UNIVERSITY OF CALIFORNIA SAN DIEGO

Robust PCA and Robust Linear Regression via Sparsity Regularization

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Signal and Image Processing)

by

Jing Liu

Committee in charge:

Professor Bhaskar D. Rao, Chair
Professor Ery Arias-Castro
Professor Pamela C. Cosman
Professor Piya Pal
Professor Rayan Saab

2019

The dissertation of Jing Liu is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California San Diego

2019

DEDICATION

To my family and teachers.

EPIGRAPH

"*If among these errors are some which appear too large to be admissible, then those equations which produced these errors will be rejected, as coming from too faulty experiments, and the unknowns will be determined by means of the other equations, which will then give much smaller errors.*"

—A.M. Legendre, *On the Method of Least Squares*. 1805

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SUPPLEMENTAL FILES

Supplemental material for Chapter 2.

Supplemental material for Chapter 3.

ACKNOWLEDGEMENTS

First and foremost, I am greatly indebted to my advisors, Prof. Bhaskar Rao and Prof. Pamela Cosman, for their excellent guidance and support. I also appreciate Prof. Piya Pal, from whom I learned a lot. I also thank my committee members Prof. Rayan Saab and Prof. Ery Arias-Castro as well as my teachers and collaborators. I would also like to thank Prof. Wotao Yin, Prof. Deanna Needell, Prof. Anant Sahai, Prof. Emmanuel Candès, and Prof. Stanley Osher. I also appreciate the help and support from my friends and family.

Chapter 2, in part, is a reprint of the material as it appears in the paper: J. Liu, P. C. Cosman and B. D. Rao, "Robust Linear Regression via ℓ_0 Regularization," in *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 698-713, Feb. 2018. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part, is a reprint of the material as it appears in the paper: J. Liu and B. D. Rao, "Robust PCA via ℓ_0-ℓ_1 Regularization," in *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 535-549, Jan. 2019. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in part, is a reprint of the material as it appears in the papers: J. Liu and B. D. Rao, "Sparse Bayesian Learning for Robust PCA: Algorithms and Analyses," Submitted, and J. Liu, Y. Ding and B. Rao, "Sparse Bayesian Learning for Robust PCA," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May. 2019, pp. 4883-4887. The dissertation author was the primary investigator and author of these papers.

VITA

2010	Bachelor of Engineering, Beijing Institute of Technology, China
2013	Master of Science, Tsinghua University, China
2019	Doctor of Philosophy, University of California San Diego

PUBLICATIONS

J. Liu, P. C. Cosman and B. D. Rao, "Robust Linear Regression via ℓ_0 Regularization," in *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 698-713, Feb. 2018.

J. Liu and B. D. Rao, "Robust PCA via ℓ_0-ℓ_1 Regularization," in *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 535-549, Jan. 2019.

J. Liu and B. D. Rao, "Sparse Bayesian Learning for Robust PCA: Algorithms and Analyses," submitted to *IEEE Transactions on Signal Processing*.

J. Liu, Y. Ding and B. D. Rao, "Sparse Bayesian Learning for Robust PCA," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May. 2019, pp. 4883-4887.

J. Liu, P. C. Cosman and B. D. Rao, "Sparsity Regularized Principal Component Pursuit," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2017, pp. 4431-4435.

J. Liu, M. Chuang, A. Chisholm and P. Cosman, "Image Registration Robust to Sparse Large Errors," in *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, 2015, pp. 1975-1980.

G. Zhang, J. Chen, G. Su, and J. Liu, "Double-pupil Localization of Face Images," *Pattern Recognition*, vol.46, Issue 3, pp. 642-648, Mar. 2013.

C. Yang, J. Chen, C. Xia, J. Liu, G. Su, "A SFM-Based Sparse to Dense 3D Face Reconstruction Method Robust to Feature Tracking Errors," *2013 IEEE International Conference on Image Processing*, Melbourne, VIC, 2013, pp. 3617-3621.

Cong Xia, Jiansheng Chen, Chang Yang, Jing Wang, Jing Liu, Guangda Su, and Gang Zhang, "Choosing Multi-illumination training Images based on the degree of linear independency," *International Conference on Machine Vision Applications*, May 2013, pp. 403-406.

Cong Xia, Jiansheng Chen, Chang Yang, Jing Wang, Jing Liu, Guangda Su, and Gang Zhang, "Scalable Illumination Robust Face Identification Using Harmonic Representation," *International Conference on Digital Image Processing*, Apr. 2013.

J. Liu, G. Su, X. Ren, and J. Chen, "Human Face Super-Resolution Based on NSCT," *11th Asian Conference on Computer Vision*, Daejeon, Korea, 2012, pp. 680-693.

Shoubin Xiang, Guangda Su, Jiansheng Chen, Jing Liu, Xiaohui Tan, "Brick Stack Anomaly Detection and Recognition Based on Machine Vision," *ACTA OPTICA SINICA*, Vol.07, pp. 191-197, 2011.

ABSTRACT OF THE DISSERTATION

Robust PCA and Robust Linear Regression via Sparsity Regularization

by

Jing Liu

Doctor of Philosophy in Electrical Engineering
(Signal and Image Processing)

University of California San Diego, 2019

Professor Bhaskar D. Rao, Chair

Robustness to outliers is of paramount importance in data analytics. However, many data analysis tools are not robust to outliers due to their criterion of minimizing the sum of *squared* errors. One essential characteristic of the outliers is that they are sparse. A significant contribution of this thesis is the development of a novel framework that directly uses genuine ℓ_0-'norm' to enforce the sparseness of the outliers, while uses ℓ_1-norm to address the inlier noise, and development of algorithms with better recovery guarantees than the state-of-the-art ℓ_1 relaxation approach.

We first study this framework in the Robust Linear Regression setting and propose an

Algorithm for Robust Outlier Support Identification (AROSI) to minimize a novel objective function. The proposed algorithm is guaranteed to converge in a finite number of iterations to a local optimum. Under certain conditions, AROSI is guaranteed to have exact recovery when only sparse outliers are present. Furthermore, the estimation error is bounded when there is dense inlier noise as well. It can also identify the outliers without any false alarm.

Then, we study this framework in the Robust Principal Component Analysis (PCA) setting and propose a novel objective that additionally uses nuclear norm to capture the low-rank matrix. The associated algorithm, termed Sparsity Regularized Principal Component Pursuit (SRPCP), is shown to converge in a finite number of iterations to a local optimum. Under certain conditions, SRPCP is guaranteed to have exact recovery in the presence of sparse outliers only, and bounded error in the noisy case. It can also identify the outliers without any false alarm. An important byproduct of our analysis is the result that, the widely used Principal Component Pursuit (PCP) method and its missing entry version are actually stable to dense inlier noise. We further propose an Iterative Reweighted SRPCP method that uses log-determinant to capture the low-rank matrix instead, which also converges and achieves even better performance.

To better enforce the low-rankness, we transform the Robust PCA objective into a novel Robust Sparse Linear Regression objective with equivalent global optima guarantee. Then we propose a concise Sparse Bayesian Learning method to solve this new objective, and the method is shown to encourage the solution to be low-rank and the outliers to be sparse. To further utilize the sparsity pattern information of the outliers in the Robust PCA problem, a modification of the above Bayesian method is proposed and analyzed. Empirical studies demonstrate the superiority of the proposed methods over existing state-of-the-art methods.

Chapter 1

Introduction

1.1 Motivation and Context

In today's big data era, outliers are becoming more and more common in various applications and datasets. The outliers may be caused by the less-controlled large-scale data collection process, e.g., user rating and crowd-sourcing, or may be caused by the failure of some cheap sensors in the large-scale sensor network, or due to some malicious tampering of the system, etc. Robustness to outliers is of paramount importance when extracting information or learning from big data. Unfortunately, many classical data analysis tools are *not* robust to the outliers, e.g., Least Squares regression, Principal Component Analysis (PCA), and Tensor Decomposition, to name a few. This is due to their criterion of minimizing the sum of *squared* errors, which is very sensitive to large fitting residuals. Even a few outliers can significantly degrade their performance. To illustrate this, we quote an example from a recent survey [1] shown in Fig. 1.1.

There are different definitions of outliers. Here we focus on the case where some measurements are corrupted, i.e., with large observation errors. Other type of the outliers could be a sub-sequence generated by a mechanism (distribution) different from that of the

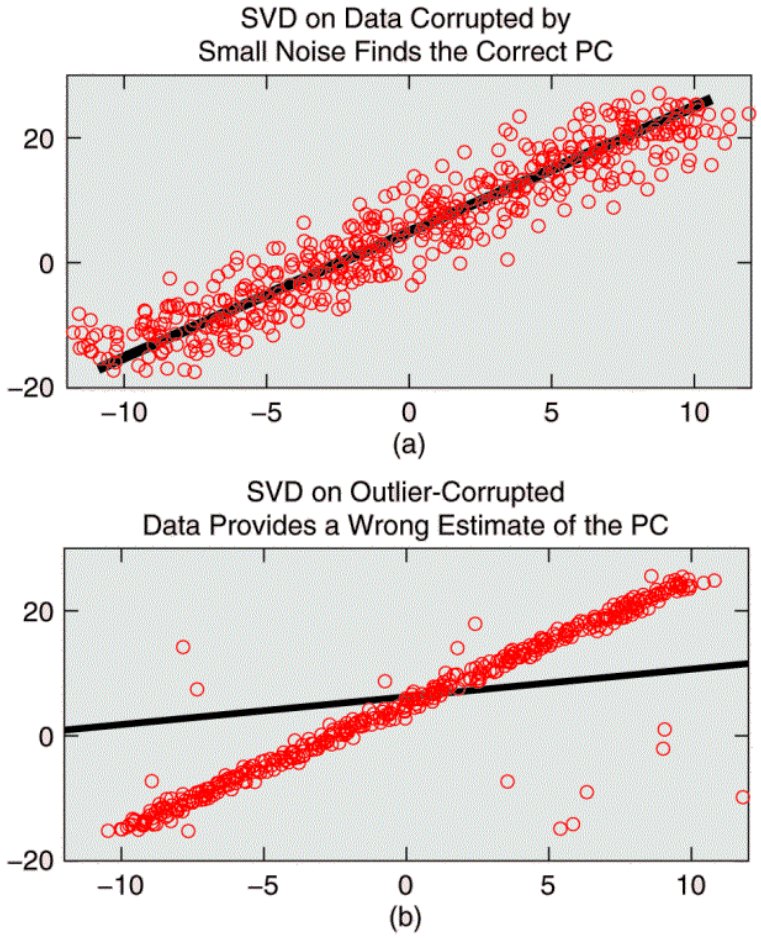

Figure 1.1. (a) PCA in small noise: the SVD solution works. The black line is the estimated principal component computed using the observed data. (b) PCA in outliers: the SVD solution fails to correctly find the direction of largest variance of the true data.

normal data, we refer the interested reader to [2] regarding this type of outliers. Robustness to outlier corruptions is very challenging as we usually have no information about the locations and values for them. Sometimes we are even not aware of the outliers in the system. In this thesis, we focus our discussion on the fundamental linear regression and PCA problems, but the framework developed is very general and can be applied to many data analysis and machine learning problems, e.g., Tensor Decomposition, Matrix Sensing, and Deep Autoencoders.

1.1.1 Robust Linear Regression

In a linear regression setting, the goal is to estimate the linear relationship between two variables: $a \in R^n$ (explanatory variable) and $y \in R$ (response variable), from m pairs of training samples $\{(y_i, a_i), i = 1, \ldots, m\}$, where $m > n$. The following model is commonly assumed:

$$y_i = a_i^T x + \mu_i, \ i = 1, \ldots, m \tag{1.1}$$

or in matrix form: $y = Ax + \mu$, where measurements $y = (y_1, \ldots, y_m)^T$, and matrix $A = [a_1, \ldots, a_m]^T$ are known. $x \in R^n$ is the model parameter to be estimated, and $\mu = (\mu_1, \ldots, \mu_m)^T$ is the observation error. It is also commonly assumed that A has full column rank. In many linear regression data sets, there are some observations y_i known as *outliers* that have been corrupted by large observation errors [3]. Such outliers often lead to the failure of Ordinary Least Square (OLS) estimation [4]. The goal of robust linear regression is to accurately estimate the model parameter in the presence of these troublesome outliers. Many robust estimators [5]–[7] have been developed in the spirit of *Robust Statistics*. Recently, this problem has received considerable interest from the signal processing community due to its underlying connections with the rapidly developing Sparse Signal Recovery (SSR) framework, which aims to recover a sparse solution from an under-determined system of linear equations. The SSR formulation often splits the observation error μ into two terms: $\mu = \eta + e$, where $\eta \in R^m$ is small magnitude bounded inlier noise, and $e \in R^m$ represents the large error component that captures outliers. So model (1.1) becomes:

$$y = Ax + \eta + e. \tag{1.2}$$

In the literature, the following two assumptions are often made about the outliers:

1. Outlier entries often have significantly larger observation errors than inlier entries

have, and $\min\{|e_i| : e_i \neq 0\} > \|\eta\|_\infty$.

2. The fraction of outliers in the whole dataset is usually small, so the outlier corruptions vector e is *sparse*, i.e., most entries in e are zero.

In Robust Statistics, many robust regression estimators aim to limit the influence of large error entries under the first assumption. The most popular family of these methods is the M-estimators [7]. For the second assumption, it is often utilized under the principle of fitting the majority of the data. Least Median of Squares (LMedS) [8], Least Trimmed Squares (LTS) [5], [6], and Random Sample Consensus (RANSAC) [9] are representative methods. However, due to the combinatorial nature, these algorithms are impractical for solving high dimensional problems.

In contrast to the robust statistics approach, most SSR methods merely use the first assumption in the final reprojection step via thresholding, e.g., [10]. One exception is [11][12], which developed a general thresholding function based iterative procedure and [11] was shown to be equivalent to a special class of M-estimators. For the second assumption, the SSR methods explicitly model the sparsity of outliers, and they deal with the outliers in two major ways: Projection Approach [13] and Joint Approach [14]. Let V denote the subspace spanned by the columns of A, and let $F \in R^{(m-n)\times m}$ be a matrix whose rows form an orthobasis of $V^\perp$. Then we have $FA = 0$. The Projection Approach applies F to the measurements and from (1.2) we obtain

$$b \triangleq Fy = FAx + Fe + F\eta = Fe + F\eta. \tag{1.3}$$

The original problem is transferred to the recovery of a sparse vector e, given the underdetermined measurement matrix F and noisy measurements b. Various SSR methods can be directly applied to solve this problem, such as BSRR [15], [16] which is based on Sparse Bayesian Learning (SBL) [17], [18], and Second-Order Cone Programming (SOCP) [10]

which is based on ℓ_1 minimization [19]–[21]. Note that the ℓ_1 estimator ($\arg\min_x \|y - Ax\|_1$) is equivalent to the SOCP case of no dense inlier noise [13]. The Joint Approach reformulates the original model into $y = [A\ I_{m\times m}]\begin{bmatrix} x \\ e \end{bmatrix} + \eta$, where $[A\ I_{m\times m}]$ is under-determined and the lower part of $\begin{bmatrix} x \\ e \end{bmatrix}$ is sparse. Many existing SSR methods can be extended to deal with this formulation via restricting the lower part of $\begin{bmatrix} x \\ e \end{bmatrix}$ to be sparse, e.g., BPRR which is based on ℓ_1 minimization [16], ℓ_p ($0< p \leq 1$) regularization which assumes a super-Gaussian prior for e to encourage its sparseness [14], [22], Giannakis's algorithm for robust sensing [23] that utilizes a log-sum penalty function [24]–[27], Jin's empirical Bayesian inference-based algorithm which is extended from SBL [22], and GARD [3] which is based on Orthogonal Matching Pursuit (OMP) [28][29].

The existing methods often tackle the ℓ_0-'norm' of e implicitly (e.g., via OMP or SBL), or through the use of surrogate measures for the ℓ_0-'norm', such as ℓ_1-norm, ℓ_p-norm, and the log-sum function. Besides these methods, the hard thresholding based iterative method [11] shows its equivalence with a family (infinitely many) of nonconvex penalties for e (one special case is the ℓ_0-'norm') to encourage its sparseness, but without any recovery guarantee and relies on a preliminary robust fit. A natural question is whether it is possible to directly use and deal with the ℓ_0-'norm' of e, and obtain even better recovery guarantees than the state-of-the-art ℓ_1 approach? In this thesis, we provide an affirmative answer to this question.

1.1.2 Robust Principal Component Analysis

Principal component analysis (PCA) is arguably one of the most widely used data analysis methods with numerous applications. However, its performance can significantly degrade if the data is corrupted by even a few outliers. As mentioned in a recent review [1], outliers are becoming even more common in today's big data era. To robustify the PCA, the idea of limiting the influence of large errors has been used to robustly estimate the covariance matrix (e.g., [7], [30]), or replace the square penalty on PCA's reconstruction residual by M-

estimator's penalty (e.g., [31]–[36]). Similar idea of Least Trimmed Squares is also proposed to robustify PCA (e.g., [37], [38]), and suffers from its combinatorial nature.

Recently, many SSR based methods are developed under the name "Robust PCA" [39], with the goal to recover the low-rank matrix $\boldsymbol{L}_0$ and sparse matrix $\boldsymbol{E}_0$ (which often models the outlier corruptions) from their composition $\boldsymbol{M}$ (possibly with additional dense noise). This problem has received a lot of interest in the past decade, with applications ranging from video analysis, face recognition, to recommendation systems.

1.1.2.1 Regularization Approach

Robust PCA was first studied in the noiseless case [39]–[41], the underlying optimization problem is [41]:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \operatorname{rank}(\boldsymbol{L}) + \lambda\|\boldsymbol{E}\|_0 \quad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{E}, \tag{1.4}$$

which is known to be NP-hard. To make the problem computationally viable, [39]–[41] suggest relaxing the rank minimization to nuclear norm minimization and the ℓ_0-'norm' penalty to an ℓ_1-norm penalty, i.e.,

$$\min_{\boldsymbol{L},\boldsymbol{E}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{E}\|_1 \quad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{E}, \tag{1.5}$$

leading to a convex optimization based approach known as Principal Component Pursuit (PCP). Interestingly, one can recover both $\boldsymbol{L}_0$ and $\boldsymbol{E}_0$ exactly under certain conditions by solving this convex program. Since then, many variants have been proposed with the goal being either lower complexity or better performance. For a comprehensive review, we refer the interested readers to [42]. We first discuss the regularization based methods and focus on dealing with the outliers as well as dense inlier noise.

As surrogates for the original ℓ_0-'norm', the ℓ_p-norm and log-sum function on the sparse outlier term $\boldsymbol{E}$ are adopted in [43].

