparse outliers $\boldsymbol{e}$), recent progress [5][6] shows that the Sparse Bayesian Learning (SBL) [7] approach provides a much better solution to the ℓ_0-'norm' problem than the ℓ_1 convex relaxation approach when the underlying $\boldsymbol{A}$ is given. The superior performance of SBL is also well known in the broader Sparse Signal Recovery (SSR) community [8][9]. So the question is: can we leverage the advantage of SBL to solve the Robust PCA problem?

It is worth mentioning that our recently proposed method SRPCP [10] uses a genuine ℓ_0-'norm' on the sparse matrix $\boldsymbol{E}$ and has provable recovery guarantees, but it still has to relax the rank minimization objective to the nuclear norm on the low rank matrix $\boldsymbol{L}$.

There have already been several Sparse Bayesian Learning methods proposed for solving the Robust PCA problem. The earliest work [11] proposed to model the low-rank matrix as $\boldsymbol{L} = \boldsymbol{D}(\text{diag}(\boldsymbol{z})\text{diag}(\boldsymbol{s}))\boldsymbol{W}$, and sparse matrix as $\boldsymbol{E} = \boldsymbol{B}\circ\boldsymbol{X}$, where $\boldsymbol{z}$ and $\boldsymbol{B}$ have binary entries obeying a Bernoulli distribution, and the hyper-parameter of the Bernoulli distribution is further assumed to be Beta distributed. The $\boldsymbol{s}$, $\boldsymbol{X}$ and noise $\boldsymbol{N}$ are drawn from Gaussian distribution with corresponding precision (inverse of the variance) parameters generated from different Gamma distributions. Finally, the columns of $\boldsymbol{D}$ and $\boldsymbol{W}$ are assumed Gaussian distributed.

Babacan et al. [12] proposed a slightly simpler model, where the low-rank matrix $\boldsymbol{L} = \boldsymbol{A}\boldsymbol{B}^T$, and the columns of

$\boldsymbol{A}$ and $\boldsymbol{B}$ are drawn from a Gaussian distribution with each precision parameter drawn from a Gamma distribution. The elements of the sparse matrix are simply drawn independently from a Gaussian distribution.

Recently, Wipf [13] proposed a even simpler model that directly assumes the low-rank matrix $\boldsymbol{L}$ to be Gaussian and proposed to learn its covariance matrix, while the sparse matrix is modeled similar to Babacan's work [12]. Some improvement over the convex PCP has been demonstrated. In [14], a modification to the model in [13] is made and the resulting method demonstrates much better performance than the convex PCP method and Bayesian approaches. However, though the method starts with a Bayesian setting, the complexity of the inference procedure forces compromises, leading to the framework to be used as a means to approximate and obtain an interesting objective function for minimization.

So far, the power of the SBL does not seem to have been fully brought to bear on this problem. The main difficulty of the current Bayesian approaches is the need to infer many parameters from the assumed distributions. Too many assumptions limit the generalization of these methods to different practical situations. Another challenge is the difficulty of inference with such complicated probabilistic models. Usually MCMC sampling or Variational Bayesian approximation have to be used. An issue of interest in this paper is, can we provide a simple model and derive a concise SBL approach that has minimum assumptions while effectively dealing with the crux of the problem? Also can such a model be supported by a simple and effective inference procedure? In this paper, we answer these questions in the affirmative.

Notation: $\text{vec}(\boldsymbol{A}) \in \mathbb{R}^{n_1 n_2 \times 1}$ is a vector obtained by stacking columns of $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_2}$, whereas $\text{Mat}(\boldsymbol{h}) \in \mathbb{R}^{n_1 \times n_2}$ is a matrix obtained by the reverse operation on vector $\boldsymbol{h} \in \mathbb{R}^{n_1 n_2 \times 1}$.