In real world applications, besides the sparse 'corruptions' $\boldsymbol{E}_0$, there is often small magnitude dense inlier noise $\boldsymbol{N}$. The resulting model is:

$$\boldsymbol{M} = \boldsymbol{L}_0 + \boldsymbol{E}_0 + \boldsymbol{N}. \tag{1.6}$$

To address inlier noise, Zhou et al. [44] solved the following relaxed version of (1.5), known as Stable Principal Component Pursuit (SPCP):

$$\min_{\boldsymbol{L},\boldsymbol{E}} ||\boldsymbol{L}||_* + \lambda||\boldsymbol{E}||_1 \quad s.t. \quad ||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}||_F \leq \delta. \tag{1.7}$$

It was shown that the estimation error can be bounded under certain conditions.

Hsu et al. [45] analyzed the Lagrange form of (1.7):

$$\min_{\boldsymbol{L},\boldsymbol{E}} ||\boldsymbol{L}||_* + \lambda||\boldsymbol{E}||_1 + \frac{1}{2\mu}||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}||_F^2. \tag{1.8}$$

In light of the M-estimators, He et al. [36] proposed replacing $||\boldsymbol{E}||_1$ by implicit regularizers of robust M-estimators, i.e., $\varphi(\boldsymbol{E})$, and then solving the following optimization problem:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \mu||\boldsymbol{L}||_* + \varphi(\boldsymbol{E}) + \frac{1}{2}||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}||_F^2. \tag{1.9}$$

Similarly, Chartrand [46] proposed to replace the ℓ_1-norm by implicit regularizers (also called proximal p-norm [46]) of the p-Huber function.

To better approximate the ℓ_0-'norm', rather than using the ℓ_1-norm, Sun et al. [47] used the capped ℓ_1-norm on both the sparse term $\boldsymbol{E}$ and the singular values of $\boldsymbol{L}$:

$$\begin{aligned} &\min_{\boldsymbol{L},\boldsymbol{E}} \frac{1}{\theta_1}\sum_i \min\{\sigma_i(\boldsymbol{L}), \theta_1\} + \frac{1}{\theta_2}\sum_{i,j} \min\{|\boldsymbol{E}_{i,j}|, \theta_2\} \\ &s.t. \quad ||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}||_F \leq \delta. \end{aligned} \tag{1.10}$$

In [48] and [49], the following greedy approach was proposed that directly constrains the ℓ_0-'norm':

$$\min_{\boldsymbol{L},\boldsymbol{E}} ||\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}||_F^2 \quad s.t. \quad \text{rank}(\boldsymbol{L}) \leq r, ||\boldsymbol{E}||_0 \leq k. \tag{1.11}$$

Also, Ulfarsson et al. [50] proposed to use an ℓ_0 penalty to enforce both sparsity and low rank:

$$\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{E}} ||\boldsymbol{M}-\boldsymbol{A}\boldsymbol{B}^T-\boldsymbol{E}||_F^2 + h^2||\boldsymbol{E}||_0 \quad s.t. \quad \boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}_r. \tag{1.12}$$

Note that these methods need to specify the rank (and sparsity), which are usually unknown in practice and hard to specify.

In the context of detecting contiguous outliers in the low-rank representation (termed DECOLOR), Zhou et al. [38] proposed an objective function whose degenerate form can be shown equivalent to the following:

$$||\boldsymbol{L}||_* + \beta||\boldsymbol{E}||_0 + \lambda||\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}||_F^2. \tag{1.13}$$

However, these ℓ_0 approaches do not have any recovery guarantee. Inspired by our robust linear regression method, we extend our framework to the matrix case and propose to solve the following:

$$||\boldsymbol{L}||_* + \beta||\boldsymbol{E}||_0 + \lambda||\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}||_1. \tag{1.14}$$

Compared with (1.8)-(1.10), we use genuine ℓ_0-'norm' to enforce the sparseness of outliers, and employ the ℓ_1-norm instead of the usual Frobenius norm on the noise term. Compared with (1.13), the only difference is the replacement of the Frobenius norm by the ℓ_1-norm on the noise term. But this replacement makes a big difference in that it not only significantly improves the recovery performances, but also enjoys many theoretical guarantees.

Inspired by the superior performance of log-determinant [51], [52] in pursuing the low-rank structure, we further replace the nuclear norm in (1.14) by the log-determinant,

propose and analyze an algorithm to minimize the corresponding objective function.

1.1.2.2 Bayesian Approach

In Robust PCA problem, ideally we want to minimize the rank of the low-rank matrix, but the corresponding objective function is hard to optimize. Though our proposed SRPCP method manages to use the genuine ℓ_0-'norm' to enforce the sparseness of the outliers, it still has to relax the rank function to the nuclear norm or log-determinant on the low-rank matrix. Recall that rank is the sparsity of the singular values, while nuclear norm and log-determinant are equivalent to the ℓ_1-norm and log-sum function of the singular values. Motivated by the superior performance of the Sparse Bayesian Learning (SBL) method [18], [53], [54] in the ℓ_0 minimization problem, we explore the Bayesian approach for Robust PCA.

There have already been several Sparse Bayesian Learning methods proposed for solving the Robust PCA problem. The earliest work [55] proposed to model the low-rank matrix as $\boldsymbol{L} = \boldsymbol{D}(\text{diag}(\boldsymbol{z})\text{diag}(\boldsymbol{s}))\boldsymbol{W}$, and the sparse matrix as $\boldsymbol{E} = \boldsymbol{B} \circ \boldsymbol{X}$, i.e., $\boldsymbol{M} = \boldsymbol{D}(\text{diag}(\boldsymbol{z})\text{diag}(\boldsymbol{s}))\boldsymbol{W} + \boldsymbol{B} \circ \boldsymbol{X} + \boldsymbol{N}$, 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. [56] 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. Some improvement has been shown compared to the previous Bayesian approach [55]. However, it is still inferior to the convex PCP approach. Note that this probabilistic model of the low-rank matrix is also adopted in some later works [57]–[59].

Recently, Wipf [60] proposed a even simpler model that directly assumes the columns of $\boldsymbol{L}$ are independent zero-mean Gaussian vectors which share the same covariance matrix, while the sparse matrix is modeled similar to Babacan's work [56]. Slight improvement over the convex PCP method has been empirically demonstrated. In [61], Jansson et al. assume vec($\boldsymbol{L}$) is zero-mean Gaussian and its covariance matrix is the Kronecker product of two Wishart distributed matrices. It also demonstrated a slight performance improvement over the PCP method, but the complexity of the inference is very high. Wipf et al. [62] further proposed a modification to the model in [60], which assumes vec($\boldsymbol{L}$) is zero-mean Gaussian with covariance matrix obeying a Kronecker-sum structure. 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 has to be used. In this thesis, we develop a concise SBL approach that has minimum assumptions and effectively deals with the requirements of the problem, and also allows exact inference with lower complexity.

1.2 Thesis Outline and Contributions

This thesis is organized as follows.

1.2.1 Provable Robust Linear Regression via ℓ_0 Regularization

In Chapter 2, we address the robust linear regression problem in the presence of outliers, which is challenging as the support of outliers is not known beforehand and the magnitudes of the outliers can be arbitrarily large. We propose an Algorithm for Robust Outlier Support Identification (AROSI) utilizing a novel objective function that uses ℓ_0-'norm' to directly enforce the sparseness of the outliers and uses ℓ_1-norm to address the inlier noise term. The optimization procedure naturally utilizes the large observation error assumption of outliers and directly operates on the ℓ_0-'norm'.

The proposed AROSI algorithm is guaranteed to converge in a finite number of iterations to a fix point, which is a local optimum. Under certain conditions, we have the following guarantees for AROSI:

1) Exact recovery of the signal under any parameter setting in the presence of outliers only, i.e., absence of dense inlier noise (Theorem 2.3).

2) The recovery error is bounded in the noisy case, and the bound is smaller than that of the ℓ_1 estimator (Theorem 2.6).

3) Exact support recovery of outliers when there is no dense inlier noise (Theorem 2.3) as well as the noisy case (Theorem 2.6.d).

4) The ability to keep all the inliers and remove significant outliers in every iteration (Theorems 2.3, 2.6-2.7, and Remark 2.2).

5) Even if the number of outliers is greater than the regression breakdown point of the ℓ_1 estimator, AROSI can still give exact recovery (no dense inlier noise case, Remarks 2.1 and 2.2) or bounded estimation error (noisy case, Remark 2.5 and Theorem 2.7). It can tolerate 50% more outliers than the ℓ_1 estimator under certain conditions.

6) AROSI has desirable recovery performance when the rows of design matrix are i.i.d. from the uniform distribution on the unit sphere.

Extensive empirical comparisons with state-of-the-art methods demonstrate the advantage of the proposed method.

1.2.2 Robust PCA via ℓ_0-ℓ_1 Regularization

In Chapter 3, we address the robust PCA problem in the presence of outliers, which is also challenging as there is no information of the underlying rank of the low-rank matrix and the sparsity/location of the outliers. We extend our AROSI framework and propose a method termed Sparsity Regularized Principal Component Pursuit (SRPCP) to solve this problem. The proposed method utilizes a novel objective function with nuclear norm to capture the low-rank term, ℓ_0-'norm' to address the sparse outlier term, and an ℓ_1-norm to deal with the additive noise term. The optimization procedure naturally utilizes the large observation error assumption of the outliers and directly operates on the ℓ_0-'norm'.

The proposed SRPCP algorithm is guaranteed to converge in a finite number of iterations to a fix point, which is a local optimum. Under certain conditions, we have the following guarantees for SRPCP:

1) Exact recovery of the underlying low-rank matrix in the presence of outliers only, i.e., absence of dense inlier noise (Theorem 3.2).

2) The recovery error is bounded in the noisy case, and the bound is smaller than that of the convex PCP method (Theorem 3.4).

3) The ability to keep all the inliers and remove significant outliers in every iteration (Theorem 3.4).

An important byproduct of our analysis is the result that the widely used Principal Component Pursuit (PCP) method and its missing entry version are both stable to dense inlier noise. Note that they were both designed for the noiseless case.

Inspired by the superior performance of log-determinant [51], [52] in pursuing the low-rank structure, we propose another objective function which replaces the above nuclear norm by the log-determinant. The proposed algorithm, termed Iterative Reweighted Sparsity Regularized Principal Component Pursuit (IR-SRPCP), is also shown to converge. In each iteration, it solves a weighted nuclear norm regularized robust matrix completion problem. We propose an ADMM algorithm to solve this nonconvex subproblem, which also converges.

Simulation studies and two image processing applications are provided and they demonstrate the efficacy of the proposed ℓ_0-ℓ_1 regularization framework to deal with the outliers as well as its superiority over the existing regularization methods.

1.2.3 Bayesian Robust PCA

To further push the performance of Robust PCA, in Chapter 4, we first derive and analyze a new Robust Sparse Linear Regression objective, and prove that it is equivalent to the fundamental minimizing "rank+sparsity" objective of the Robust PCA problem. This equivalence guarantee builds the connection between Robust PCA and Robust Sparse Linear Regression. It offers a new viewpoint for the Robust PCA problem. Many existing methods and analyses in Robust Sparse Linear Regression (e.g., [23-26]) can be leveraged to solve and understand the Robust PCA problem, and vice versa. To solve the proposed objective function, a new Bayesian model and corresponding concise Sparse Bayesian Learning (SBL) approach are proposed, which has minimum assumptions and effectively deals with the requirements of the problem and allows exact inference. Convergence guarantee of the proposed algorithm is provided. The underlying cost function of the proposed Bayesian model is analyzed, and shown to encourage the solution to be low-rank and the outliers to be sparse.

To further utilize the sparsity pattern information of the outliers in Robust PCA problem (i.e., in Sparse and Low-rank decomposition problem, the outliers are usually assumed to be spread out, i.e., sparse in each column and each row), a modification of the above Bayesian method is proposed, which leads to further performance improvement. Empirical studies demonstrate the superiority of the proposed methods over the existing state-of-the-art methods.

Finally, in Chapter 5, we discuss some other linear and non-linear problems where the proposed ℓ_0 framework can be straightforwardly applied to solve. For example, Robust Tensor Decomposition, Robust Matrix Sensing, and Robust Deep Autoencoders.

Chapter 2

Robust Linear Regression via ℓ_0 Regularization

Linear regression in the presence of outliers is challenging as the support/magnitude of the outliers are not known beforehand. Many robust estimators solve this problem via explicitly or implicitly assuming that outliers are sparse and result in large observation errors. In this chapter, we propose an Algorithm for Robust Outlier Support Identification (AROSI) utilizing a novel objective function that uses ℓ_0-'norm' to directly enforce the sparseness of the outliers and uses ℓ_1-norm to address the inlier noise. The advantage over the ℓ_1 relaxation approach will be shown on both theoretical side and performance side.

2.1 Introduction

In a linear regression setting, the goal is to estimate the linear relationship between two variables: $a \in R^n$ (explanatory variable) and $y \in R$ (response variable), from m pairs of training samples $\{(y_i, a_i), i = 1, \ldots, m\}$, where $m > n$. The following model is commonly assumed:

$$y_i = a_i^T x + \mu_i,\ i = 1, \ldots, m \tag{2.1}$$

or in matrix form: $y = Ax + \mu$, where measurements $y = (y_1, \ldots, y_m)^T$, and matrix $A = [a_1, \ldots, a_m]^T$ are known. $x \in R^n$ is the model parameter to be estimated, and $\mu = (\mu_1, \ldots, \mu_m)^T$ is the observation error. It is also commonly assumed that A has full column rank. In many linear regression data sets, there are some observations y_i known as *outliers* that have been corrupted by large observation errors [3]. Such outliers often lead to the failure of Ordinary Least Square (OLS) estimation [4]. The goal of robust linear regression is to accurately estimate the model parameter in the presence of these troublesome outliers. Many robust estimators [5]–[7] have been developed in the spirit of *Robust Statistics*. Recently, this problem has received considerable interest from the signal processing community due to its underlying connections with the rapidly developing Sparse Signal Recovery (SSR) framework, which aims to recover a sparse solution from an under-determined system of linear equations. The SSR formulation often splits the observation error μ into two terms: $\mu = \eta + e$, where $\eta \in R^m$ is small magnitude bounded inlier noise, and $e \in R^m$ represents the large error component that captures outliers. So model (2.1) becomes:

$$y = Ax + \eta + e. \tag{2.2}$$

Additional prior information or assumptions are needed in order to solve the problem. We make the following two reasonable and common assumptions about outliers:

1. Outlier entries often have significantly larger observation errors than inlier entries have, and $\min\{|e_i| : e_i \neq 0\} > \|\eta\|_\infty$.

2. The fraction of outliers in the whole dataset is usually small, so the outlier corruptions vector e is *sparse*, i.e., most entries in e are zero.

In Robust Statistics, many robust regression estimators aim to limit the influence of

large error entries under the first assumption. The most popular family of these methods is the M-estimators [7]. For the second assumption, it is often utilized under the principle of fitting the majority of the data. Least Median of Squares (LMedS) [8], Least Trimmed Squares (LTS) [5], [6], and Random Sample Consensus (RANSAC) [9] are representative methods. LMedS was introduced by Rousseeuw [8]; it minimizes the median of squared residuals instead of the mean (or equivalently, sum). To improve estimation efficiency, Rousseeuw further introduced LTS [5], [6], which aims to minimize $\sum_{i=1}^{h} r_{(i)}^2$, where $r_{(1)}^2 \leq r_{(2)}^2 \cdots \leq r_{(m)}^2$ are the ordered squared residuals, and the value of h is set between $\frac{m}{2}$ and m. RANSAC [9] uses random sampling to calculate possible model parameters and pick the best among them which can fit most of the data. However, due to the combinatorial nature, all of these algorithms are impractical for solving high dimensional problems.

In contrast to the robust statistics approach, most SSR methods merely use the first assumption in the final reprojection step via thresholding, e.g., [10]. One exception is [11][12], which developed a general thresholding function based iterative procedure and [11] was shown to be equivalent to a special class of M-estimators. For the second assumption, the SSR methods explicitly model the sparsity of outliers. Recently many works [63]–[66] address the outliers in the SSR framework, where x is also sparse (in the typically overcomplete dictionary A), and the corruptions may also admit a sparse representation in another general dictionary [67], [68]. Here we focus on the traditional linear regression problem, where x is general, A is over-determined and we have no freedom to design A. Under this setting, the existing SSR methods deal with outliers in two major ways, Projection Approach [13] and Joint Approach [14]. Let V denote the subspace spanned by the columns of A, and let $F \in R^{(m-n)\times m}$ be a matrix whose rows form an orthobasis of $V^{\perp}$. Then we have $FA = 0$. The Projection Approach applies F to the measurements and from (2.2) we obtain

$$b \triangleq Fy = FAx + Fe + F\eta = Fe + F\eta. \tag{2.3}$$

The original problem is transferred to the recovery of a sparse vector e, given the under-determined measurement matrix F and noisy measurements b. Various SSR methods can be directly applied to solve this problem, such as BSRR [15], [16] which is based on Sparse Bayesian Learning (SBL) [17], [18], and Second-Order Cone Programming (SOCP) [10] which is based on ℓ_1 minimization [19]–[21]. Note that the ℓ_1 estimator ($\arg\min_x \|y - Ax\|_1$) was shown to be equivalent to the SOCP case of no dense inlier noise [13]. The Joint Approach reformulates the original model into $y = [A\ I_{m\times m}]\begin{bmatrix} x \\ e \end{bmatrix} + \eta$, where $[A\ I_{m\times m}]$ is under-determined and the lower part of $\begin{bmatrix} x \\ e \end{bmatrix}$ is sparse. Many existing SSR methods can be extended to deal with this formulation via restricting the lower part of $\begin{bmatrix} x \\ e \end{bmatrix}$ to be sparse, e.g., BPRR which is based on ℓ_1 minimization [16], ℓ_p ($0< p \leq 1$) regularization which assumes a super-Gaussian prior for e to encourage sparsity [14], [22], Giannakis's algorithm for robust sensing [23] that utilizes a log-sum penalty function [24]–[27], Jin's empirical Bayesian inference-based algorithm which is extended from SBL [22], and GARD [3] which is based on Orthogonal Matching Pursuit (OMP) [28][29]. An important finding in sparse recovery theory is that although finding the sparsest solution from under-determined linear equations is also of a combinatorial nature, some polynomial-time sparse recovery methods are guaranteed to find the sparsest solution under certain conditions on the sparsity of e and conditioning of matrix F [69], [70]. It was shown in [16] that BSRR outperforms LMedS and RANSAC.

The key to successful sparse recovery lies in identifying the support (nonzero entries), as one can simply add a reprojection step to estimate magnitude later. We propose a novel objective function and corresponding algorithm to help identify the *support of outliers*. The method is developed under the paradigm of the Joint Approach, but there is a fundamental difference with existing SSR methods. The existing methods often tackle the ℓ_0-'norm' of e implicitly (e.g., via OMP or SBL), or through the use of surrogate measures for the ℓ_0-'norm', such as the log-sum function or the ℓ_p-norm ($0< p \leq 1$). Besides these methods, the hard thresholding based iterative method [11] shows its equivalence with a family (infinitely many)

of nonconvex penalties for e (plus the ℓ_2-norm on the noise term), thus promoting the sparsity of e (the author noted that this method relies on a preliminary robust fit). In contrast to all these methods, we explicitly model and operate on the ℓ_0-'norm' of e, and the optimization procedure naturally utilizes the large observation error prior, and does not need a preliminary robust fit. Theoretical guarantees regarding exact recovery or error bounds are derived to support the efficacy of the method. The overall best performance in terms of the quality of recovery and lower complexity (over competing methods) further demonstrates the notable benefits of the proposed method.