2. SPARSE BAYESIAN LEARNING OBJECTIVE

Before presenting the method, let us first consider the fundamental problem that our Bayesian approach attempts to solve:

$$\min_{\boldsymbol{A},\boldsymbol{s},\boldsymbol{e}} \|\boldsymbol{m} - \boldsymbol{A}\boldsymbol{s} - \boldsymbol{e}\|_2^2 + \lambda_1 \|\boldsymbol{s}\|_0 + \lambda_2 \|\boldsymbol{e}\|_0$$
$$s.t.\ \boldsymbol{A}_i = \text{vec}(\boldsymbol{U}_i \boldsymbol{V}_i^T), \|\boldsymbol{U}_i\|_2 = \|\boldsymbol{V}_i\|_2 = 1, \forall i, \quad (6)$$

which is the Lagrange form of

$$\min_{\boldsymbol{A},\boldsymbol{s},\boldsymbol{e}} \|\boldsymbol{s}\|_0 + \lambda \|\boldsymbol{e}\|_0 \ s.t.\ \|\boldsymbol{m} - \boldsymbol{A}\boldsymbol{s} - \boldsymbol{e}\|_2 \leq \delta,$$
$$\boldsymbol{A}_i = \text{vec}(\boldsymbol{U}_i \boldsymbol{V}_i^T),\ \|\boldsymbol{U}_i\|_2 = \|\boldsymbol{V}_i\|_2 = 1, \forall i. \quad (7)$$

Compared with (3), we have removed the *non-negative constraint* on $\boldsymbol{s}$ and the *orthogonality constraint* on $\boldsymbol{U}$ and $\boldsymbol{V}$. This makes our inference procedure much easier. More importantly, the following proposition guarantees that this simplification does not change the optimal solution in terms of $\boldsymbol{L} (= \boldsymbol{U}\text{diag}(\boldsymbol{s})\boldsymbol{V}^T)$ and $\boldsymbol{E}$. The proof is deferred to [15].

Proposition 1. *The optimization problems in (1), (2) and (7) have the same minimal objective value. Furthermore, they have the same global optimal solution(s) in terms of the low-rank matrix $\boldsymbol{L}$ and the sparse matrix $\boldsymbol{E}$, where $\boldsymbol{L} = \boldsymbol{U}diag(\boldsymbol{s})\boldsymbol{V}^T$ in (2) and (7).*

3. SPARSE BAYESIAN LEARNING MODEL

Now we present our SBL approach to tackle (6). Our observation model is

$$\boldsymbol{m} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} + \boldsymbol{n},\ s.t.\ \boldsymbol{A}_i = \text{vec}(\boldsymbol{U}_i \boldsymbol{V}_i^T), \|\boldsymbol{U}_i\|_2 = \|\boldsymbol{V}_i\|_2 = 1$$

Denote the parameter space of $\boldsymbol{A}$ which satisfies the above constraints/structure as $\mathscr{A}$. The distinguishing part of the proposed approach compared to the existing SBL approaches is that we assume $\boldsymbol{A}$ is a deterministic parameter that lies in the space $\mathscr{A}$, without assuming any distribution on it, which makes the proposed method more general.

Thanks to the removal of the non-negative constraint on $\boldsymbol{s}$ in (6) and Proposition 1, the remaining modeling can now directly follow the well-established SBL procedure. Assume $\boldsymbol{s} \sim \mathcal{N}(0, \boldsymbol{\Gamma}), \boldsymbol{\Gamma} \triangleq \text{diag}(\boldsymbol{\gamma})$. The outlier vector $\boldsymbol{e} \sim \mathcal{N}(0, \boldsymbol{\Lambda})$, $\boldsymbol{\Lambda} \triangleq \text{diag}(\boldsymbol{\alpha})$, so the elements of $\boldsymbol{e}$ are assumed to be independent and zero mean Gaussian, and their variances are to be learned. The noise $\boldsymbol{n} \sim \mathcal{N}(0, \beta \boldsymbol{I})$, and all the elements of $\boldsymbol{n}$ share the same variance β. The goal of SBL (evidence maximization) is to infer the unknown parameters[1] (e.g., $\hat{\boldsymbol{A}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\alpha}}$) from the data $\boldsymbol{m}$. Then $\boldsymbol{s}$ and $\boldsymbol{e}$ can be estimated via the posterior mean of the respective posterior distributions, i.e., $p(\boldsymbol{s}|\boldsymbol{m}, \hat{\boldsymbol{A}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\alpha}})$ and $p(\boldsymbol{e}|\boldsymbol{m}, \hat{\boldsymbol{A}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\alpha}})$.

For tractable derivation, define diagonal matrix $\boldsymbol{D} = (\boldsymbol{\Lambda} + \beta \boldsymbol{I})^{-1}$, and matrix $\boldsymbol{F} = (\boldsymbol{\Gamma}^{-1} + \boldsymbol{A}^T \boldsymbol{D} \boldsymbol{A})^{-1}$. We have that $\boldsymbol{m}$ is zero mean Gaussian vector with covariance matrix $\boldsymbol{\Sigma_m} = \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \boldsymbol{\Lambda} + \beta \boldsymbol{I}$, whose inverse is given by $\boldsymbol{\Sigma_m^{-1}} = (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \boldsymbol{\Lambda} + \beta \boldsymbol{I})^{-1} = \boldsymbol{D} - \boldsymbol{D}\boldsymbol{A}\boldsymbol{F}\boldsymbol{A}^T\boldsymbol{D}$.