The remainder of the chapter is organized as follows: In Section 2.2, we introduce the nonconvex objective function and the associated optimization procedure to help identify the support of outliers to be used in the reprojection step. Section 2.3 gives theoretical results regarding its convergence, exact recovery or recovery error. We empirically study the performance of the proposed method and compare with other state-of-the-art methods in Section 2.4. Conclusions are made in Section 2.5.

Notation: Capital letters denote matrices, e.g., A, while lowercase letters denote vectors, e.g., e. The ith row of matrix A is denoted by a_i^T, while the ith element of vector e is denoted by e_i. The ℓ_0-'norm'[1] of e, i.e., $\|e\|_0$, counts the number of nonzero elements of e. Bold capital letters are reserved for sets, e.g., $\boldsymbol{S}$, where $\boldsymbol{S}^c$ and $|\boldsymbol{S}|$ denote the complement and the cardinality of $\boldsymbol{S}$ respectively, and $\boldsymbol{S}_k$ denotes the set $\boldsymbol{S}$ obtained from the kth iteration. We use $A_{\boldsymbol{S}}$ to denote the $|\boldsymbol{S}| \times n$ submatrix of A containing the rows indexed by $\boldsymbol{S}$. Similarly, $e_{\boldsymbol{S}}$ denotes the subvector of e containing the entries indexed by $\boldsymbol{S}$. The indicator function is denoted as $I(\cdot)$.

[1] ℓ_0-'norm'is not a norm as it does not satisfy the axioms of a norm.

2.2 Robust Linear Regression via ℓ_0 Regularization

We propose minimizing the following objective function to help identify the support of outliers.

$$J(x, e) = \|y - Ax - e\|_1 + \alpha\|e\|_0. \tag{2.4}$$

In the second term, we directly use the ℓ_0-'norm' to enforce the sparseness in the outlier corruptions e, rather than relaxing it to the ℓ_p-norm ($0< p \leq 1$).

We use the alternating minimization "like" approach to minimize the nonconvex objective function in (2.4). The detailed procedure is summarized in Algorithm 1, where $x^{(k+1)}$ and $e^{(k+1)}$ denote the updated x and e at the $(k+1)$ st iteration. $\boldsymbol{S}_k$ is the complementary set of the support of $e^{(k)}$, which is the index set for "valid" entries of y that are estimated to be free of outliers in the kth iteration. Here the convergence of $J(x, e)$ means $J(x^{(k+1)}, e^{(k+1)}) = J(x^{(k)}, e^{(k)})$, and $\boldsymbol{S}_k = \boldsymbol{S}_{k-1}$ is a sufficient condition for convergence (see Appendix 2.6.1).

Algorithm 1 Algorithm for Robust Outlier Support Identification (AROSI)

Input: y, A, $\alpha > 0$

Initialization: $k = 0$, $e^{(0)} = \boldsymbol{0}$, $\boldsymbol{S}_0 = \{1, \ldots, m\}$

While $J(x, e)$ **not converged DO:**

Iteration $k + 1$

Step 1 (update x): $x^{(k+1)} = \arg\min_x \|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x\|_1$;

If $\|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x^{(k+1)}\|_1 = \|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x^{(k)}\|_1$, further update $x^{(k+1)} = x^{(k)}$.

Step 2 (update e and $\boldsymbol{S}$): $e_i^{(k+1)} = \begin{cases} 0, & |(y - Ax^{(k+1)})_i| \leq \alpha \\ (y - Ax^{(k+1)})_i, & \text{otherwise} \end{cases}$

$\boldsymbol{S}_{k+1} := \{i : e_i^{(k+1)} = 0\}$

$k := k + 1$

End While

Output: solution $\tilde{x}$

At first glance, it seems more reasonable to use the ℓ_2-norm rather than the ℓ_1-norm in

the first term of the objective function (2.4) and in Step 1, especially for Gaussian noise. Here we emphasize that the minimizer of the objective function (2.4) is not our final solution; it will be followed by a reprojection step described later. In Step 1 of each iteration, we only use our estimated "valid" outlier free entries/rows indicated by S to estimate x. However, we do not expect that all the outliers are identified by the previous iteration; it is very likely that some outliers have not been removed. So it is safer to use the ℓ_1-norm in Step 1, as the ℓ_1 estimator is more robust to outliers than OLS. In case there are multiple solutions[2] [71] for $\min_x \|y_{S_k} - A_{S_k}x\|_1$ and $x^{(k)}$ happens to be one of these solutions, we set $x^{(k+1)} = x^{(k)}$ to make the algorithm more stable.

At the beginning, we have no information about the positions of outliers except that they are sparse. So we simply initialize $e^{(0)} = 0$, and index set $\boldsymbol{S}_0 := \{i : e_i^{(0)} = 0\} = \{1, \ldots, m\}$. So in Step 1 of the first iteration, all the data will be used and it is equivalent to the ℓ_1 estimator, which has been justified by many authors (e.g., [13], [72]).

In Step 2, when x is fixed, define $r \triangleq y - Ax$,

$$\min_e(\|y - Ax - e\|_1 + \alpha\|e\|_0) = \min_e(\|r - e\|_1 + \alpha\|e\|_0)$$

$$= \min_e \textstyle\sum_{i=1}^m (|r_i - e_i| + \alpha I(e_i \neq 0)) = \sum_{i=1}^m \min_{e_i}(|r_i - e_i| + \alpha I(e_i \neq 0))$$

$$\hat{e}_i := \begin{cases} 0, & |r_i| \leq \alpha \\ r_i, & \text{otherwise} \end{cases} \in \arg\min_{e_i}(|r_i - e_i| + \alpha I(e_i \neq 0)), \tag{2.5}$$

[2]In practice, when A_{S_k} is full column rank, this rarely happens, and we have not experienced this in our numerical experiments.

and

$$\min_{e_i}(|r_i - e_i| + \alpha I(e_i \neq 0)) = \min(|r_i|, \alpha) = \begin{cases} |r_i|, & |r_i| \leq \alpha \\ \alpha, & \text{otherwise} \end{cases}. \tag{2.6}$$

We can see from (2.5) that Step 2 directly promotes the sparsity of e via hard thresholding. Any entry of $|y - Ax|$ larger than α will be considered an outlier corrupted entry. In general, α should be set at least larger than the inlier noise level. Our analysis shows that under certain reasonable conditions on the model parameters, if α is greater than some certain threshold, we can guarantee that all the inliers are kept in *every* iteration. Conservatively, one may use a very large α, aiming to keep most of the inliers while safely removing some large outliers. Alternatively, one may use a small α (e.g., 4σ), aiming to get rid of more outliers, with the possibility one may also lose more inliers. If there is no prior knowledge of σ, it can be estimated from the residuals of the ℓ_1 estimation (which is also Step 1 of our first iteration) [6]: $\hat{\sigma} = \frac{1}{0.675} median(|r_i^{(1)}| \mid r_i^{(1)} \neq 0)$.

Reprojection Step for the Joint Approach: Our theoretical results in Section 2.3 show that AROSI can guarantee the exact support recovery of outliers. This motivates us to add a reprojection step in the end. The reprojection step [73] is widely used in sparse recovery methods; it often improves the estimation of the magnitudes of the nonzero entries. In the Projection Approach, as the original problem is transferred to the conventional sparse recovery problem form, it is straightforward to use reprojection (e.g., [10]). Here we present the reprojection step for the Joint Approach. Recall that the original model (2.2) is reformulated as $y = [A\ I_{m\times m}] \begin{bmatrix} x \\ e \end{bmatrix} + \eta$, where the lower part of $\begin{bmatrix} x \\ e \end{bmatrix}$ is sparse. With estimated x or e by some Joint Approach algorithm, the reprojection step is as follows:

1. Estimate the support $\boldsymbol{E}$ of e by thresholding $|\tilde{e}|$ or $|y - A\tilde{x}|$, e.g., $\hat{\boldsymbol{E}} := \{i : |\tilde{e}_i| > p\sigma\}$, where σ is the standard deviation of the inlier noise, and p is a scaling factor.

2. Regress y onto the selected columns of $[A\ I_{m\times m}]$, i.e., $[A\ (I_{\hat{\boldsymbol{E}}})^T]$ by least squares:

$$\hat{z} = \arg\min_{z} \|y - [A\ (I_{\hat{\boldsymbol{E}}})^T]z\|_2. \tag{2.7}$$

3. Finally, obtain $\hat{x} = \hat{z}_{\{1,\dots,n\}}$, and $\hat{e}_{\hat{\boldsymbol{E}}} = \hat{z}_{\{n+1,\dots,end\}}$, which is the estimated outlier corruption values corresponding to $\hat{\boldsymbol{E}}$.

In general, setting p is a tradeoff between false alarms and false negatives in identifying outliers, and so a relatively small p is recommended to have fewer false negatives. If it is known that the magnitudes of outliers are much larger than inlier noise (or if we are less concerned about the noise level outliers), a slightly larger p can be employed to decrease false alarms. When thresholding $|y - A\tilde{x}|$, since the inlier noise is present in this residual, the scaling factor p should be greater than 2. While when thresholding $|\tilde{e}|$, since e is already separated from the inlier noise in the model, a small p can be employed, e.g., [10] uses $p = 1$ in their Projection Approach.

A sufficient condition for $[A\ (I_{\hat{\boldsymbol{E}}})^T]$ to be full column rank is $|\hat{\boldsymbol{E}}| \leq \max(2\mathcal{m}(A) - 1, 0)$ (defined in Definition 2.1, guaranteed by Theorem 2.2). When $p \to \infty$, $|\hat{\boldsymbol{E}}| \to 0$. In case the generated $[A\ (I_{\hat{\boldsymbol{E}}})^T]$ is under-determined or not full column rank, we can always increase the scaling factor p to make $[A\ (I_{\hat{\boldsymbol{E}}})^T]$ full column rank, thus (2.7) has a unique solution.

The major difference with the reprojection step in the Projection Approach is the alternative way to estimate the support of e, i.e., via thresholding $|y - A\tilde{x}|$, if we have more confidence in estimated $\tilde{x}$ than $\tilde{e}$. In AROSI, we are more confident about the estimated $\tilde{x}$, as it is less sensitive to the parameter α than $\tilde{e}$. So, to estimate the support of e, we threshold $|y - A\tilde{x}|$, i.e., $\hat{\boldsymbol{E}} := \{i : |(y - A\tilde{x})_i| > p\sigma\}$.

Complexity: AROSI alternates between ℓ_1 estimation (Step 1) and entrywise thresholding (Step 2). So the main computational step (complexity) is ℓ_1 estimation in each iteration, which can be recast as Linear Programming. If AROSI converges in K iterations (usually a

few iterations), the worst run time estimate will be K times the run time of the ℓ_1 estimator. In fact, the total run time is often less than that. This is not only because some entries are pruned in Step 1, but also because the result of the previous iteration is used as the initial point for the current iteration (*a.k.a.* warm-start). This is usually a good initial point and improves the speed of ℓ_1.

2.3 Theoretical Analysis

In this section, we analyze AROSI (without adding the reprojection step unless otherwise noted) and establish some theoretical guarantees which support its robustness and effectiveness. The theoretical results depend on the matrix A, the bounds for the inlier noise, and the sparsity of the outlier component. The exact conditions are included as part of the theorem statements. The main results include the following:

1. Exact recovery of the signal under any parameter setting in the presence of outliers only, i.e., absence of dense inlier noise (Theorem 2.3).

2. The recovery error is bounded in the noisy case, and the bound is smaller than that of the ℓ_1 estimator (Theorem 2.6).

3. Exact support recovery of outliers in both no dense inlier noise case (Theorem 2.3) and noisy case (Theorem 2.6.d).

4. The ability to keep all the inliers and remove significant outliers in every iteration (Theorems 2.3, 2.6-2.7, and Remark 2.2).

5. Even if the number of outliers is greater than the regression breakdown point of the ℓ_1 estimator, AROSI can still give exact recovery (no dense inlier noise case, Remarks 2.1 and 2.2) or bounded estimation error (noisy case, Remark 2.5 and Theorem 2.7). It can tolerate 50% more outliers than the ℓ_1 estimator under certain conditions.

6. AROSI has desirable recovery performance when the rows of design matrix are i.i.d. from the uniform distribution on the unit sphere $\mathbb{S}^{n-1}$. (Which is shown to have a large $m(A)$ [74], see Definition 2.1. And further due to the slowly decreasing property of $m(A)$, i.e., Theorem 2.2.)

2.3.1 Convergence Property

Note that Step 1 of the algorithm deviates from the standard alternating minimization approach. Thus, the convergence of the algorithm is not assured based on the alternating minimization framework and needs to be established.

Theorem 2.1. *AROSI converges in a finite number of iterations to a fixed point, which is a local optimum. Moreover, the objective function is strictly decreasing before convergence.*

The proof of the theorem is in Appendix 2.6.1.

2.3.2 Characterization of AROSI when Only Outliers Present

Here we discuss the case when there are only sparse outliers present and no dense inlier noise. Our model in (2.2) degenerates to $y = Ax + e$. The analysis benefits greatly from the analysis of the ℓ_1 estimator in [72], which is equivalent to the Step 1 of our first iteration. We further build and extend the work to understand AROSI, based on an important property stated in Lemma 2.1. We first introduce some definitions and properties regarding the leverage constants and their related quantity $m(A)$ for matrix A that are important to the analysis.

Definition 2.1 *(from [72]): Define $M = \{ 1, \ldots, m \}$ as the index set of all the observations.*

Define for every $q \in \{1, \ldots, m\}$ the leverage constants c_q of A as

$$c_q(A) = \min_{\substack{\boldsymbol{E}\subset\boldsymbol{M}\\|\boldsymbol{E}|=q}} \min_{\substack{g\in R^n\\ g\neq 0}} \frac{\sum_{i\in\boldsymbol{M}\setminus\boldsymbol{E}} |a_i^T g|}{\sum_{i\in\boldsymbol{M}} |a_i^T g|} = \min_{\substack{\boldsymbol{E}\subset\boldsymbol{M}\\|\boldsymbol{E}|=q}} \min_{\substack{g\in R^n\\ \|g\|_2=1}} \frac{\sum_{i\in\boldsymbol{M}\setminus\boldsymbol{E}} |a_i^T g|}{\sum_{i\in\boldsymbol{M}} |a_i^T g|},$$

and $\mathcal{m}(A) = \max\{q \in \boldsymbol{M} \mid c_q(A) > \frac{1}{2}\}$.

Note that [75] provides an algorithm to compute $\mathcal{m}(A)$ for any given A. The complexity is $O(\binom{m}{n}(n^3 + m^2))$, which is prohibitive for large m and n, making the computation of $\mathcal{m}(A)$ limited to a small size matrix A.

Proposition 2.1 *(from [72]): $c_0(A) = 1, c_m(A) = 0$, and for every $q \in \{1, \ldots, m\}$, $c_q(A) \leq c_{q-1}(A)$.*

Proposition 2.2 *If $\mathcal{m}(A) \geq q$, then we must have $c_q(A) > \frac{1}{2}$, and $m > 2q$.*

The proof can be found in the supplemental material.

In [72], it is shown that the regression breakdown point of the ℓ_1 estimator is $\mathcal{m}(A)+1$. Since in the iterations of AROSI, it detects and removes 'outliers' and uses the remaining entries to do ℓ_1 estimation, two fundamental questions arise: When deleting some entries, 1) will the regression matrix become singular? 2) how does $\mathcal{m}(A)$ change (will it suddenly become 0)? The following Lemma 2.1 and Theorem 2.2 address these concerns.

Lemma 2.1 *Let matrix A be full column rank and $\mathcal{m}(A) \geq q$. Then for any index set $\boldsymbol{T} \subset \boldsymbol{M}, |\boldsymbol{T}| = t \leq q$, we have that $A_{\boldsymbol{T}^c}$ must be full column rank, $\mathcal{m}(A_{\boldsymbol{T}^c}) \geq q - \lceil 0.5t \rceil \geq q - t$, and $c_{q-t}(A_{\boldsymbol{T}^c}) \geq c_{q-\lceil 0.5t\rceil}(A_{\boldsymbol{T}^c}) \geq c_q(A) > \frac{1}{2}$.*

The proof of the lemma is in Appendix 2.6.2.

Theorem 2.2. *Let matrix A be full column rank and $\mathcal{m}(A) \geq q > 0$. Then for any index set $\boldsymbol{T} \subset \boldsymbol{M}, |\boldsymbol{T}| = t \leq 2q - 1$, we have that $A_{\boldsymbol{T}^c}$ must be full column rank, $\mathcal{m}(A_{\boldsymbol{T}^c}) \geq q - \lceil 0.5t \rceil$, and $c_{q-\lceil 0.5t\rceil}(A_{\boldsymbol{T}^c}) \geq c_q(A) > \frac{1}{2}$.*

The proof utilizes the above Lemma and can be found in the supplemental material.

The above theorem is significant because it characterizes the slowly decreasing property of $m(A)$ w.r.t. m (the number of rows of A), which enables AROSI to go beyond ℓ_1 estimation and deal with more outliers, as we will show later.

Now we first introduce our main theorem of exact recovery when $\|e\|_0 \leq m(A)$.

Theorem 2.3. *AROSI running with any $\alpha > 0$ will find x exactly if $\|e\|_0 \leq m(A)$. If additionally $\alpha < \min\{|e_i| : e_i \neq 0\}$, AROSI will find both x and e exactly.*

Proof: Proved as a special case of Theorem 2.6 with $\eta = 0$.

Actually when $\|e\|_0 \leq m(A)$, AROSI running with *any* $\alpha > 0$ recovers x exactly in every iteration, so it will converge in 2 iterations.

The above theorem shows the robustness of AROSI in two contexts: First, it succeeds in a wide range of parameter settings; Second, it is robust to the undetected outliers (even if α is set too large such that only a few outliers are detected). This robustness is a result of the slowly decreasing property of $m(A)$ w.r.t. m. When only sparse outliers are present, we want the first term in the objective function (2.4) to be 0, as there is no dense inlier noise. We need to put infinitely large weight on the first term, or equivalently, set $\alpha \to 0^+$ in the second term. So $\alpha < \min\{|e_i| : e_i \neq 0\}$ will be satisfied. Then we can recover both x and e exactly under the given condition. When $\alpha \to 0^+$, minimizing the objective function (2.4) is equivalent to the following problem:

$$\min_{e,\ x} \|e\|_0 \quad s.t.\ y = Ax + e, \tag{2.8}$$

which is the problem of interest when there is no dense noise, under the principle of fitting most of the data, and which would give exact recovery under mild conditions [16]. To minimize our objective function (2.4) with $\alpha \to 0^+$, AROSI starts with $\min_x \|y - Ax\|_1$, which is proven to give exactly the same solution as (2.8) under certain conditions [13][16]. The above analysis gives a justification for our objective function (2.4) and AROSI.