The posterior distribution of $\boldsymbol{e}$ given $\boldsymbol{m}$ is Gaussian with

$$\boldsymbol{\mu_{e|m}} = \boldsymbol{\Lambda}\boldsymbol{\Sigma_m^{-1}}\boldsymbol{m}, \boldsymbol{\Sigma_{e|m}} = \boldsymbol{\Lambda} - \boldsymbol{\Lambda}\boldsymbol{\Sigma_m^{-1}}\boldsymbol{\Lambda}. \quad (8)$$

The posterior distribution of $\boldsymbol{s}$ given $\boldsymbol{m}$ is Gaussian with[2]

$$\boldsymbol{\mu_{s|m}} = \boldsymbol{\Gamma}\boldsymbol{A^T}\boldsymbol{\Sigma_m^{-1}}\boldsymbol{m}, \boldsymbol{\Sigma_{s|m}} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\boldsymbol{A^T}\boldsymbol{\Sigma_m^{-1}}\boldsymbol{A}\boldsymbol{\Gamma}. \quad (9)$$

The posterior cross-covariance between $\boldsymbol{s}$ and $\boldsymbol{e}$ given $\boldsymbol{m}$ is

$$\boldsymbol{\Sigma_{se|m}} = \boldsymbol{F}\boldsymbol{A}^T(\boldsymbol{I} - \beta \boldsymbol{D}). \quad (10)$$

4. PARAMETER ESTIMATION

Let $\Psi = (\boldsymbol{A}, \boldsymbol{\gamma}, \boldsymbol{\alpha})$ represents the whole parameter set that we want to estimate. Our goal is to maximize $p(\Psi|\boldsymbol{m}) \propto$

[1] In this work, we specify the value of β instead of inferring it.

[2] Directly plugging in the expression of $\Sigma_{\boldsymbol{m}}^{-1}$ greatly reduces complexity.

$p(\boldsymbol{m}|\Psi)p(\Psi)$. Here we restrict $\boldsymbol{A} \in \mathscr{A}$ and employ Inverse-gamma prior on each element of $\boldsymbol{\gamma}$, i.e., $p(\boldsymbol{\gamma}_i) = \text{IG}(a, b)$, with $b \to 0$, while do not assume any prior (or say use non-informative prior) on $\boldsymbol{\alpha}$. For the inference, we use the MAP-EM [16] procedure to optimize $p(\Psi|\boldsymbol{m})$.

In the E-step, we have the Q-function as

$$
\begin{aligned}
&Q(\Psi|\Psi^{(k)}) = Q(\boldsymbol{A}, \boldsymbol{\gamma}, \boldsymbol{\alpha}|\boldsymbol{A}^{(k)}, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\alpha}^{(k)}) \\
=&\mathbb{E}_{\boldsymbol{s},\boldsymbol{e}|\boldsymbol{m};\boldsymbol{A}^{(k)},\boldsymbol{\gamma}^{(k)},\boldsymbol{\alpha}^{(k)},\beta}\{-\log p(\boldsymbol{m}, \boldsymbol{s}, \boldsymbol{e}|\boldsymbol{A}, \boldsymbol{\gamma}, \boldsymbol{\alpha}, \beta)\} \\
=&\mathbb{E}_{\boldsymbol{s},\boldsymbol{e}|\boldsymbol{m};\boldsymbol{A}^{(k)},\boldsymbol{\gamma}^{(k)},\boldsymbol{\alpha}^{(k)},\beta}\{-\log p(\boldsymbol{m}|\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{A}, \beta) \\
&-\log p(\boldsymbol{s}|\boldsymbol{\gamma}) - \log p(\boldsymbol{e}|\boldsymbol{\alpha})\} \\
=&\frac{1}{2\beta}\langle\|\boldsymbol{m} - \boldsymbol{A}\boldsymbol{s} - \boldsymbol{e}\|_2^2\rangle + \frac{1}{2}\sum_i(\log\boldsymbol{\gamma}_i + \frac{\langle\boldsymbol{s}_i^2\rangle}{\boldsymbol{\gamma}_i}) \\
&+\frac{1}{2}\sum_i(\log\boldsymbol{\alpha}_i + \frac{\langle\boldsymbol{e}_i^2\rangle}{\boldsymbol{\alpha}_i}) + C_1 \\
=&\frac{1}{2\beta}\|\boldsymbol{m} - \boldsymbol{A}\langle\boldsymbol{s}\rangle - \langle\boldsymbol{e}\rangle\|_2^2 + 2\,\text{Tr}(\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{se}|\boldsymbol{m}}) + \text{Tr}(\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}\boldsymbol{A}^T) \\
&+\frac{1}{2}\sum_i(\log\boldsymbol{\gamma}_i + \frac{\langle\boldsymbol{s}_i^2\rangle}{\boldsymbol{\gamma}_i}) + \frac{1}{2}\sum_i(\log\boldsymbol{\alpha}_i + \frac{\langle\boldsymbol{e}_i^2\rangle}{\boldsymbol{\alpha}_i}) + C_2,
\end{aligned}
$$

where $\langle\cdot\rangle$ stands for the posterior expectation.

In M-step, the objective function to minimize is $Q(\Psi|\Psi^{(k)}) - \log p(\boldsymbol{\gamma}) = Q(\Psi|\Psi^{(k)}) + \sum_i((a+1)\log\boldsymbol{\gamma}_i) + const$. The update rules for $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are obtained by taking derivatives:
Update $\boldsymbol{\alpha}$: $\boldsymbol{\alpha}_i = \langle\boldsymbol{e}_i^2\rangle = \boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}^2(i) + \boldsymbol{\Sigma}_{\boldsymbol{e}|\boldsymbol{m}}(i,i), \forall i$.
Update $\boldsymbol{\gamma}$: $\boldsymbol{\gamma}_i = \frac{\langle\boldsymbol{s}_i^2\rangle}{2a+3} = \frac{\boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}^2(i)+\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(i,i)}{2a+3}, \forall i$.

Directly updating the whole matrix $\boldsymbol{A}$ under the constraints $\boldsymbol{A} \in \mathscr{A}$ is difficult. However, we can update each column of $\boldsymbol{A}$ with other columns fixed and still obey the constraints $\boldsymbol{A} \in \mathscr{A}$. The order to update the columns follows the decreasing order of the magnitudes of the elements in $\langle\boldsymbol{s}\rangle$. This is inspired by the Successive Interference Cancellation (SIC) strategy and we omit the explanations here due to space limit. To simplify the presentation, we ***assume*** that $|\langle\boldsymbol{s}_1\rangle|$ is the largest, and therefore we first update $\boldsymbol{A}_1$.
Update $\boldsymbol{A}_1$: Given $\boldsymbol{A}_2^{(k)}, \boldsymbol{A}_3^{(k)}, \cdots, \boldsymbol{A}_d^{(k)}$,

$$
\begin{aligned}
\boldsymbol{A}_1^{(k+1)} = \underset{\substack{\boldsymbol{A}_1=\text{vec}(\boldsymbol{U}_1\boldsymbol{V}_1^T)\\ \|\boldsymbol{U}_1\|_2=1,\|\boldsymbol{V}_1\|_2=1}}{\arg\min} &\{\|\boldsymbol{m}-\langle\boldsymbol{e}\rangle-\sum_{i=2}^{d}\langle\boldsymbol{s}_i\rangle\boldsymbol{A}_i^{(k)}-\langle\boldsymbol{s}_1\rangle\boldsymbol{A}_1\|_2^2 \\
&+2\,\text{Tr}(\boldsymbol{A}_1\boldsymbol{\Sigma}_{\boldsymbol{se}|\boldsymbol{m}}(1,:))+\text{Tr}\left(\boldsymbol{A}_1\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(1,1)\boldsymbol{A}_1^T\right. \\
&\left.+2\sum_{i=2}^{d}\boldsymbol{A}_i^{(k)}\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(1,i)\boldsymbol{A}_1^T\right)\} \\
= \underset{\substack{\boldsymbol{A}_1=\text{vec}(\boldsymbol{U}_1\boldsymbol{V}_1^T)\\ \|\boldsymbol{U}_1\|_2=1,\|\boldsymbol{V}_1\|_2=1}}{\arg\min} &\|\boldsymbol{h} - \boldsymbol{A}_1\|_2^2
\end{aligned}
$$

$$
\boldsymbol{h} \triangleq \frac{\langle\boldsymbol{s}_1\rangle\boldsymbol{m}-\langle\boldsymbol{s}_1\rangle\langle\boldsymbol{e}\rangle-\boldsymbol{\Sigma}_{\boldsymbol{se}|\boldsymbol{m}}^T(1,:)-\sum_{i=2}^{d}[\langle\boldsymbol{s}_1\rangle\langle\boldsymbol{s}_i\rangle+\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(1,i)]\boldsymbol{A}_i^{(k)}}{\langle\boldsymbol{s}_1\rangle^2+\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(1,1)}.
$$