Next, we deal with the case where $\|e\|_0 > m(A)$.

Suppose $\|e\|_0 \leq m(A)$ is not satisfied for the ℓ_1 estimator, which is also Step 1 in our first iteration. In the following steps we remove some entries that may contain both inliers and outliers. If the number of remaining outliers $\|e_{S_k}\|_0 \leq m(A_{S_k})$, we can recover x exactly (see quoted Theorem in Appendix 2.6.3).

The key question is whether it is possible that $\|e_{S_k}\|_0 \leq m(A_{S_k})$, given that $\|e\|_0 > m(A)$. Theorem 2.2 shows the slowly decreasing property of $m(A)$, which makes it possible.

Remark 2.1 *Suppose that $\|e\|_0 > m(A) \geq q$, and that when AROSI converges at the $(k+1)$ st iteration, $|\boldsymbol{S}_k^c| = t$, i.e., we have removed t entries. Among these t entries, $p \times t$ of them are outliers, so $\|e_{S_k}\|_0 = \|e\|_0 - p \times t$. When $t \leq 2q-1$, from Theorem 2.2, we know that A_{S_k} is full column rank and $m(A_{S_k}) \geq q - \lceil 0.5t \rceil$. So if $\|e\|_0 - p \times t \leq q - \lceil 0.5t \rceil$, i.e., $p \geq \frac{\|e\|_0 + \lceil 0.5t \rceil - q}{t}$, we can guarantee the exact recovery of x. When $t > 2q-1$, then a sufficient condition for exact recovery of x is that A_{S_k} has full column rank and $p = \frac{\|e\|_0}{t}$, i.e., all the outliers are within the t removed entries.*

The exact recovery test in Section 2.4.2 demonstrates that there are cases where the ℓ_1 estimator fails (this must be the case $\|e\|_0 > m(A)$ according to the quoted theorem in Appendix 2.6.3) while AROSI gives exact recovery.

Remark 2.2 *In case both large outliers and moderate outliers exist, as a special case of Theorem 2.7 with $\eta = 0$, we show that under certain conditions AROSI can recover x exactly even if there are up to $\lfloor 1.5 \times m(A) \rfloor$ outliers. More specifically, when $0 < m(A) \leq \|e\|_0 \leq m(A) + \lfloor \frac{t}{2} \rfloor$, where $1 \leq t \leq m(A)$, define $\boldsymbol{G} :=$ {indices of $m(A)$ largest entries of $|e|$}, $\boldsymbol{P} :=$ {indices of t largest entries of $|e|$}. If $\min\{|e_i| : i \in \boldsymbol{P}\} > \frac{2\sum_{i \in \boldsymbol{E} \setminus \boldsymbol{G}} |e_i|}{c_{m(A)}(A) - 0.5}$, then any α satisfying $\frac{\sum_{i \in \boldsymbol{E} \setminus \boldsymbol{G}} |e_i|}{c_{m(A)}(A) - 0.5} < \alpha < \min\{|e_i| : i \in \boldsymbol{P}\} - \frac{\sum_{i \in \boldsymbol{E} \setminus \boldsymbol{G}} |e_i|}{c_{m(A)}(A) - 0.5}$ guarantees the exact recovery of x from the second iteration, and it will converge in no more than three iterations. It is natural to think about this guarantee in comparison with the so called "masking effect" [76], where some*

extreme outliers (e.g., those indexed by $\boldsymbol{P}$), help hide another group of mild but perhaps more structured outliers (e.g., indexed by $\boldsymbol{E}\backslash\boldsymbol{G}$), which are usually more difficult to detect. AROSI effectively identifies and removes those extreme outliers, and more importantly, is resistant to the remaining unidentified outliers and recovers x exactly.

2.3.3 Both Dense Noise and Sparse Outliers Present

Now we deal with the more general case where both dense inlier noise and sparse outliers exist. In the first subsection, we establish the error bound for AROSI. Then we characterize the behaviors of AROSI in the second subsection.

2.3.3.1 Recovery Error Bound

We first quote a definition and theorem from [75] regarding the ℓ_1 estimation error bound, and present our Corollary 2.1, which establishes the bound for AROSI.

Definition 2.2 *(from [75]): Given an arbitrary $q \in \{0, 1, \ldots, m\}$, we call a set $\boldsymbol{B}$ a possibly extreme set if there exists a set $\boldsymbol{L}, \boldsymbol{L} \supseteq \boldsymbol{B}, |\boldsymbol{L}| = m - q$, such that the following holds:*

$$\sum_{i \in \boldsymbol{B} \bigcup \boldsymbol{L}^c} |a_i^T v| \geq \sum_{i \in (\boldsymbol{L} \backslash \boldsymbol{B})} |a_i^T v|, \tag{2.9}$$

where v is any of the singular vectors corresponding to the smallest singular value of the $|\boldsymbol{B}| \times n$ submatrix $A_{\boldsymbol{B}}$ of A : $\|A_{\boldsymbol{B}} v\|_2 = \sigma_{\min}(A_{\boldsymbol{B}})\|v\|_2$. We define $\boldsymbol{Q}_q$ to be the set of all possibly extreme sets for a given q.

Theorem 2.4. *(from [75]) Let $y = Ax + e + \eta$, $\boldsymbol{E} = supp(e)$, the ℓ_1 estimation error is bounded as follows:*

$$\|x_{\ell_1} - x\|_2 \leq \left(\max_{\boldsymbol{B} \in \boldsymbol{Q}_{|\boldsymbol{E}|}} \frac{1}{\sigma_{\min}(A_{\boldsymbol{B}})} \right) \|\eta\|_2. \tag{2.10}$$

It can be proved that if $|\boldsymbol{E}| \leq m(A)$, then $\forall \boldsymbol{B} \in \boldsymbol{Q}_{|\boldsymbol{E}|}$, $\sigma_{\min}(A_{\boldsymbol{B}}) > 0$.

Now we are ready to establish the error bound for AROSI.

Corollary 2.1 *In the* $(k+1)$ *st iteration of AROSI, define the index set* $\boldsymbol{R} := \boldsymbol{E} \cap \boldsymbol{S}_k$. *If* $|\boldsymbol{R}| \leq m(A_{\boldsymbol{S}_k})$, *and* $A_{\boldsymbol{S}_k}$ *has full column rank, then the following holds for* $x^{(k+1)}$ *:*

$$\|x^{(k+1)} - x\|_2 \leq \Big(\max_{\boldsymbol{B}' \in \boldsymbol{Q}'_{|\boldsymbol{R}|}} \frac{1}{\sigma_{\min}((A_{\boldsymbol{S}_k})_{\boldsymbol{B}'})}\Big)\|\eta_{\boldsymbol{S}_k}\|_2, \tag{2.11}$$

where $\sigma_{\min}((A_{\boldsymbol{S}_k})_{\boldsymbol{B}'}) > 0$, $\forall \boldsymbol{B}' \in \boldsymbol{Q}'_{|\boldsymbol{R}|}$. *Here* $\boldsymbol{Q}'_q$ *follows the same definition in Definition 2.2, except that A is replaced by* $A_{\boldsymbol{S}_k}$, *and m is replaced by the number of rows of* $A_{\boldsymbol{S}_k}$.

Proof: This is apparent from Theorem 2.4, as $x^{(k+1)}$ is the ℓ_1 estimate on the model $y_{\boldsymbol{S}_k} = A_{\boldsymbol{S}_k} x + e_{\boldsymbol{S}_k} + \eta_{\boldsymbol{S}_k}$, and $\boldsymbol{R} = \boldsymbol{E} \cap \boldsymbol{S}_k$ corresponds to $supp(e_{\boldsymbol{S}_k})$.

Remark 2.3 $\boldsymbol{R} := \boldsymbol{E} \cap \boldsymbol{S}_k$ *is the index set of outliers that remained in* $\boldsymbol{S}_k$. *Note that Corollary 2.1 does not need the initial condition* $|\boldsymbol{E}| \leq m(A)$. *It only needs the number of remaining outliers* $|\boldsymbol{R}| \leq m(A_{\boldsymbol{S}_k})$, *which can be guaranteed by* $|\boldsymbol{E}| \leq m(A)$ *and proper* α *(see Remark 2.4) for any* $k \in Z_{\geq 0}$. *Even if* $|\boldsymbol{E}| > m(A)$, *it is still possible that* $|\boldsymbol{R}| \leq m(A_{\boldsymbol{S}_k})$ *for any* $k \in Z_{\geq 1}$, *e.g., under the condition of Theorem 2.7 (details can be found in the proof).*

Then a natural question of interest is whether the bound for AROSI is better than that of the ℓ_1 estimator. The following theorem provides a positive answer.

Theorem 2.5. *Let* $y = Ax + e + \eta$, $\boldsymbol{E} = supp(e)$, $|\boldsymbol{E}| = q \leq m(A)$. *In the* $(k+1)$ *st iteration of AROSI, if* $\boldsymbol{E}^c \subseteq \boldsymbol{S}_k$, *then* $\|x^{(k+1)} - x\|_2$ *is bounded as in (2.11), and the bound is smaller than or equal to the bound in (2.10).*

The proof of the theorem is in Appendix 2.6.5.

Theorem 2.5 is applicable for *any* iteration. The condition $\boldsymbol{E}^c \subseteq \boldsymbol{S}_k$ required by Theorem 2.5 can be guaranteed with proper α, given $|\boldsymbol{E}| \leq m(A)$, as we will see in Theorem 2.6.a,

and it follows immediately that the bound for Theorem 2.6.c is smaller than or equal to the bound for ℓ_1 estimation error provided in Theorem 2.4.

2.3.3.2 Characterization of AROSI in Noisy Case

In this subsection, we first present Lemma 2.2, which describes the behavior of AROSI in any iteration and is an important step in deriving our main results in Theorems 2.6 and 2.7.

Lemma 2.2 *Let $y = Ax + e + \eta$ and $\boldsymbol{E} = supp(e)$ satisfying $|\boldsymbol{E}| = q \leq m(A)$. Denote $r^{(k+1)}_{\boldsymbol{S}_k} = y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x^{(k+1)}$. If $\boldsymbol{S}_k \supseteq \boldsymbol{E}^c$ for a particular k, then we must have $A_{\boldsymbol{S}_k}$ full column rank, $m(A_{\boldsymbol{S}_k}) \geq q - |\boldsymbol{S}_k^c|$, and $\|(e+\eta)_{\boldsymbol{S}_k} - r^{(k+1)}_{\boldsymbol{S}_k}\|_1 \leq \frac{\sqrt{m-q}\|\eta\|_2}{c_q(A)-0.5}$. Also $\forall i \in \boldsymbol{E}^c$, $|r_i^{(k+1)}| \leq \|\eta\|_\infty + \frac{\sqrt{m-q}\|\eta\|_2}{c_q(A)-0.5}$.*

The proof of the theorem is in Appendix 2.6.6.

Now we are in position to present our main results in the noisy case. Theorem 2.6 shows that when $\|e\|_0 \leq m(A)$, the estimation error of AROSI (with proper α) is bounded, and from Theorem 2.5 we know its bound is smaller than or equal to the ℓ_1 estimation error bound.

Theorem 2.6. *Let $y = Ax + e + \eta$, $\boldsymbol{E} = supp(e)$ and $|\boldsymbol{E}| = q \leq m(A)$. Define $C_1 = \frac{\sqrt{m-q}\|\eta\|_2}{c_q(A)-0.5}$, $C_2 = \max(C_1, \frac{2\sqrt{m-q}\|\eta\|_2 \sigma_{\max}(A_{\boldsymbol{E}})}{\sigma_{\min}(A_{\boldsymbol{E}^c})})$, $C_3 = \frac{\sigma_{\max}(A_{\boldsymbol{E}})}{\sigma_{\min}(A_{\boldsymbol{E}^c})} C_1$. For any $\alpha > \|\eta\|_\infty + C_1$, AROSI guarantees that:*

1) *All the inlier entries (indexed by $\boldsymbol{E}^c$) are kept in every iteration (i.e., $\boldsymbol{E}^c \subseteq \boldsymbol{S}_k$ for any $k \in Z_{\geq 0}$);*

2) *Significant outlier entries indexed by $\boldsymbol{P} := \{i : |e_i| > \alpha + \|\eta\|_\infty + C_3\}$ are identified and removed in every iteration (i.e., $\boldsymbol{P} \subseteq \boldsymbol{S}_{k+1}^c$ for any $k \in Z_{\geq 0}$);*

3) *$\|x^{(k+1)} - x\|_2$ is bounded for any $k \in Z_{\geq 0}$.*

Moreover, if $\min\{|e_i| : e_i \neq 0\} > 2\|\eta\|_\infty + C_1 + C_2$, then any α satisfying $\|\eta\|_\infty + C_1 < \alpha < \min\{|e_i| : e_i \neq 0\} - \|\eta\|_\infty - C_2$ for AROSI guarantees that:

4) AROSI converges in 3 iterations, and the support of e is recovered exactly;

5) After the reprojection step (whose threshold is within the range $(\|\eta\|_\infty + C_1, \min\{|e_i| : e_i \neq 0\} - \|\eta\|_\infty - C_2)$ *) , we have* $\|\hat{x} - x\|_2 \leq \|\eta_{\boldsymbol{E}^c}\|_2 / \sigma_{\min}(A_{\boldsymbol{E}^c})$.

The proof of the theorem is in Appendix 2.6.7.

Remark 2.4 *In Theorem 2.6.e, x is equivalent to the least squares solution on all the inlier entries. The bound is tight and is better than the bound in (2.10) (details in the proof). Theorem 2.6 is an exciting result for the noisy case: If the large magnitude corruptions are sparse ($\|e\|_0 \leq m(A)$), with proper value of α (which depends on the inlier noise level, matrix A, and the sparsity of outliers, and does not depend on the magnitude of outliers), we can guarantee that all the inliers are kept in every iteration. At the same time, all the removed entries are guaranteed to be outliers. This shows another aspect of AROSI robustness: under certain conditions there are no false alarms when identifying and removing outliers during iterations. Purely removing some outliers often leads to better signal estimation in our Step 1 ($x^{(k+1)} = \arg\min_x \|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x\|_1$) than the ℓ_1 estimation, especially as we can also guarantee (by Lemma 2.2) that $A_{\boldsymbol{S}_k}$ is full column rank and the number of remaining outliers $(|\boldsymbol{E}| - |\boldsymbol{S}_k^c|) \leq m(A_{\boldsymbol{S}_k})$ for any $k \in Z_{\geq 1}$. In addition, we can also guarantee that the significant outliers, which are usually the most troublesome ones, are identified and removed in every iteration. Further, if the magnitudes of the corruptions are all large enough, we can even guarantee all the outliers are removed in every iteration. Finally, note that we showed that the estimation error is bounded in every iteration.*

The following Remark 2.5 and Theorem 2.7 demonstrate that even if $\|e\|_0 > m(A)$ (recall that $m(A) + 1$ is the regression breakdown point of the ℓ_1 estimator [72]), AROSI can still provide a bounded estimation error.

Remark 2.5 *When $\|e\|_0 > m(A)$, we have provided a sufficient requirement in Remark 2.1 to satisfy the condition of Corollary 2.1, thus guaranteeing that the estimation error of x by AROSI is bounded in the noisy case.*

In the following theorem, we establish conditions under which AROSI is guaranteed to handle more than $m(A)$ outliers.

Theorem 2.7. *Suppose $y = Ax + e + \eta$, $\boldsymbol{E} = supp(e)$, $0 < m(A) \leq |\boldsymbol{E}| = q \leq m(A) + \left\lfloor \frac{t}{2} \right\rfloor$, where $1 \leq t \leq m(A)$. Define $\boldsymbol{G} := \{$indices of $m(A)$ largest entries of $|e|\}$,*
$\boldsymbol{P} := \{$*indices of t largest entries of* $|e|\}$, $q_1 = m(A)$, $q_2 = m(A_{\boldsymbol{P}^c})$,
$w_1 = \max(\frac{\sqrt{m-q_1}\|\eta\|_2 + \sum_{i\in \boldsymbol{E}\setminus \boldsymbol{G}}|e_i|}{c_{q_1}(A)-0.5}, \frac{\sqrt{m-q}\|\eta\|_2}{c_{q_2}(A_{\boldsymbol{P}^c})-0.5})$, $w_2 = \max(\frac{\sqrt{m-q_1}\|\eta\|_2 + \sum_{i\in \boldsymbol{E}\setminus \boldsymbol{G}}|e_i|}{c_{q_1}(A)-0.5}, \frac{\sigma_{\max}(A_{\boldsymbol{P}})\sqrt{m-q}\|\eta\|_2}{\sigma_{\min}(A_{\boldsymbol{E}^c})\times\left(c_{q_2}(A_{\boldsymbol{P}^c})-0.5\right)})$.
If $\min\{|e_i| : i \in \boldsymbol{P}\} > 2\|\eta\|_\infty + w_1 + w_2$, *then any* α *satisfying* $\|\eta\|_\infty + w_1 < \alpha < \min\{|e_i| : i \in \boldsymbol{P}\} - \|\eta\|_\infty - w_2$ *for AROSI guarantees that:*

1) *All the inlier entries (indexed by $\boldsymbol{E}^c$) are kept in every iteration (i.e., $\boldsymbol{E}^c \subseteq \boldsymbol{S}_k$ for any $k \in Z_{\geq 0}$);*

2) *Significant outlier entries indexed by $\boldsymbol{P}$ are identified and removed in every iteration (i.e., $\boldsymbol{P} \subseteq \boldsymbol{S}_{k+1}^c$ for any $k \in Z_{\geq 0}$);*

3) *$\|x^{(k+1)} - x\|_2$ is bounded for any $k \in Z_{\geq 1}$.*

The proof of the theorem is in Appendix 2.6.8.

As our first iteration is equivalent to ℓ_1 estimation, we can not guarantee the estimation error is bounded when there are more than $m(A)$ outliers. However, we can guarantee it is bounded in the following iterations.

The basic idea underlying behind Theorem 2.7 is based on the following intuition: when there are $\|e\|_0 > m(A)$ outliers, if the smallest $\|e\|_0 - m(A)$ of them are moderate, we can treat them as very noisy inliers, so the number of outliers reduces to $m(A)$. Then according to Theorem 2.6, we can use a large α to safely remove the very large outliers.

2.4 Empirical Studies

For relatively small size matrix A, we can compute $m(A)$ using the algorithm provided in [75], this gives us an opportunity to further study the behavior of $m(A)$ w.r.t. m (the number of rows of A). On the other hand, although we provided some theoretical guarantees/bounds for AROSI, they often involve $c_q(A)$, which itself is hard to compute. In this section, we empirically study the performance of AROSI (including the reprojection step unless noted) as well as the following state-of-the-art methods, where the complexity analysis is presented for $m > n$.