At first glance, this still seems hard to solve. However, utilizing the structure of $\boldsymbol{A}_1$, we can transform this problem to the equivalent matrix form:

$$
(\boldsymbol{U}_1^{(k+1)}, \boldsymbol{V}_1^{(k+1)}) = \underset{\substack{\boldsymbol{U}_1,\boldsymbol{V}_1\\ \|\boldsymbol{U}_1\|_2=1,\|\boldsymbol{V}_1\|_2=1}}{\arg\min} \|\text{Mat}(\boldsymbol{h}) - \boldsymbol{U}_1\boldsymbol{V}_1^T\|_F^2.
$$

Algorithm 1 Sparse Bayesian Learning for Robust PCA

Input: Observation $\boldsymbol{M} \in \mathbb{R}^{n_1\times n_2}$, noise variance $\beta > 0$, Inverse-gamma prior parameter a
Initialize: $k = 0, \quad \boldsymbol{\gamma}_i^{(0)} = 1, \quad \boldsymbol{\alpha}_i^{(0)} = 1, \forall i$
$d = \min(n_1, n_2), \boldsymbol{U}^{(0)} \in \mathbb{R}^{n_1\times d}, \boldsymbol{V}^{(0)} \in \mathbb{R}^{n_2\times d}$
While not converged Do
Step 1. Fix $\boldsymbol{A}^{(k)}$, repeat update $\boldsymbol{\gamma}$, $\boldsymbol{\alpha}$ to certain precision:
Calculate $\boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}, \boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}, \boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}$, and $\text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{e}|\boldsymbol{m}})$ use (8)-(9)
$\boldsymbol{\alpha}_i = \boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}^2(i) + \boldsymbol{\Sigma}_{\boldsymbol{e}|\boldsymbol{m}}(i,i)$;
$\boldsymbol{\gamma}_i = \left(\boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}^2(i) + \boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(i,i)\right)/(2a+3)$.
Step 2. Fix $\boldsymbol{\gamma}^{(k+1)}$ and $\boldsymbol{\alpha}^{(k+1)}$, update $\boldsymbol{A}$:
Calculate $\boldsymbol{\Sigma}_{\boldsymbol{se}|\boldsymbol{m}}, \boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}, \boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}, \boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}$ use (8)-(10)
$\langle\boldsymbol{s}\rangle \triangleq \boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}, \langle\boldsymbol{e}\rangle \triangleq \boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}$;
$\boldsymbol{index}$=sort([$|\langle\boldsymbol{s}_1^{(k+1)}\rangle|, \cdots, |\langle\boldsymbol{s}_d^{(k+1)}\rangle|$], 'descend');
for j=1:d
//update $\boldsymbol{A}_{\boldsymbol{index}(j)}^{(k+1)}$ use $\boldsymbol{A}_{\boldsymbol{index}(i)}^{(k+1)}, i = 1, \cdots, j-1$,
//and $\boldsymbol{A}_{\boldsymbol{index}(i)}^{(k)}, i = j+1, \cdots, d$.
$j' \triangleq \boldsymbol{index}(j)$;
$\boldsymbol{h} = \frac{1}{\langle\boldsymbol{s}_{j'}\rangle^2+\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(j',j')}\{\langle\boldsymbol{s}_{j'}\rangle\boldsymbol{m} - \langle\boldsymbol{s}_{j'}\rangle\langle\boldsymbol{e}\rangle - \boldsymbol{\Sigma}_{\boldsymbol{se}|\boldsymbol{m}}^T(j',:)$
$-\sum_{l\in\{\boldsymbol{index}(i):i<j\}}[\langle\boldsymbol{s}_{j'}\rangle\langle\boldsymbol{s}_l\rangle + \boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(j',l)]\boldsymbol{A}_l^{(k+1)}$
$-\sum_{l\in\{\boldsymbol{index}(i):i>j\}}[\langle\boldsymbol{s}_{j'}\rangle\langle\boldsymbol{s}_l\rangle + \Sigma_{\boldsymbol{s}|\boldsymbol{m}}(j',l)]\boldsymbol{A}_l^{(k)}\}$.

$(\boldsymbol{U}_{j'}^{(k+1)}, \boldsymbol{V}_{j'}^{(k+1)})$ = first singular vector pair of $\text{Mat}(\boldsymbol{h})$
$\boldsymbol{A}_{j'}^{(k+1)} = \text{vec}(\boldsymbol{U}_{j'}^{(k+1)}{\boldsymbol{V}_{j'}^{(k+1)}}^T)$.
end
$k := k+1$.
End While
Output: $\boldsymbol{E} = \text{Mat}(\langle\boldsymbol{e}\rangle), \boldsymbol{L} = \text{Mat}(\hat{\boldsymbol{A}}\langle\boldsymbol{s}\rangle)$

The *optimal* solution is given by the first singular vector pair of $\text{Mat}(\boldsymbol{h})$. Updating $\boldsymbol{U}_1$ and $\boldsymbol{V}_1$ as a pair is inspired by the success of K-SVD [17]. But note that in K-SVD, $\boldsymbol{V}_1$ is not restricted to be unit length.

To update the jth column, the derivation is similar to updating $\boldsymbol{A}_1$, except that we use the latest updates of the other columns. The whole algorithm is summarized in Algorithm 1. One can set d as maximal target rank for large-scale problems. The proof of the following theorem is based on [16, Th. 7].

Theorem 1. *Algorithm 1 guarantees that* $p(\Psi^{(k+1)}|\boldsymbol{m}) \geq p(\Psi^{(k)}|\boldsymbol{m})$ *in each iteration.*

Why it leads to sparse solution? Essentially we are doing Type-II MAP, i.e., maximize $p(\Psi|\boldsymbol{m}) \propto p(\boldsymbol{m}|\Psi)p(\gamma)$, which

is guaranteed by Theorem 1. We can show that

$$\begin{aligned}
&\min_{\boldsymbol{\gamma},\boldsymbol{\alpha},\boldsymbol{A}\in\mathscr{A}} -2\log[p(\boldsymbol{m}|\Psi)p(\boldsymbol{\gamma})] \\
&= \min_{\boldsymbol{\gamma},\boldsymbol{\alpha},\boldsymbol{A}\in\mathscr{A}} \boldsymbol{m}^T\boldsymbol{\Sigma}_{\boldsymbol{m}}^{-1}\boldsymbol{m} + \log|\boldsymbol{\Sigma}_{\boldsymbol{m}}| + 2(a+1)\log|\boldsymbol{\Gamma}| + C \\
&= \min_{\boldsymbol{\gamma},\boldsymbol{\alpha},\boldsymbol{A}\in\mathscr{A}} \{\min_{\boldsymbol{s},\boldsymbol{e}}[\frac{1}{\beta}\|\boldsymbol{m}-\boldsymbol{A}\boldsymbol{s}-\boldsymbol{e}\|_2^2 + \boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s} + \boldsymbol{e}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{e}] \\
&\quad + \log|\boldsymbol{\Sigma}_{\boldsymbol{m}}| + 2(a+1)\log|\boldsymbol{\Gamma}|\} + C \\
&= \min_{\boldsymbol{s},\boldsymbol{e},\boldsymbol{A}\in\mathscr{A}} \{\frac{1}{\beta}\|\boldsymbol{m}-\boldsymbol{A}\boldsymbol{s}-\boldsymbol{e}\|_2^2 + \min_{\boldsymbol{\gamma},\boldsymbol{\alpha}}[\boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s} + \boldsymbol{e}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{e} \\
&\quad + \log|\boldsymbol{\Sigma}_{\boldsymbol{m}}| + 2(a+1)\log|\boldsymbol{\Gamma}|]\} + C
\end{aligned}$$