1. ℓ_1 estimator [13]: $x_{\ell_1} = \arg\min_x \|y - Ax\|_1$. We also add a reprojection step for comparison. The complexity in practice is $O(m^3)$ [77].

2. Second-Order Cone Programming (SOCP) [10], which is a direct application (via the Projection Approach) of ℓ_1 minimization sparse recovery [19]–[21] to model (2.3). There is a reprojection step in the end. The complexity of this method is $O(m^3)$ [78].

3. Ideal solution where we know e exactly: $x_{Ideal} = \arg\min_x \|y - e - Ax\|_2$.

4. Oracle solution [10] where we know the support of e exactly: $x_{Oracle} = \arg\min_x \|y_{\boldsymbol{S}} - A_{\boldsymbol{S}}x\|_2$, where $\boldsymbol{S} := \{i : e_i = 0\}$ is the index set of all the inliers.

5. Bayesian Sparse Robust Regression (BSRR) [15], which is a direct application (via the Projection Approach) of Sparse Bayesian Learning to model (2.3). The complexity of each iteration is $O(m^3)$. We add a reprojection step in the end.

6. Generalized M-estimators with Bisquare weighting function [7], [79]–[82]. It is solved via Iteratively Reweighted Least Squares (IRLS), and the complexity of each iteration is $O(mn^2)$. We set its tuning constant $c = 3$ to generate better results than the default value.

7. ℓ_1 regularization algorithm [14], [22], which solves $\min_{x,e} \|y - Ax - e\|_2^2 + \lambda\|e\|_1$, where the parameter λ is set as $\frac{\sigma\sqrt{2log(m)}}{3}$ according to [22]. It can be solved using the approach described in [3], where the complexity is $O(m^3)$ per iteration [3]. We add a reprojection step in the end.

8. Greedy Algorithm for Robust Denoising (GARD) [3], which aims to minimize the number of outliers via OMP by restricting the selection over columns of $I_{m\times m}$:

   $\min_{x,e} \|e\|_0 \ s.t. \ \|y - [A \ I_{m\times m}][\begin{smallmatrix} x \\ e \end{smallmatrix}]\|_2^2 \leq \epsilon^2$.

   The total complexity is $O(\frac{K^3}{2} + (m + 3K)n^2 + 3Kmn)$, where K is the total number of iterations. We add a reprojection step in the end.

9. Thresholding-based Iterative Procedure for Outlier Detection (Θ-IPOD) [11], which iterates between least squares regression and hard thresholding. We initialize it by ℓ_1 estimation, and set the threshold to 5σ. The algorithm's pre-computation costs $O(mn^2)$, and each iteration costs $O(mn)$. We add a reprojection step in the end.

For AROSI, we fix α as 5σ throughout the experiments unless otherwise noted. In the reprojection step of BSRR, SOCP, AROSI, Θ-IPOD, GARD and the ℓ_1 regularization method, the threshold is tuned individually from $\{p\sigma : p = 1, 2, 3, 4, 5\}$ for each method.

For our experimental setup, below are the general steps:

1. Choose a fraction ρ of grossly corrupted entries and define the number of corrupted entries as k = round($\rho \cdot m$);

2. Generate an m by n standard Gaussian matrix A.

3. Generate $x \in R^n$ with i.i.d. $\mathcal{N}(0, \sigma_x^2)$ entries. Compute Ax.

4. Select k locations uniformly at random and add corruptions to these locations.

5. Generate the vector $\eta = (\eta_1, \ldots, \eta_m)$ of smaller errors with η_i i.i.d. $\mathcal{N}(0, \sigma^2)$, and add η to the outcome of the previous step. Obtain y.

6. Estimate x using different methods.

We first set $m = 512$, $\sigma_x = 1$, and $\sigma = median(|Ax|)/16$ as in [10]. The corruption values are drawn from $0.5\times\mathcal{N}(12\sigma, (4\sigma)^2)+0.5\times\mathcal{N}(-12\sigma, (4\sigma)^2)$. For each $n \in \{256, 128, 64\}$, we repeat Step 2 - Step 6 fifty times for each corruption rate. We denote this setting as experimental setup A.

Next, we use the experimental setup in [3] (denoted as experimental setup B), where $m = 600$, $\sigma_x = 5$, $\sigma = 1$, and the rows of matrix A are obtained by uniformly sampling an n-dimensional hypercube centered around the origin, i.e., $A_{ij} \sim U(-1, 1)$. The corruption values are drawn from $\{-25, 25\}$ with equal chance. For each $n \in \{170, 100, 50\}$, we repeat Step 2 - Step 6 fifty times for each corruption rate.

For evaluation, each estimate is compared with ground truth x. We measure its Relative ℓ_2-Error [83]: $\|\hat{x} - x\|_2/\|x\|_2$. We also compute the distance between the supports of e and $\hat{e}$. Denoting the two supports as $\boldsymbol{E}$ and $\hat{\boldsymbol{E}}$, $\hat{\boldsymbol{E}}$ is estimated by thresholding $|\hat{e}|$ or $|y - A\hat{x}|$ with $p\sigma$, where p is tuned individually for each method. The distance is defined as in [47]: $\text{dist}\left(\hat{\boldsymbol{E}}, \boldsymbol{E}\right) = \frac{\max\{|\hat{\boldsymbol{E}}|,|\boldsymbol{E}|\}-|\hat{\boldsymbol{E}}\cap\boldsymbol{E}|}{\max\{|\hat{\boldsymbol{E}}|,|\boldsymbol{E}|\}}$. We denote the average of $\text{dist}\left(\hat{\boldsymbol{E}}, \boldsymbol{E}\right)$ over Monte Carlo runs as the Probability of Error in Support (PES) [83].

2.4.1 Slowly Decreasing Property of $m(A)$

AROSI uses the submatrix of A to estimate x in every iteration. Besides $m(A)$ itself, it is also important to know how $m(A)$ changes as m (the number of rows of A) decreases. For relatively small size matrix A, we gradually delete its rows from the bottom to obtain the submatrices of A (denoted $A_{\{1:m-k\}}, k = 0, 1, \ldots, m - n$) in the experiment, and calculate the corresponding $m(A_{\{1:m-k\}})$.

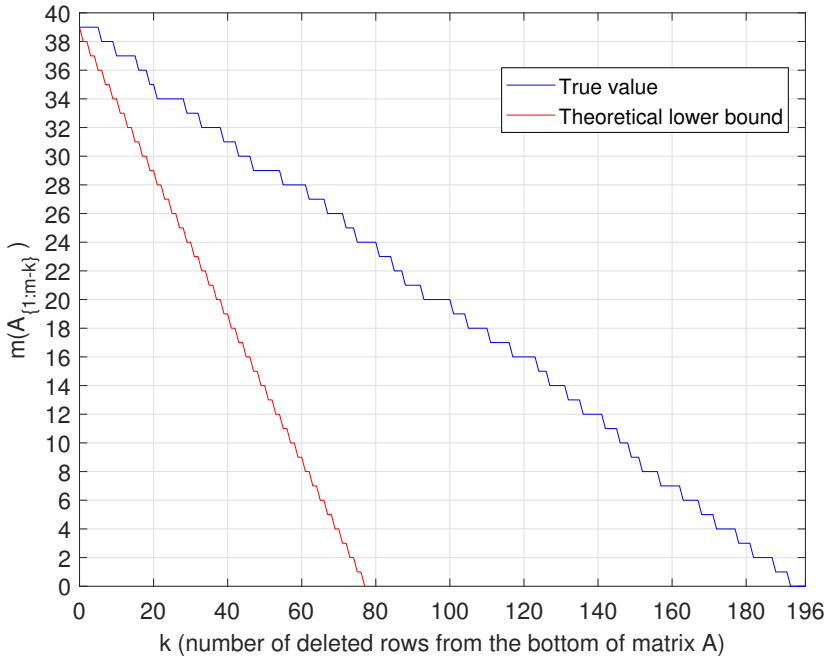

Figure 2.1. $m(A_{\{1:m-k\}})$ w.r.t. k for a 200 by 4 standard Gaussian matrix A.

We experiment with a randomly generated 200 by 4 standard Gaussian matrix. Fig. 2.1 shows the calculated $m(A_{\{1:m-k\}})$ w.r.t. k as well as the theoretical lower bound (red line) provided by Theorem 2.2. For the full matrix, $m(A)$=39. When deleting 10 rows, $m(A_{\{1:m-10\}})$=37, only decreasing by 2 from the original $m(A)$. When deleting 30 rows, we find a decrease of 6 from the original $m(A)$. Even when half of the total rows (i.e., 100) are deleted, $m(A_{\{1:m-100\}})$=20, a decrease of 19 from the original $m(A)$. It can be seen that $m(A_{\{1:m-k\}})$ decreases very slowly w.r.t. the number of deleted rows k, and it is above our theoretical lower bound. Note that the theoretical lower bound provided by Theorem 2.2 is for deleting arbitrary k rows of an arbitrary full column rank matrix A, while the experiment here only deletes k rows from the bottom of a specific standard Gaussian matrix A.

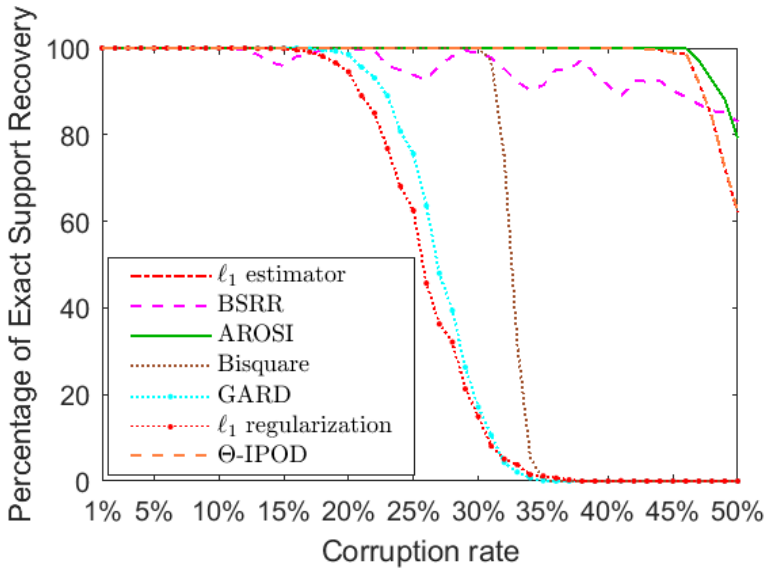

Figure 2.2. Percentage of exact support recovery vs. corruption rate.

2.4.2 Exact Recovery Test

In this subsection, we empirically verify the exact recovery performance of AROSI when only sparse outliers are present, i.e., $y = Ax + e$. Recall that in the reprojection step, exact recovery of the support of e will suffice for the exact recovery of both x and e, as long as $[A\ (I_{\boldsymbol{E}})^T]$ is full column rank.

We use the same experimental setup as the Support Recovery Test in [3]. This is under experimental setup B with n =100, except that there is no dense inlier noise. Fig. 2.2 shows the percentage of exact support recovery for each corruption rate (over 1000 trials) for each method. The support of BSRR, ℓ_1 estimator, Bisquare, AROSI, Θ-IPOD and the ℓ_1 regularization method (all without reprojection) is estimated by thresholding $|e|$ or $|y - Ax|$ with a small numerical constant 1×10^{-4}. Over 1000 trials, Bisquare keeps fully exact support recovery up to 29% corruption rate. For BSRR, ℓ_1 regularization method, GARD, ℓ_1 estimator, Θ-IPOD, and AROSI, it is up to 11% , 12% , 16% , 42% , 42% , and 44% , respectively. Θ-IPOD performs similarly to its initialization (ℓ_1 estimation), while AROSI demonstrates an improvement over ℓ_1 estimation.

When the corruption rates are 43% and 44%, there are cases where AROSI has exact

support recovery while the ℓ_1 estimator does not. From the quoted theorem in Appendix C, we know it must be the case that $\|e\|_0 > m(A)$. Since we also use the same ℓ_1 estimation in our first iteration, we do not have a perfect initialization. However, at the end of the iterations, we are able to identify and remove some outliers through the index set S_k. The number of remaining outliers is very likely less than $m(A_{S_k})$, thus we get the exact solution. This shows the advantage of AROSI over the ℓ_1 estimator.

2.4.3 Both Dense Noise and Sparse Outliers Present

In this subsection, we test and compare the performance of each method in the noisy case under experimental setup A. Fig. 2.3 shows the average Relative ℓ_2-Error and the PES from 50 samples vs. corruption rate. In general, AROSI has similar performance to BSRR and outperforms other methods. We can see that the reprojection step alone does help improve the performance of the ℓ_1 estimator. However, AROSI performs even better, which verifies that the advantage of AROSI over the ℓ_1 estimator is non-trivial. We can also see that, under the same corruption rate, increasing the signal dimension n makes the recovery harder for all methods, as the number of unknowns gets larger.

We have also tested on several non-Gaussian regression matrices, which can be found in the supplemental material. The relative performance of each method is almost unchanged, except some degradation of the relative performance of the ℓ_1 regularization method under some regression matrices.

2.4.4 Phase Transition Curves

We measure the Phase Transition Curves of each method under experimental setup A. For each dimension of x and each method, we test each outlier fraction and find the maximum fraction where the probability of successful recovery (Relative ℓ_2-Error less than 1.3× that of

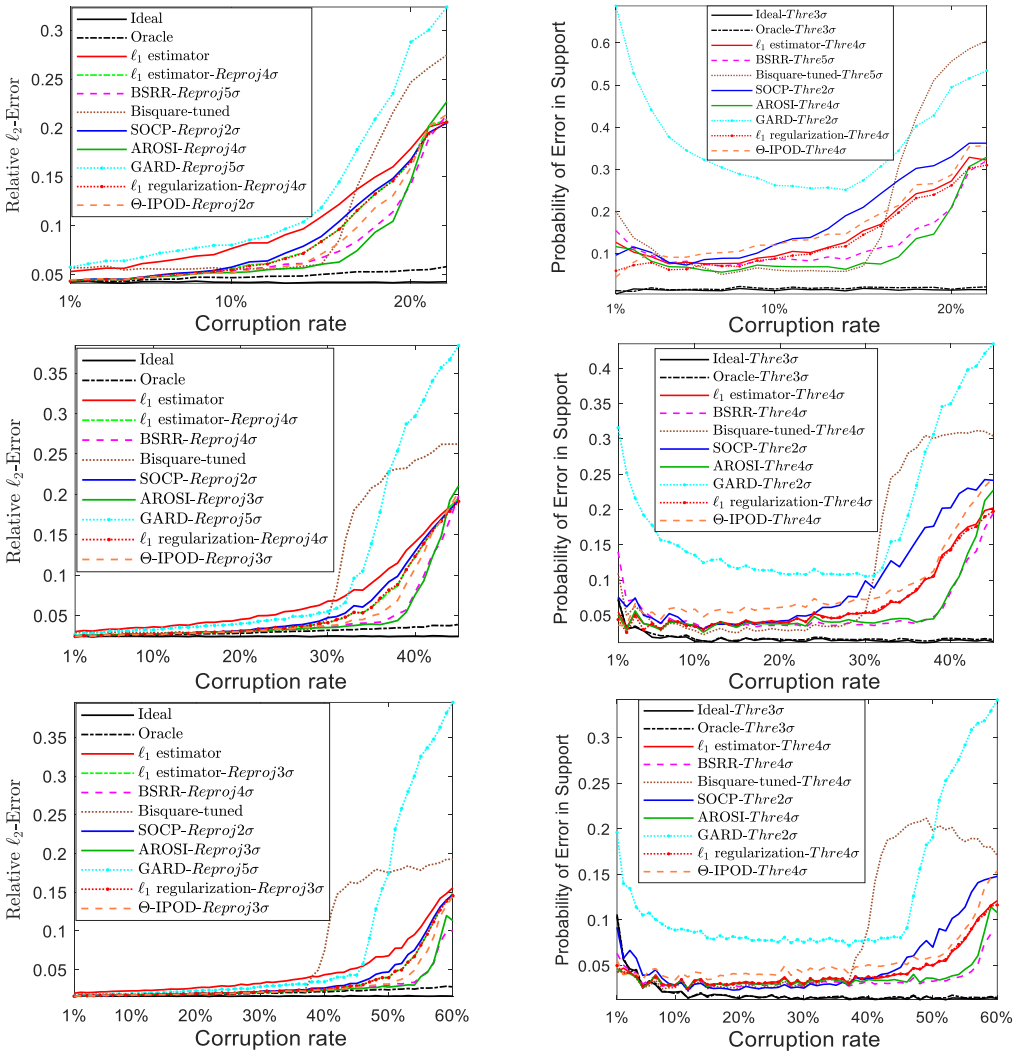

Figure 2.3. Average relative ℓ_2-error (left) and PES (right) vs. corruption rate with different n (upper: 256; middle: 128; bottom: 64).

Oracle) remains greater than 0.5. Fig. 2.4 shows the Phase Transition Curves of each method. AROSI outperforms all the other methods.

2.4.5 Different Magnitude of Corruptions

In this subsection, we use experimental setup A but with corruption values drawn from $\mathcal{N}(0,(\kappa\sigma)^2)$ instead (recall that $\sigma = median(|Ax|)/16$). We gradually increase the

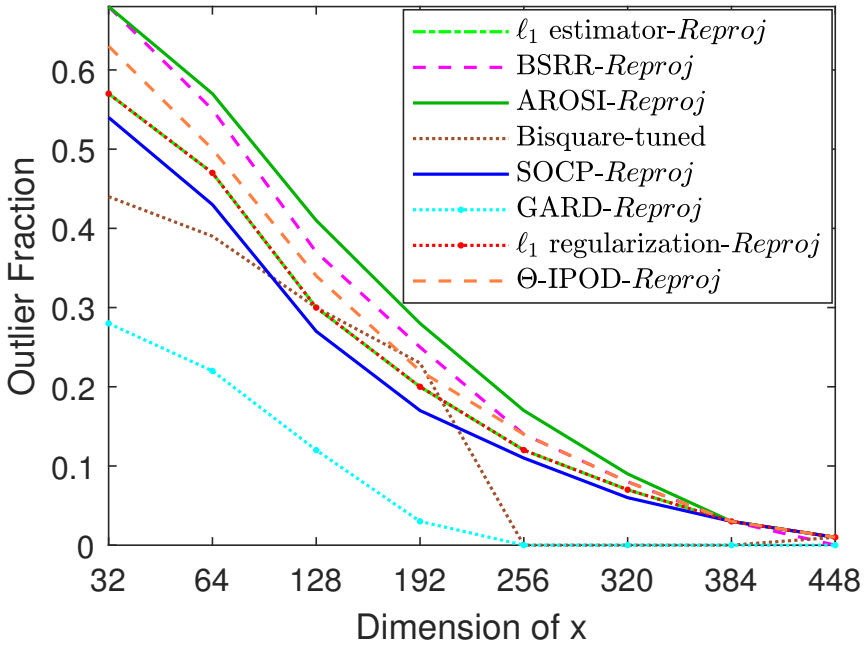

Figure 2.4. Phase transition curves.

magnitude of corruptions (by increasing κ) to see how each method behaves. Fig. 2.5 shows the average Relative ℓ_2-Error on 50 samples vs. corruption rate for different scales ($\kappa\sigma$) of corruptions. We can see that, when the magnitude of corruptions is small (e.g., $\kappa = 4$), even the least squares works well, and all the robust linear regression methods have very nominal differences and are slightly better than the least squares (we note that the performance of AROSI can be slightly improved if we set α larger). As κ is increased further, the robust linear regression methods begin to show their benefits. We note that when κ increases from 4 to 16, the performances of the ℓ_1 estimator (with or without the reprojection step), Bisquare, SOCP, and the ℓ_1 regularization method degrade. In contrast, BSRR and AROSI are quite resistant to the larger magnitudes of corruptions.