The first term is the data-fidelity term, while the remaining quantity is the underlying SBL penalty term. Recall that $\boldsymbol{\Sigma}_{\boldsymbol{m}} = \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \boldsymbol{\Lambda} + \beta\boldsymbol{I}$. It is known that log-determinant encourages low-rank [18]. So $\log|\boldsymbol{\Sigma}_{\boldsymbol{m}}|$ and $\log|\boldsymbol{\Gamma}|$ push both $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$ to be sparse. As a result of the variances going to zero, the corresponding entries of $\boldsymbol{s}$ and $\boldsymbol{e}$ will be driven to 0. **Parameter setting and initialization:** Recall that we assume an Inverse-gamma prior on $\boldsymbol{\gamma}$, i.e., $p(\gamma_i) = \text{IG}(a,b)$, with $b \to 0$. The reason to have an extra prior on γ becomes clear if we reformulate the observation model as $\boldsymbol{m} = [\boldsymbol{A} \quad \boldsymbol{I}][\boldsymbol{s};\boldsymbol{e}] + \boldsymbol{n}$. If no prior on $\boldsymbol{\gamma}$ is assumed, the elements of $\boldsymbol{s}$ will be treated equally as the elements of $\boldsymbol{e}$ in the long vector $[\boldsymbol{s};\boldsymbol{e}]$, which is similar to setting $\lambda = 1$ in (2). Note that the dimension of $\boldsymbol{e}$ is much larger than that of $\boldsymbol{s}$. There will be a trivial sparse solution with $\boldsymbol{e} = \boldsymbol{0}$ and dense $\boldsymbol{s}$. Putting a prior on $\boldsymbol{\gamma}$ is analogous to setting the weight parameter λ in (2). Motivated by the objective in the M-step, we set a such that $2a+3 = \max\left(\sqrt{\|\boldsymbol{E}_0\|_0/\text{rank}(\boldsymbol{L}_0)}, 1\right)$. Since usually there is no knowledge of rank and sparsity, we estimate them from the data by thresholding $\boldsymbol{\gamma}^{(k)}$ and $\boldsymbol{\alpha}^{(k)}$ at the end of Step 1.

Since a good initialization can help accelerate the convergence and avoid some local minima, we initialize $\boldsymbol{U}^{(0)} \in \mathbb{R}^{n_1\times\min(n_1,n_2)}$ and $\boldsymbol{V}^{(0)} \in \mathbb{R}^{n_2\times\min(n_1,n_2)}$ as the *full* singular vectors of some pre-estimated low-rank matrix $\hat{\boldsymbol{L}} \in \mathbb{R}^{n_1\times n_2}$. Here we emphasize that the dimension of $\boldsymbol{U}^{(0)}$ is *not* $n_1 \times \text{rank}(\hat{\boldsymbol{L}})$. We set $\beta = (3\sigma)^2$ to accommodate any modeling errors especially at the beginning of the iterations.

5. EXPERIMENTS

We compare with PCP, SPCP [4][19], Iterative Reweighted PCP(IR-PCP) [20][21], AltProj [22], PB_RPCA [14] (correct typos in Eq.22), BRMF [23], SRPCP [10], VB_RPCA [12] (result is poor and not shown), and the oracle Matrix Completion (MC) [24] solution where only outlier-free entries are observed. Our $\boldsymbol{U}^{(0)}$ and $\boldsymbol{V}^{(0)}$ are initialized from full singular vectors of the low-rank matrix estimated by SRPCP [10].

5.1. Comparison on Simulated Data

We first follow the benchmark simulation where the low-rank matrix is generated by $\boldsymbol{A}\boldsymbol{B}^T$, where $\boldsymbol{A} \in \mathbb{R}^{100\times r}$ and $\boldsymbol{B} \in \mathbb{R}^{100\times r}$ are standard Gaussian matrices. The sparse matrix is generated by selecting non-zero entries uniformly at random, and their values are drawn from $U[0,100]$. The elements of the dense noise matrix are i.i.d. and drawn from $\mathcal{N}(0, 0.1^2)$.

The estimated low-rank matrix $\hat{\boldsymbol{L}}$ is compared with the ground truth. Fig. 1 shows the average Relative Error over 10 trials in log scale, i.e., $2\log(average(\|\hat{\boldsymbol{L}} - \boldsymbol{L}_0\|_F/\|\boldsymbol{L}_0\|_F))$. The color bar is at top-right. We can see that the proposed SBL approach improves upon SRPCP, and nearly matches the performance of the oracle matrix completion solution.