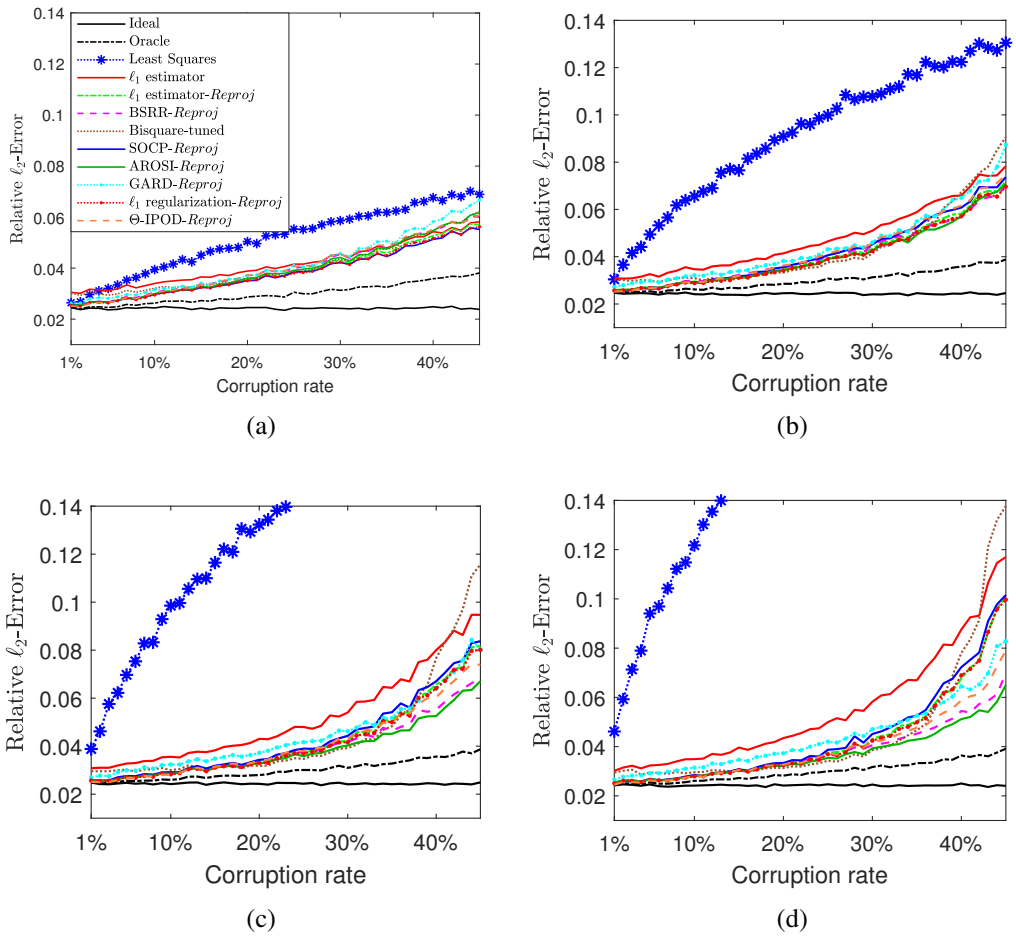

Figure 2.5. Average relative ℓ_2-error vs. corruption rate for different scales ($\kappa\sigma$) of Gaussian corruptions: a) κ=4; b) κ=8; c) κ=12; d) κ=16.

2.4.6 Sensitivity to Parameter α of AROSI

SOCP, GARD, ℓ_1 regularization method, Θ -IPOD, AROSI, and the initialization of BSRR all need the knowledge of inlier noise level. In the previous experiments, we assume we know the standard deviation σ of the inlier noise, and set $\alpha = 5\sigma$ for AROSI. However, in practice, the estimated σ may be slightly greater or less than the true σ, which is equivalent to setting α slightly greater or less than 5σ. We test AROSI with α varying from 2σ to 8σ. In the reprojection step of AROSI and the ℓ_1 estimator, we fix $p = 5$.

Fig. 2.6 shows the average Relative ℓ_2-Error on 50 samples vs. corruption rate for ℓ_1 estimation (with or without the reprojection step) and AROSI with different α, under experimental setup A with $n = 128$.

When the corruption rate is moderate (e.g., $\leq 35\%$ when n =128), we have two observations:

- AROSI often performs better than the ℓ_1 estimator even with different α (from 2σ to 8σ).
- With α ranging from 2σ to 8σ, AROSI has similar performance, which indicates the method is not very sensitive to small variations of α.

2.4.7 Run Time

In this subsection, we compare run times under experimental setup A. Fig. 2.7 shows the Average Run Time (seconds) on 100 samples vs. corruption rate with $n = 64$. We can see that AROSI is an order of magnitude faster than BSRR.

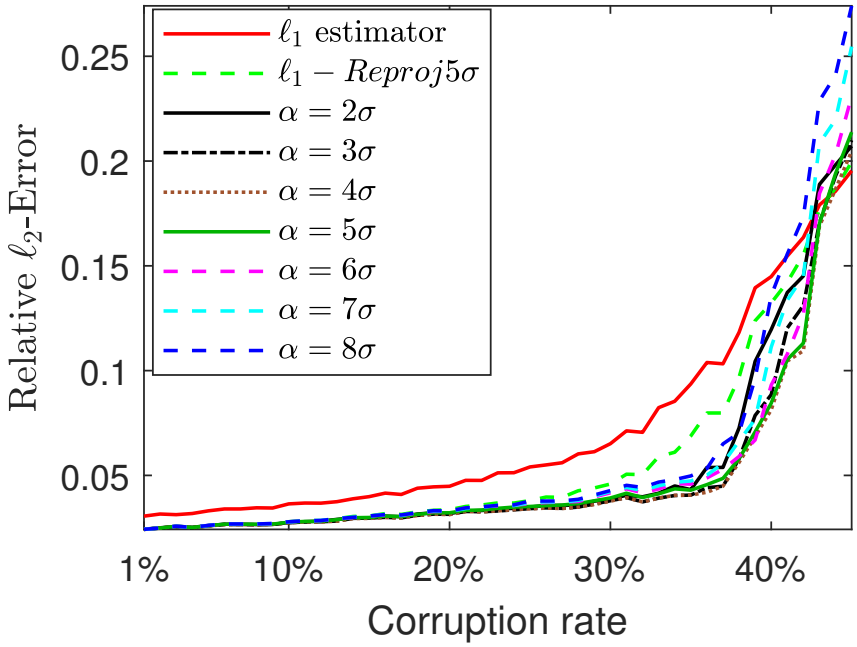

Figure 2.6. Average relative ℓ_2-error vs. corruption rate for ℓ_1 estimator and AROSI with different α. In the reprojection step of AROSI, p=5.

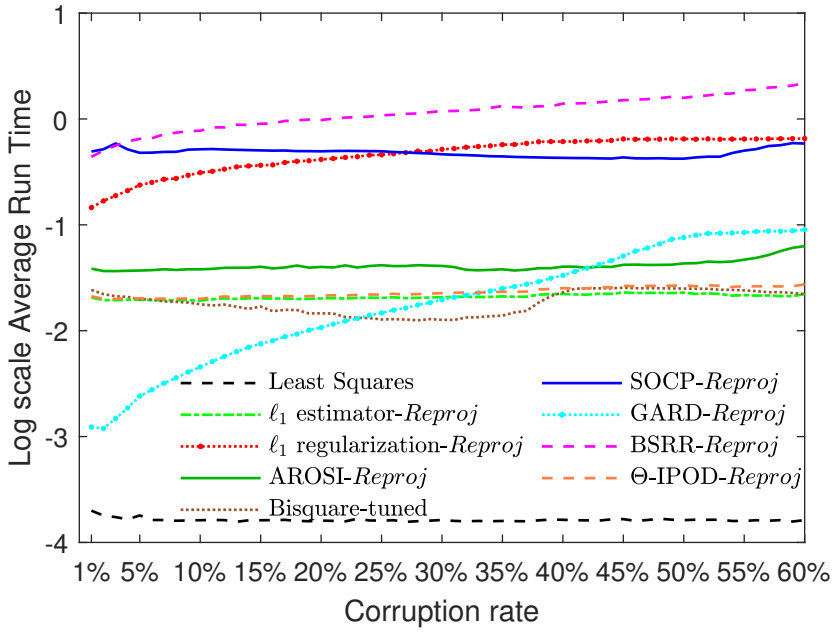

Figure 2.7. Log scale average run time vs. corruption rate.

2.4.8 Real Data

Finally, we compare the performance of each method on a real dataset, the Belgian Phone data, from the Belgian Statistical Survey (published by the Ministry of Economy). It contains large outliers as well as moderate outliers, and the swamping/masking effects could arise. There are 24 measurements. The response is the number of international phone calls (in millions), and the predictor is the year. It is known afterwards that observations 15-20 are large outliers and observations 14 and 21 are moderate outliers. For such a small size regression matrix A, using the algorithm provided in [75], we easily get $m(A) = 5$, which is unfortunately smaller than the number of the outliers.

To see the difference between each method more clearly, we do not perform the reprojection step, except for the Projection Approach methods, i.e., for BSRR and SOCP, the threshold is tuned to obtain the best result. The results are plotted in Fig. 2.8 (a). Most methods have very similar results on this data, and fit the inliers very well, except for the ℓ_1 estimator, SOCP, and the ℓ_1 regularization method. We can see that these three methods are biased by outliers, and the residual of the outlier observation 14 is very small (it is perfectly masked by large outliers), even smaller than many inlier observations, e.g., observations 1, 2, 22-24. So, even if we add a reprojection step for the ℓ_1 estimator and the ℓ_1 regularization method, the outlier observation 14 is hard to get rid of.

Though AROSI is equivalent to the ℓ_1 estimator at the beginning, it successfully eliminates the effect of outliers with a wide range of parameter α. Fig. 2.8 (b) shows the results of the ℓ_1 estimator and AROSI with integer α ranging from 3 to 180, as well as Θ-IPOD with the same threshold ranging from 3 to 180, all without the reprojection step. We can see that even with very different α, AROSI still fits the inliers very well, and is better than the ℓ_1 estimator. While Θ -IPOD (initialized by the ℓ_1 estimator) is sometimes severely biased by the outliers; it only works better than the ℓ_1 estimator when its threshold is set properly such that the outlier observations 15-20 are *all* identified at the beginning. When the threshold of

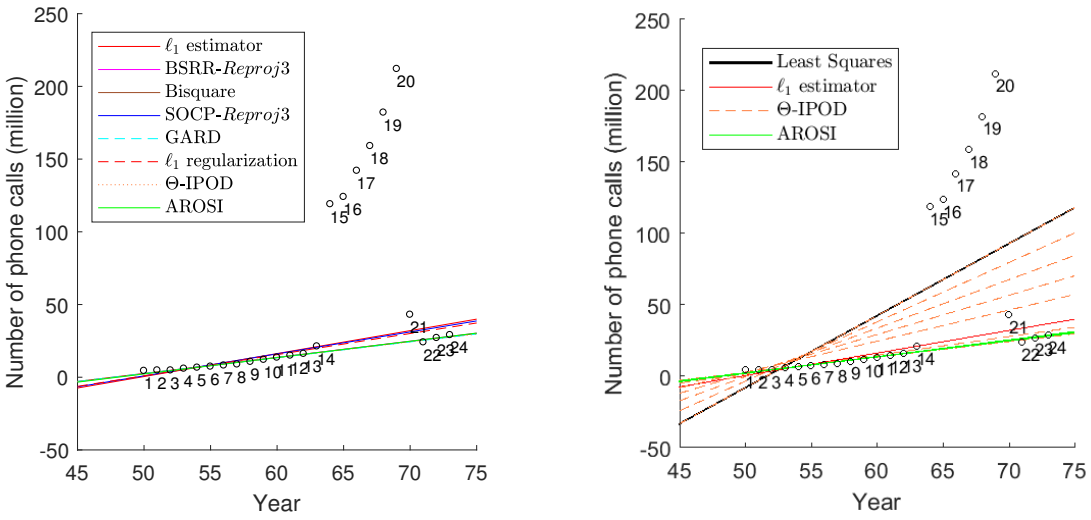

Figure 2.8. Number of phone calls (million) in the years 1950-1973 fitted by: (a) all methods (with tuned parameter). (b) Least Squares, ℓ_1 estimator, AROSI, and Θ-IPOD (the parameters of AROSI and Θ-IPOD both vary from 3 to 180).

Θ -IPOD is set larger than 137 (the outlier observations 19 and 20 can still be detected at the beginning), it will finally converge to the least squares solution. This demonstrates one important robustness property of AROSI over Θ -IPOD: the tolerance to unidentified outliers.

Table 2.1 documents the details of AROSI regarding its estimated outlier support set $\boldsymbol{S}_k^c$ and the corresponding $m(A_{\boldsymbol{S}_k})$ at the end of each iteration k under different α, as well as the estimated x upon convergence (without the reprojection step). AROSI converges in either 2 or 3 iterations (note that $\boldsymbol{S}_k^c = \boldsymbol{S}_{k-1}^c$ implies convergence). The least squares gives the solution $x_{LS} = (5.041, -260.059)$, which is severely biased by the outliers. The ℓ_1 estimator gives the solution $x_{\ell_1} = (1.580, -78.522)$. As $m(A) = 5$, which is smaller than the number of outliers (there are 6 large outliers and 2 moderate outliers), the performance of the ℓ_1 estimator is not guaranteed. However, we can see that with α ranging from 3 to 121, AROSI successfully identifies some outliers, and more importantly, the number of remaining outliers contained in $\boldsymbol{S}_{k-1}$ is less than the corresponding $m(A_{\boldsymbol{S}_{k-1}})$, which guarantees the performance

of AROSI in the last iteration k.

Table 2.1. Behavior of AROSI under different α

	1^{st} *iter.*		2^{nd} *iter.*		3^{rd} *iter.*	$\hat{x}$ (no reprojection)	
α	S_1^c	$m(A_{S_1})$	S_2^c	$m(A_{S_2})$	S_3^c	$\hat{x}(1)$	$\hat{x}(2)$
3	1,13, 15-24	2	14-21	2	14-21	1.125	-54.000
4	15-24	3	14-21	2	14-21	1.125	-54.000
5-7	15-24	3	15-21	2	15-21	1.115	-53.280
8	15-23	3	15-21	2	15-21	1.115	-53.280
9	15-22	2	15-21	2	15-21	1.115	-53.280
10	15-21	2	15-21	2	CNVG	1.115	-53.280
11-18	15-20	3	15-21	2	15-21	1.115	-53.280
19-96	15-20	3	15-20	3	CNVG	1.115	-53.280
97-99	16-20	3	15-20	3	15-20	1.115	-53.280
100	17-20	4	15-20	3	15-20	1.115	-53.280
101-104	17-20	4	16-20	3	16-20	1.133	-54.233
105-116	17-20	4	17-20	4	CNVG	1.151	-55.309
117-121	18-20	4	17-20	4	17-20	1.152	-55.349
122-131	18-20	4	18-20	4	CNVG	1.173	-56.609
132-137	19-20	5	18-20	4	18-20	1.173	-56.609
138-153	19-20	5	19-20	5	CNVG	1.173	-56.609
154-158	20	5	19-20	5	19-20	1.173	-56.609
159-180	20	5	20	5	CNVG	1.173	-56.609

2.5 Conclusion

We proposed a novel robust linear regression method AROSI based on ℓ_0 regularization. It assumes that outliers are sparse and result in large observation errors. Several properties of AROSI such as convergence, exact recovery or recovery error bound are derived.

Through extensive simulation studies and comparisons with state-of-the-art methods, we have shown that AROSI achieves the overall best quality of recovery (in terms of exact recovery, recovery error, outlier support recovery), and it runs much faster than the competing

methods like BSRR. Comparisons on a real dataset further demonstrate the robustness of AROSI and its advantage over the ℓ_1 estimator, Θ -IPOD, and the ℓ_1 regularization method.

Chapter 2, in part, is a reprint of the material as it appears in the paper: J. Liu, P. C. Cosman and B. D. Rao, "Robust Linear Regression via ℓ_0 Regularization," in *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 698-713, 1 Feb.1, 2018. The dissertation author was the primary investigator and author of this paper.

2.6 Appendices

2.6.1 Proof of Theorem 2.1

The proof is divided into the following three parts: a) monotonic decrease in the objective function prior to convergence, b) convergence in a finite number of steps, and c) local optimality of the cluster point.

a) Strictly decreasing behavior of $J(x^{(k)}, e^{(k)})$ before convergence

As defined earlier, $\boldsymbol{S}_k := \{i : e_i^{(k)} = 0\}$. We now denote its complementary set $\boldsymbol{S}_k^c := \{i : e_i^{(k)} \neq 0\}$. Define $J_{\boldsymbol{S}_k}(x, e) \triangleq \sum_{i \in \boldsymbol{S}_k}(|(y - Ax - e)_i| + \alpha I(e_i \neq 0))$ and $J_{\boldsymbol{S}_k^c}(x, e) \triangleq \sum_{i \in \boldsymbol{S}_k^c}(|(y - Ax - e)_i| + \alpha I(e_i \neq 0))$. So we have $J(x, e) = J_{\boldsymbol{S}_k}(x, e) + J_{\boldsymbol{S}_k^c}(x, e)$.

For any $i \in \boldsymbol{S}_k$, $e_i^{(k)} = 0$. Hence

$$\begin{aligned} J_{\boldsymbol{S}_k}(x, e^{(k)}) &= \sum_{i \in \boldsymbol{S}_k}(|(y - Ax - e^{(k)})_i| + \alpha I(e_i^{(k)} \neq 0)) \\ &= \sum_{i \in \boldsymbol{S}_k} |(y - Ax)_i| = \|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x\|_1. \end{aligned} \tag{2.12}$$

In Step 1, since $x^{(k+1)} \in \arg\min_x \|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x\|_1$, we have

$$J_{\boldsymbol{S}_k}(x^{(k+1)}, e^{(k)}) \leq J_{\boldsymbol{S}_k}(x^{(k)}, e^{(k)}), \tag{2.13}$$

where the equality holds if and only if

$$\|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x^{(k+1)}\|_1 = \|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x^{(k)}\|_1. \tag{2.14}$$

In Step 2, $J_{\boldsymbol{S}_k}(x^{(k+1)}, e) = \sum_{i \in \boldsymbol{S}_k} (|(y - Ax^{(k+1)})_i - e_i| + \alpha I(e_i \neq 0))$, and from (2.5) we know that $e_i^{(k+1)} \in \arg\min_{e_i} (|(y - Ax^{(k+1)})_i - e_i| + \alpha I(e_i \neq 0))$. Thus $J_{\boldsymbol{S}_k}(x^{(k+1)}, e^{(k+1)}) \leq J_{\boldsymbol{S}_k}(x^{(k+1)}, e^{(k)})$.

Utilizing (2.13) we have $J_{\boldsymbol{S}_k}(x^{(k+1)}, e^{(k+1)}) \leq J_{\boldsymbol{S}_k}(x^{(k)}, e^{(k)})$.

For any $i \in \boldsymbol{S}_k^c$, $e_i^{(k)} \neq 0$. From (2.5)-(2.6), we know that the upper bound for $J_{\boldsymbol{S}_k^c}(x^{(j)}, e^{(j)})$, $j = 1, 2, \ldots$ is $\alpha \times |\boldsymbol{S}_k^c|$, and $J_{\boldsymbol{S}_k^c}(x^{(k)}, e^{(k)})$ equals this upper bound. Hence $J_{\boldsymbol{S}_k^c}(x^{(k+1)}, e^{(k+1)}) \leq J_{\boldsymbol{S}_k^c}(x^{(k)}, e^{(k)})$.

In sum, we have $J(x^{(k+1)}, e^{(k+1)}) \leq J(x^{(k)}, e^{(k)})$. So the value of the objective function is non-increasing in each iteration. As the objective function is non-negative, it will always converge.

If $J(x^{(k+1)}, e^{(k+1)}) = J(x^{(k)}, e^{(k)})$, we must have equality to hold in (2.13), which implies $x^{(k+1)} = x^{(k)}$ according to (2.14) and Step 1. $x^{(k+1)} = x^{(k)}$ ensures $e^{(k+1)} = e^{(k)}$ and $S_{k+1} = \boldsymbol{S}_k$. Similarly $S_{k+1} = \boldsymbol{S}_k$ implies $x^{(k+2)} = x^{(k+1)}$, and further $e^{(k+2)} = e^{(k+1)}$ and $S_{k+2} = S_{k+1}$ and so on. So $(x^{(k)}, e^{(k)}) = (x^{(k+1)}, e^{(k+1)}) = (x^{(k+2)}, e^{(k+2)}) = \ldots$, which is a fixed point of AROSI.

Thus it follows that the objective function is strictly decreasing before convergence.

b) Convergence in a finite number of iterations

Now, we show that the objective function must converge in a finite number of iterations. As the number of different index sets $\boldsymbol{S}_k$ is finite (less than 2^m), it suffices to show that the same index set will not appear again before the objective function converges.

Note that the value of the objective function $J(x^{(k+1)}, e^{(k+1)})$ is determined by $x^{(k+1)}$ (as $e^{(k+1)}$ is also determined by $x^{(k+1)}$ according to Step 2).

We first show that the same index set can not reappear in nearby iterations before convergence. Suppose $\boldsymbol{S}_p = \boldsymbol{S}_{p-1}$, as $x^{(p)} = \arg\min_x \|y_{\boldsymbol{S}_{p-1}} - A_{\boldsymbol{S}_{p-1}} x\|_1 = \arg\min_x \|y_{\boldsymbol{S}_p} -$

$A_{\boldsymbol{S}_p}x\|_1$, and $x^{(p+1)} = \arg\min_x \|y_{\boldsymbol{S}_p} - A_{\boldsymbol{S}_p}x\|_1$, we must have $\|y_{\boldsymbol{S}_p} - A_{\boldsymbol{S}_p}x^{(p+1)}\|_1 = \|y_{\boldsymbol{S}_p} - A_{\boldsymbol{S}_p}x^{(p)}\|_1$, so the algorithm sets $x^{(p+1)} = x^{(p)}$ in Step 1. Then we must have convergence of the objective function.

Then it remains to show that the same index set can not reappear in non-consecutive iterations before convergence.

Before convergence, we have $J(x^{(1)}, e^{(1)}) > \ldots > J(x^{(p+1)}, e^{(p+1)}) > ... > J(x^{(r)}, e^{(r)}) > J(x^{(r+1)}, e^{(r+1)}) > \ldots$ The corresponding index sets in Step 1 of each iteration are $\boldsymbol{S}_0, \ldots, \boldsymbol{S}_p, \ldots, \boldsymbol{S}_{r-1}, \boldsymbol{S}_r, \ldots$ We only need to show that $\boldsymbol{S}_r \neq \boldsymbol{S}_p$ for any $r > p+1$. As proved earlier, any $x^{(r+1)} \in \arg\min_x \|y_{\boldsymbol{S}_r} - A_{\boldsymbol{S}_r}x\|_1$ ensures $J(x^{(r+1)}, e^{(r+1)}) \leq J(x^{(r)}, e^{(r)})$, see (2.13). Suppose $\boldsymbol{S}_r = \boldsymbol{S}_p$, then for any $x^{(p+1)} \in \arg\min_x \|y_{\boldsymbol{S}_p} - A_{\boldsymbol{S}_p}x\|_1$, $x^{(p+1)} \in \arg\min_x \|y_{\boldsymbol{S}_r} - A_{\boldsymbol{S}_r}x\|_1$, thus $J(x^{(p+1)}, e^{(p+1)}) \leq J(x^{(r)}, e^{(r)})$, which is contradictory to $J(x^{(p+1)}, e^{(p+1)}) > J(x^{(r)}, e^{(r)})$.

c) Convergence to a local optimum

We now prove that when $J(x, e)$ converges ($J(x^{(k+1)}, e^{(k+1)}) = J(x^{(k)}, e^{(k)})$), ($x^{(k)}, e^{(k)}$) is a local optimum. From (2.4), we have $J(x^{(k)}, e^{(k)}) = \|y - Ax^{(k)} - e^{(k)}\|_1 + \alpha\|e^{(k)}\|_0$.

Let $(\Delta x, \Delta e)$ be a small deformation vector around ($x^{(k)}, e^{(k)}$). Then

$$J(x^{(k)} + \Delta x, e^{(k)} + \Delta e) = \|y - A(x^{(k)} + \Delta x) - (e^{(k)} + \Delta e)\|_1 + \alpha\|e^{(k)} + \Delta e\|_0. \tag{2.15}$$

Next we will show that $J(x^{(k)} + \Delta x, e^{(k)} + \Delta e) \geq J(x^{(k)}, e^{(k)})$ as long as $\|\Delta e\|_1$ is small enough.

Notice that when $\|\Delta e\|_1$ is small enough, $\alpha I(e_i^{(k)} + \Delta e_i \neq 0) = \begin{cases} \alpha I(\Delta e_i \neq 0), e_i^{(k)} = 0 \\ \alpha I(e_i^{(k)} \neq 0), \text{otherwise} \end{cases}$.

So

$$\alpha\|e^{(k)} + \Delta e\|_0 = \alpha\|e^{(k)}\|_0 + \alpha \sum_{i \in \boldsymbol{S}_k} I(\Delta e_i \neq 0) = \alpha\|e^{(k)}\|_0 + \alpha\|\Delta e_{\boldsymbol{S}_k}\|_0. \tag{2.16}$$

As

$$
\begin{aligned}
&\|y - A(x^{(k)} + \Delta x) - (e^{(k)} + \Delta e)\|_1 \\
\geq &\|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k}(x^{(k)} + \Delta x) - (e^{(k)}_{\boldsymbol{S}_k} + \Delta e_{\boldsymbol{S}_k})\|_1 \\
\overset{(a)}{=} &\|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k}(x^{(k)} + \Delta x) - \Delta e_{\boldsymbol{S}_k}\|_1 \\
\geq &\|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k}(x^{(k)} + \Delta x)\|_1 - \|\Delta e_{\boldsymbol{S}_k}\|_1 \\
\overset{(b)}{\geq} &\|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x^{(k+1)}\|_1 - \|\Delta e_{\boldsymbol{S}_k}\|_1 \\
\overset{(c)}{=} &\|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x^{(k)}\|_1 - \|\Delta e_{\boldsymbol{S}_k}\|_1 \\
\overset{(d)}{=} &\|y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k} x^{(k)} - e^{(k)}_{\boldsymbol{S}_k}\|_1 - \|\Delta e_{\boldsymbol{S}_k}\|_1 \\
\overset{(e)}{=} &\|y - Ax^{(k)} - e^{(k)}\|_1 - \|\Delta e_{\boldsymbol{S}_k}\|_1,
\end{aligned} \tag{2.17}
$$

where step (a) and (d) follow from the fact that $e^{(k)}_{\boldsymbol{S}_k} = 0$, step (b) is from our Step 1, step (c) is from the convergence, see (2.14), and step (e) is from (2.5).

Substituting (2.16) and (2.17) in (2.15), we have

$$
\begin{aligned}
&J(x^{(k)} + \Delta x, e^{(k)} + \Delta e) \\
\geq &\|y - Ax^{(k)} - e^{(k)}\|_1 + \alpha\|e^{(k)}\|_0 + \alpha\|\Delta e_{\boldsymbol{S}_k}\|_0 - \|\Delta e_{\boldsymbol{S}_k}\|_1 \\
= &J(x^{(k)}, e^{(k)}) + \alpha\|\Delta e_{\boldsymbol{S}_k}\|_0 - \|\Delta e_{\boldsymbol{S}_k}\|_1.
\end{aligned}
$$

As long as $\|\Delta e\|_1$ is small enough (as $\|\Delta e_{\boldsymbol{S}_k}\|_1 \leq \|\Delta e\|_1$,then $\|\Delta e_{\boldsymbol{S}_k}\|_1$ is also small enough), we will have $\alpha\|\Delta e_{\boldsymbol{S}_k}\|_0 - \|\Delta e_{\boldsymbol{S}_k}\|_1 \geq 0$, and thus $J(x^{(k)} + \Delta x, e^{(k)} + \Delta e) \geq J(x^{(k)}, e^{(k)})$. So $(x^{(k)}, e^{(k)})$ is a local optimum of $J(x, e)$.

In the extreme case where $\boldsymbol{S}_k = \varnothing$, AROSI also sets $x^{(k+1)} = x^{(k)}$. Theorem 2.1 still holds.

2.6.2 Proof of Lemma 2.1

Let us first show that $A_{\boldsymbol{T}^c}$ must be full column rank for any $\boldsymbol{T} \subset \boldsymbol{M}$ with $|\boldsymbol{T}| = t \leq q$. As $m(A) \geq q \geq t$, from Proposition 2.2, $c_t(A) > \frac{1}{2}$. If $t = 0$, $A_{\boldsymbol{T}^c} = A$ is full column rank. If $t > 0$, suppose $A_{\boldsymbol{T}^c}$ is not full column rank. Then there exists $g \in R^n$ and $g \neq 0$, such that $A_{\boldsymbol{T}^c} g = 0$. Thus $\min_{\substack{g \in R^n \\ g \neq 0}} \frac{\sum_{i \in \boldsymbol{T}^c} |a_i^T g|}{\sum_{i \in \boldsymbol{M}} |a_i^T g|} = 0$. This contradicts $c_t(A) = \min_{\substack{\boldsymbol{T} \subset \boldsymbol{M} \\ |\boldsymbol{T}| = t}} \min_{\substack{g \in R^n \\ g \neq 0}} \frac{\sum_{i \in \boldsymbol{T}^c} |a_i^T g|}{\sum_{i \in \boldsymbol{M}} |a_i^T g|} > \frac{1}{2}$, so $A_{\boldsymbol{T}^c}$ must be full column rank.

The following proof is motivated by the proof of Theorem 3.4 in [84].

From Proposition 2.2, we must have $m > 2q$. Let $q' = q - \lceil 0.5t \rceil$. As $0 \leq t \leq q$, we have $0 \leq q' \leq q$, so $m > 2q \geq t + q'$. For any given index set $\boldsymbol{T} \subset \boldsymbol{M}$ with $|\boldsymbol{T}| = t$, and any index set $\boldsymbol{R} \subset \boldsymbol{T}^c$ with $|\boldsymbol{R}| = q'$, and any $g \in R^n$, $g \neq 0$, define index set $\boldsymbol{L} :=$ { indices of the largest $\lceil 0.5t \rceil$ entries of $|A_{\boldsymbol{T}} g|$ } , and index set $\boldsymbol{E} = \boldsymbol{R} \cup \boldsymbol{L}$. As $\boldsymbol{R} \subset \boldsymbol{T}^c$ and $\boldsymbol{L} \subset \boldsymbol{T}$, so $\boldsymbol{R} \cap \boldsymbol{L} = \varnothing$, $|\boldsymbol{E}| = |\boldsymbol{R}| + |\boldsymbol{L}| = q' + \lceil 0.5t \rceil = q$. We have $\boldsymbol{T} = (\boldsymbol{T} \backslash \boldsymbol{L}) \cup \boldsymbol{L}$, $\boldsymbol{T}^c = \boldsymbol{R} \cup (\boldsymbol{T}^c \backslash \boldsymbol{R})$, $\boldsymbol{M} = \boldsymbol{T} \cup \boldsymbol{T}^c = (\boldsymbol{T} \backslash \boldsymbol{L}) \cup \boldsymbol{L} \cup \boldsymbol{R} \cup (\boldsymbol{T}^c \backslash \boldsymbol{R}) = (\boldsymbol{T} \backslash \boldsymbol{L}) \cup \boldsymbol{E} \cup (\boldsymbol{T}^c \backslash \boldsymbol{R})$, $\boldsymbol{E}^c = (\boldsymbol{T} \backslash \boldsymbol{L}) \cup (\boldsymbol{T}^c \backslash \boldsymbol{R})$. As $(\boldsymbol{T} \backslash \boldsymbol{L}) \cap (\boldsymbol{T}^c \backslash \boldsymbol{R}) = \varnothing$, we have $\sum_{i \in \boldsymbol{T}^c \backslash \boldsymbol{R}} |a_i^T g| = \sum_{i \in \boldsymbol{E}^c} |a_i^T g| - \sum_{i \in \boldsymbol{T} \backslash \boldsymbol{L}} |a_i^T g|$.

Let us first consider the case $q > 0$.

As $m(A) \geq q = |\boldsymbol{E}|$, from Definition 2.1 we know $\frac{\sum_{i \in \boldsymbol{M} \backslash \boldsymbol{E}} |a_i^T g|}{\sum_{i \in \boldsymbol{M}} |a_i^T g|} \geq c_q(A)$ with $\frac{1}{2} < c_q(A) < 1$, this leads to $\sum_{i \in \boldsymbol{E}^c} |a_i^T g| \geq \frac{c_q(A)}{1 - c_q(A)} \sum_{i \in \boldsymbol{E}} |a_i^T g|$, where $\frac{c_q(A)}{1 - c_q(A)} > 1$.

So we have

$$
\begin{aligned}
&\sum_{i \in \boldsymbol{T}^c \backslash \boldsymbol{R}} |a_i^T g| \\
\geq &\frac{c_q(A)}{1 - c_q(A)} \sum_{i \in \boldsymbol{E}} |a_i^T g| - \sum_{i \in \boldsymbol{T} \backslash \boldsymbol{L}} |a_i^T g| \\
= &\frac{c_q(A)}{1 - c_q(A)} (\sum_{i \in \boldsymbol{R}} |a_i^T g| + \sum_{i \in \boldsymbol{L}} |a_i^T g|) - \sum_{i \in \boldsymbol{T} \backslash \boldsymbol{L}} |a_i^T g| \\
\geq &\frac{c_q(A)}{1 - c_q(A)} \sum_{i \in \boldsymbol{R}} |a_i^T g| + \sum_{i \in \boldsymbol{L}} |a_i^T g| - \sum_{i \in \boldsymbol{T} \backslash \boldsymbol{L}} |a_i^T g|. \qquad (2.18)
\end{aligned}
$$

As $|\boldsymbol{T}| = t$, $|\boldsymbol{L}| = \lceil 0.5t \rceil$, by the definition of index set $\boldsymbol{L}$, we must have

$$\sum_{i \in \boldsymbol{L}} |a_i^T g| - \sum_{i \in \boldsymbol{T} \setminus \boldsymbol{L}} |a_i^T g| \geq 0. \tag{2.19}$$

So from (2.18) and (2.19), we have

$$\sum_{i \in \boldsymbol{T}^c \setminus \boldsymbol{R}} |a_i^T g| \geq \frac{c_q(A)}{1 - c_q(A)} \sum_{i \in \boldsymbol{R}} |a_i^T g|. \tag{2.20}$$

As $\frac{1}{2} < c_q(A) < 1$, (2.20) implies $\frac{\sum_{i \in \boldsymbol{T}^c \setminus \boldsymbol{R}} |a_i^T g|}{\sum_{i \in \boldsymbol{T}^c} |a_i^T g|} \geq c_q(A)$.

So $c_{q-\lceil 0.5t \rceil}(A_{\boldsymbol{T}^c}) = \min_{|\boldsymbol{R}| = q - \lceil 0.5t \rceil} \min_{\substack{g \in R^n \\ g \neq 0}} \frac{\sum_{i \in \boldsymbol{T}^c \setminus \boldsymbol{R}} |a_i^T g|}{\sum_{i \in \boldsymbol{T}^c} |a_i^T g|} \geq c_q(A)$.

For the case $q = 0$, t must be zero. So $c_{q-\lceil 0.5t \rceil}(A_{\boldsymbol{T}^c}) = c_0(A_{\boldsymbol{T}^c}) = 1 = c_q(A)$.

In sum, we have $c_{q-\lceil 0.5t \rceil}(A_{\boldsymbol{T}^c}) \geq c_q(A)$. As $q - t \leq q - \lceil 0.5t \rceil$, from Proposition 2.1, we further have $c_{q-t}(A_{\boldsymbol{T}^c}) \geq c_{q-\lceil 0.5t \rceil}(A_{\boldsymbol{T}^c}) \geq c_q(A) > \frac{1}{2}$. From Definition 2.1, we must have $m(A_{\boldsymbol{T}^c}) \geq q - \lceil 0.5t \rceil \geq q - t$.

2.6.3 Theorem 2 of [72]

Let $x \in R^n, e \in R^m$, and set $y = Ax + e$, where $A \in R^{m \times n}$ is full column rank. Then, x is the unique solution of the problem $\min_{g \in R^n} \|y - Ag\|_1$ for any $\|e\|_0 \leq q$ if and only if $q \leq \mathcal{m}(A)$.

2.6.4 Lemma 2.3

The following Lemma facilitates the proof of Lemma 2.2 and Theorem 2.7, and is not introduced in the main text.