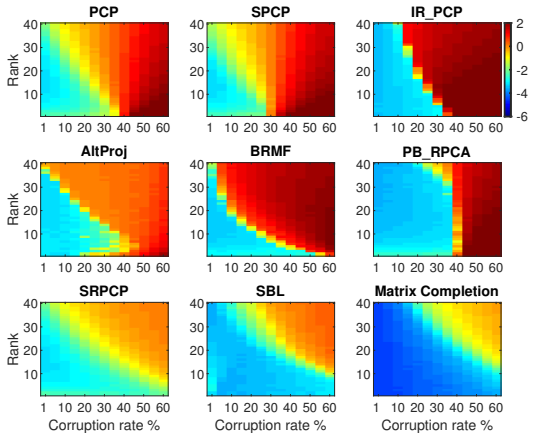

Fig. 1: Average Relative Error in log scale.

5.2. Comparison on Text Extraction

We follow [23] to conduct a text extraction image processing simulation, such that the results are directly visible. The ground truth low-rank image is a rank ten 256×256 matrix. We embed black text in the image, whose values are randomly drawn from $\text{U}[-1,0]$. The text here can be viewed as sparse matrix, whose support (mask) is of interest, which is obtained by thresholding the estimated $\hat{\boldsymbol{E}}$. The threshold is automatically adjusted to find the maximum F-measure [23] for each method. The results are shown in Fig. 2. The proposed approach performs best both visually and in terms of F-measure.

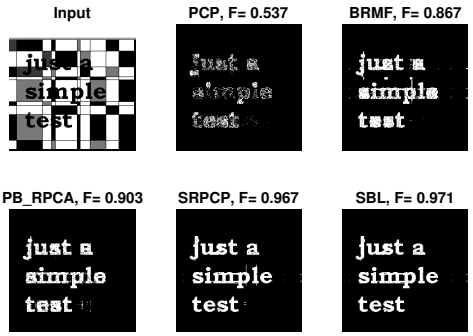

Fig. 2: Recovered text mask by each method.

6. REFERENCES

[1] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, 2011.

[2] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 2080–2088.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011.

[4] Z. Zhou, X. Li, J. Wright, E. J. Candès, and Y. Ma, "Stable principal component pursuit," in *2010 IEEE International Symposium on Information Theory*, June 2010, pp. 1518–1522.

[5] K. Mitra, A. Veeraraghavan, and R. Chellappa, "Analysis of sparse regularization based robust regression approaches," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1249–1257, 2012.

[6] Y. Jin and B. D. Rao, "Algorithms for robust linear regression by exploiting the connection to sparse signal recovery," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3830–3833.

[7] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.

[8] D. P. Wipf and B. D. Rao, "ℓ_0-norm minimization for basis selection," in *Advances in Neural Information Processing Systems 17*, 2005, pp. 1513–1520.

[9] D. P. Wipf and B. D. Rao, "Sparse bayesian learning for basis selection," *IEEE Transactions on Signal processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[10] J. Liu and B. D. Rao, "Robust pca via ℓ_0-ℓ_1 regularization," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 535–549, Jan 2019.

[11] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3419–3430, 2011.

[12] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse bayesian methods for low-rank matrix estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3964–3977, 2012.

[13] D. Wipf, "Non-convex rank minimization via an empirical bayesian approach," in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'12. Arlington, Virginia, United States: AUAI Press, 2012, pp. 914–923.

[14] T.-H. Oh, Y. Matsushita, I. Kweon, and D. Wipf, "A pseudo-bayesian algorithm for robust pca," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 1390–1398.

[15] J. Liu and B. D. Rao, "Sparse bayesian learning for robust pca: Algorithms and analyses," In preparation.

[16] Y. Chen and M. R. Gupta, "Em demystified: An expectation-maximization tutorial," Department of Electrical Engineering, University of Washington, Tech. Rep., February 2010.

[17] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[18] M. Fazel, H. Hindi, and S. P. Boyd, "Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 3, June 2003, pp. 2156–2162 vol.3.

[19] N. S. Aybat and G. Iyengar, "An alternating direction method with increasing penalty for stable principal component pursuit," *Computational Optimization and Applications*, vol. 61, no. 3, pp. 635–668, 2015.

[20] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *Int. J. Comput. Vision*, vol. 121, no. 2, pp. 183–208, Jan 2017.

[21] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, "Compressive sensing via nonlocal low-rank regularization," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3618–3632, Aug 2014.

[22] P. Netrapalli, N. U N, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust pca," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 1107–1115.

[23] N. Wang and D. Y. Yeung, "Bayesian robust matrix factorization for image and video processing," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 1785–1792.

[24] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, June 2010.