Let $y = Ax + e + \eta$ and $\boldsymbol{E} = supp(e)$ satisfying $|\boldsymbol{E}| = q \leq \mathcal{m}(A)$. Denote $r_{\ell_1} = y - Ax_{\ell_1}$, where $x_{\ell_1} = \arg\min_x \|y - Ax\|_1$. Then $\|(e + \eta) - r_{\ell_1}\|_1 \leq \frac{\sum_{i \in \boldsymbol{E}^c} |\eta_i|}{c_q(A) - 0.5} \leq \frac{\sqrt{m-q}\|\eta\|_2}{c_q(A) - 0.5}$.

Proof: Let us first quote an important Lemma, from Lemma 1 of [72]: Let $\boldsymbol{E} \subset \boldsymbol{M}$,

and $y, b^* \in R^m$, as well as $g^*, g \in R^n$ be arbitrary. Define $\boldsymbol{E}^c = \boldsymbol{M}\backslash\boldsymbol{E}$. If $|\boldsymbol{E}| = q \leq \mathcal{m}(A)$, then $\|y - Ag - b^*\|_1 - \|y - Ag^* - b^*\|_1 \geq (2c_q(A) - 1) \times \|(A(g - g^*)\|_1 - 2\sum_{i\in\boldsymbol{E}^c} |y_i - a_i^T g^* - b^*|$.

Setting $b^* = 0$, $g = x_{\ell_1}$, and $g^* = x$ in this Lemma, we have $0 \geq \|y - Ax_{\ell_1}\|_1 - \|y - Ax\|_1 \geq (2c_q(A) - 1) \times \|A(x_{\ell_1} - x)\|_1 - 2\sum_{i\in\boldsymbol{E}^c} |y_i - a_i^T x| = (2c_q(A) - 1) \times \|(A(x_{\ell_1} - x)\|_1 - 2\sum_{i\in\boldsymbol{E}^c} |\eta_i|$, where the first inequality is from the optimality of x_{ℓ_1}, and the last equality is from the fact that $y_i = a_i^T x + \eta_i$, $\forall\ i \in \boldsymbol{E}^c$.

As $q \leq \mathcal{m}(A)$, from Proposition 2.2, we have $c_q(A) > \frac{1}{2}$. So we have $\frac{\sum_{i\in\boldsymbol{E}^c} |\eta_i|}{c_q(A)-0.5} \geq \|A(x_{\ell_1} - x)\|_1 = \|(y - Ax) - (y - Ax_{\ell_1})\|_1 = \|(e + \eta) - r_{\ell_1}\|_1$.

Using the inequality of the norm, we have $\sum_{i\in E^c} |\eta_i| \leq \sqrt{|E^c|}\sqrt{\sum_{i\in E^c} |\eta_i|^2} \leq \sqrt{m - q}\|\eta\|_2$. So $\|(e + \eta) - r_{\ell_1}\|_1 \leq \frac{\sum_{i\in\boldsymbol{E}^c} |\eta_i|}{c_q(A)-0.5} \leq \frac{\sqrt{m-q}\|\eta\|_2}{c_q(A)-0.5}$.

2.6.5 Proof of Theorem 2.5

As $|\boldsymbol{E}| = q \leq \mathcal{m}(A)$ and $\boldsymbol{E}^c \subseteq \boldsymbol{S}_k$, we have $\boldsymbol{S}_k^c \subseteq \boldsymbol{E}$ and $|\boldsymbol{R}| = |\boldsymbol{E} \cap \boldsymbol{S}_k| = |\boldsymbol{E}| - |\boldsymbol{S}_k^c| \leq \mathcal{m}(A_{\boldsymbol{S}_k})$, and $A_{\boldsymbol{S}_k}$ is full column rank from Lemma 2.2, thus the condition of Corollary 2.1 is satisfied, $\|x^{(k+1)} - x\|_2$ is bounded as in (2.11).

As $\boldsymbol{S}_k \subseteq \boldsymbol{M}$, f*or* $\forall \boldsymbol{B'} \in \boldsymbol{Q'}_{|\boldsymbol{R}|}$ defined on $A_{\boldsymbol{S}_k}$, it has corresponding index set $\boldsymbol{B}$ defined on A, and $(A_{\boldsymbol{S}_k})_{\boldsymbol{B'}} = A_{\boldsymbol{B}}$. From the definition of $\boldsymbol{Q'}_{|\boldsymbol{R}|}$, there exists a set $\boldsymbol{L} \subseteq \boldsymbol{S}_k$ (both defined on A), $\boldsymbol{L} \supseteq \boldsymbol{B}$, $|\boldsymbol{L}| = |\boldsymbol{S}_k| - |\boldsymbol{R}|$, such that the following holds: $\sum_{i\in\boldsymbol{B}\cup(\boldsymbol{S}_k\backslash\boldsymbol{L})} |{a_i}^T v| \geq \sum_{i\in(\boldsymbol{L}\backslash\boldsymbol{B})} |{a_i}^T v|$, where v is any of the singular vectors corresponding to the smallest singular value of the $|\boldsymbol{B}| \times n$ submatrix $A_{\boldsymbol{B}}$ (of $A_{\boldsymbol{S}_k}$): $\|A_{\boldsymbol{B}} v\|_2 = \sigma_{\min}(A_{\boldsymbol{B}})\|v\|_2$. Then we have $\sum_{i\in\boldsymbol{B}\cup(\boldsymbol{M}\backslash\boldsymbol{L})} |{a_i}^T v| \geq \sum_{i\in\boldsymbol{B}\cup(\boldsymbol{S_k}\backslash\boldsymbol{L})} |{a_i}^T v| \geq \sum_{i\in(\boldsymbol{L}\backslash\boldsymbol{B})} |{a_i}^T v|$. As $|\boldsymbol{L}| = |\boldsymbol{S}_k| - |\boldsymbol{R}| = |\boldsymbol{S}_k| - \left(|\boldsymbol{E}| - \left|\boldsymbol{S}_k^c\right|\right) = |\boldsymbol{S}_k| + \left|\boldsymbol{S}_k^c\right| - |\boldsymbol{E}| = m - |\boldsymbol{E}|$, from Definition 2.2 we know that $\boldsymbol{B} \in \boldsymbol{Q}_{|\boldsymbol{E}|}$. So $\boldsymbol{Q'}_{|\boldsymbol{R}|}$ (defined in terms of $A_{\boldsymbol{S}_k}$) corresponds to a subset of $\boldsymbol{Q}_{|\boldsymbol{E}|}$ (defined in terms of A). Thus $\left\{(A_{\boldsymbol{S}_k})_{\boldsymbol{B'}} : \boldsymbol{B'} \in \boldsymbol{Q'}_{|\boldsymbol{R}|}\right\} \subseteq \{A_{\boldsymbol{B}} : \boldsymbol{B} \in \boldsymbol{Q}_{|\boldsymbol{E}|}\}$. So $\max_{\boldsymbol{B'}\in\boldsymbol{Q'}_{|\boldsymbol{R}|}} \frac{1}{\sigma_{\min}\left((A_{\boldsymbol{S}_k})_{\boldsymbol{B'}}\right)} \leq \max_{\boldsymbol{B}\in\boldsymbol{Q}_{|\boldsymbol{E}|}} \frac{1}{\sigma_{\min}(A_{\boldsymbol{B}})}$. Together with $\left\|\eta_{\boldsymbol{S}_k}\right\|_2 \leq \|\eta\|_2$, this shows that the bound in (2.11) is smaller than or equal to the bound in (2.10).

2.6.6 Proof of Lemma 2.2

As $\boldsymbol{S}_k \supseteq \boldsymbol{E}^c$, so $\boldsymbol{S}_k^c \subseteq \boldsymbol{E}$ and $|\boldsymbol{S}_k^c| \leq |\boldsymbol{E}| = q$. From Lemma 2.1 we know that $A_{\boldsymbol{S}_k}$ is full column rank, $\mathcal{m}(A_{\boldsymbol{S}_k}) \geq q - |\boldsymbol{S}_k^c|$ and $c_{(q-|\boldsymbol{S}_k^c|)}(A_{\boldsymbol{S}_k}) \geq c_q(A) > \frac{1}{2}$. So

$$c_{(q-|\boldsymbol{S}_k^c|)}(A_{\boldsymbol{S}_k}) - 0.5 \geq c_q(A) - 0.5 > 0. \tag{2.21}$$

As $\boldsymbol{S}_k^c \subseteq \boldsymbol{E}$, so $|supp(e_{\boldsymbol{S}_k})| = \|e_{\boldsymbol{S}_k}\|_0 = \|e\|_0 - |\boldsymbol{S}_k^c| = q - |\boldsymbol{S}_k^c| \leq \mathcal{m}(A_{\boldsymbol{S}_k})$. From Lemma 2.3 and (2.21), we have $\|(e+\eta)_{\boldsymbol{S}_k} - r_{\boldsymbol{S}_k}^{(k+1)}\|_1 \leq \frac{\sqrt{m-q}\|\eta\|_2}{c_{(q-|\boldsymbol{S}_k^c|)}(A_{\boldsymbol{S}_k})-0.5} \leq \frac{\sqrt{m-q}\|\eta\|_2}{c_q(A)-0.5}$.

For $\forall i \in \boldsymbol{E}^c \subseteq \boldsymbol{S}_k$, $e_i = 0$, $|r_i^{(k+1)}| - |\eta_i| \leq |\eta_i - r_i^{(k+1)}| = |(e+\eta)_i - r_i^{(k+1)}| \leq \|(e+\eta)_{\boldsymbol{S}_k} - r_{\boldsymbol{S}_k}^{(k+1)}\|_1$.

So $|r_i^{(k+1)}| \leq |\eta_i| + \|(e+\eta)_{\boldsymbol{S}_k} - r_{\boldsymbol{S}_k}^{(k+1)}\|_1 \leq \|\eta\|_\infty + \|(e+\eta)_{\boldsymbol{S}_k} - r_{\boldsymbol{S}_k}^{(k+1)}\|_1 \leq \|\eta\|_\infty + \frac{\sqrt{m-q}\|\eta\|_2}{c_q(A)-0.5}$.

2.6.7 Proof of Theorem 2.6

a) In Step 1 of the $(k+1)$ st (e.g., $k = 0, 1, \dots$) iteration, if $\boldsymbol{S}_k \supseteq \boldsymbol{E}^c$, from Lemma 2.2 we have $\forall i \in \boldsymbol{E}^c$, $|r_i^{(k+1)}| \leq \|\eta\|_\infty + C_1 < \alpha$, then $e_i^{(k+1)} = 0$ according to (2.5). Then $S_{k+1} := \{i : e_i^{(k+1)} = 0\} \supseteq \boldsymbol{E}^c$.

As $S_0 = \boldsymbol{M} \supseteq \boldsymbol{E}^c$, we will have $\boldsymbol{S}_k \supseteq \boldsymbol{E}^c$ for any $k \in Z_{\geq 0}$.

b) As $\boldsymbol{S}_k \supseteq \boldsymbol{E}^c$ for any $k \in Z_{\geq 0}$, from Lemma 2.2 we have $\|(e+\eta)_{\boldsymbol{S}_k} - r_{\boldsymbol{S}_k}^{(k+1)}\|_1 \leq \frac{\sqrt{m-q}\|\eta\|_2}{c_q(A)-0.5}$ for any $k \in Z_{\geq 0}$. From Lemma 2.1, we know $A_{\boldsymbol{E}^c}$ is full column rank and thus $\sigma_{\min}(A_{\boldsymbol{E}^c}) > 0$. As $\frac{\sqrt{m-q}\|\eta\|_2}{c_q(A)-0.5} \geq \|(e+\eta)_{\boldsymbol{S}_k} - r_{\boldsymbol{S}_k}^{(k+1)}\|_1 = \|(y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k}x) - (y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k}x^{(k+1)})\|_1 = \|A_{\boldsymbol{S}_k}(x - x^{(k+1)})\|_1$ $\geq \|A_{\boldsymbol{E}^c}(x - x^{(k+1)})\|_1 \geq \|A_{\boldsymbol{E}^c}(x - x^{(k+1)})\|_2 \geq \sigma_{\min}(A_{\boldsymbol{E}^c})\|x - x^{(k+1)}\|_2$, we have $\|x - x^{(k+1)}\|_2 \leq \frac{\sqrt{m-q}\|\eta\|_2}{\sigma_{\min}(A_{\boldsymbol{E}^c})\times(c_q(A)-0.5)}$ for any $k \in Z_{\geq 0}$.

For *any* $k \in \mathbb{Z}_{\geq 0}$,$\forall i \in \boldsymbol{P}\subseteq\boldsymbol{E}$, we have $|e_i| - |\eta_i| - \left|r_i^{(k+1)}\right| \leq \left|(e+\eta)_i\right| - \left|r_i^{(k+1)}\right| \leq \left|(e+\eta)_i - r_i^{(k+1)}\right| \leq \left\| \left(e+\eta-r^{(k+1)}\right)_{\boldsymbol{E}}\right\|_2 = \left\| (y-Ax)_{\boldsymbol{E}} - \left(y - Ax^{(k+1)}\right)_{\boldsymbol{E}}\right\|_2 = \left\| A_{\boldsymbol{E}}\left(x - x^{(k+1)}\right)\right\|_2$ $\leq \sigma_{\max}(A_{\boldsymbol{E}})\left\| x - x^{(k+1)}\right\|_2 \leq \frac{\sigma_{\max}(A_{\boldsymbol{E}})\sqrt{m-q}\ \|\ \eta\|_2}{\sigma_{\min}(A_{\boldsymbol{E}^c})\times\left(c_q(A)-0.5\right)} = C_3$, so $\left|r_i^{(k+1)}\right| \geq |e_i| - |\eta_i| - C_3 \geq$

$\min\{|e_i|:\ i \in \boldsymbol{P}\} - \|\eta\|_\infty - C_3 > \alpha$, then $e_i^{(k+1)} \neq 0$ according to (2.5). Then $\boldsymbol{P} \subseteq \boldsymbol{S}_{k+1}^{\boldsymbol{c}} := \{i : e_i^{(k+1)} \neq 0\}$ for *any* $k \in \mathbb{Z}_{\geq 0}$.

c) For any $k \in Z_{\geq 0}$, as $\boldsymbol{S}_k \supseteq \boldsymbol{E}^c$ and $|\boldsymbol{E}| = q \leq m(A)$, from Theorem 2.5 we know $\|x^{(k+1)} - x\|_2$ is bounded.

d) In Step 1 of the first iteration, as the condition of Lemma 2.2 is satisfied, we have $\|e + \eta - r^{(1)}\|_1 \leq C_1$. So $\forall i \in E$, we have $|e_i| - |\eta_i| - |r_i^{(1)}| \leq |(e+\eta)_i| - |r_i^{(1)}| \leq |(e+\eta)_i - r_i^{(1)}| \leq \|e + \eta - r^{(1)}\|_1 \leq C_1$, thus $|r_i^{(1)}| \geq |e_i| - |\eta_i| - C_1 \geq \min\{|e_i| : e_i \neq 0\} - \|\eta\|_\infty - C_1 \geq \min\{|e_i| : e_i \neq 0\} - \|\eta\|_\infty - C_2 > \alpha$. Then $e_i^{(1)} \neq 0$ according to (2.5). Then $\boldsymbol{E} \subseteq \boldsymbol{S}_1^c := \{i : e_i^{(1)} \neq 0\}$. As $\alpha > \|\eta\|_\infty + C_1$ guarantees $\boldsymbol{S}_1 \supseteq \boldsymbol{E}^c$, we have $\boldsymbol{S}_1 = \boldsymbol{E}^c$.

In Step 1 of the second iteration, as $\boldsymbol{S}_1 = \boldsymbol{E}^c$, from Lemma 2.2, we have $\|(e+\eta)_{\boldsymbol{E}^c} - r_{\boldsymbol{E}^c}^{(2)}\|_1 \leq \frac{\sqrt{m-q}\|\eta\|_2}{c_0(A)-0.5} = 2\sqrt{m-q}\|\eta\|_2$. As $\|(e+\eta)_{\boldsymbol{E}^c} - r_{\boldsymbol{E}^c}^{(2)}\|_1 = \|(y_{\boldsymbol{E}^c} - A_{\boldsymbol{E}^c}x) - (y_{\boldsymbol{E}^c} - A_{\boldsymbol{E}^c}x^{(2)})\|_1 = \|A_{\boldsymbol{E}^c}(x - x^{(2)})\|_1 \geq \|A_{\boldsymbol{E}^c}(x - x^{(2)})\|_2 \geq \sigma_{\min}(A_{\boldsymbol{E}^c})\|x - x^{(2)}\|_2$, we have $\|x - x^{(2)}\|_2 \leq 2\sqrt{m-q}\|\eta\|_2/\sigma_{\min}(A_{\boldsymbol{E}^c})$.

For $\forall i \in \boldsymbol{E}$, we have $|e_i| - |\eta_i| - |r_i^{(2)}| \leq |(e+\eta)_i| - |r_i^{(2)}| \leq |(e+\eta)_i - r_i^{(2)}| \leq \|(e+\eta-r^{(2)})_{\boldsymbol{E}}\|_2 = \|(y - Ax)_{\boldsymbol{E}} - (y - Ax^{(2)})_{\boldsymbol{E}}\|_2 = \|A_{\boldsymbol{E}}x - x^{(2)})\|_2 \leq \sigma_{\max}(A_{\boldsymbol{E}})\|x - x^{(2)}\|_2 \leq \sigma_{\max}(A_{\boldsymbol{E}}) \times \frac{2\sqrt{m-q}\|\eta\|_2}{\sigma_{\min}(A_{\boldsymbol{E}^c})} \leq C_2$, thus $|r_i^{(2)}| \geq |e_i| - |\eta_i| - C_2 \geq \min\{|e_i| : e_i \neq 0\} - \|\eta\|_\infty - C_2 > \alpha$.

Then $e_i^{(2)} \neq 0$ according to (2.5). Then $\boldsymbol{E} \subseteq \boldsymbol{S}_2^c := \{i : e_i^{(2)} \neq 0\}$. As $\alpha > \|\eta\|_\infty + C_1$ guarantees $\boldsymbol{S}_2 \supseteq \boldsymbol{E}^c$, we must have $\boldsymbol{S}_2 = \boldsymbol{E}^c$.

Finally, $\boldsymbol{S}_2 = \boldsymbol{E}^c = \boldsymbol{S}_1$ implies $x^{(3)} = x^{(2)}$, and further $\boldsymbol{S}_3 = \boldsymbol{S}_2 = \boldsymbol{E}^c$. So AROSI converges in 3 iterations and recovers the support of outliers exactly.

e) In the reprojection step, with a threshold in the range of $(\|\eta\|_\infty + C_1, \min\{|e_i|:\ e_i \neq 0\} - \|\eta\|_\infty - C_2)$, we have $\hat{\boldsymbol{E}} = \boldsymbol{E}$ and $\hat{z} = \arg\min_z \left\| y - [A\ (I_{\boldsymbol{E}})^T]z \right\|_2$, $\hat{x} = \hat{z}_{\{1,\ldots,n\}}$. As $A_{\boldsymbol{E}^c}$ is full column rank, $[A\ (I_{\boldsymbol{E}})^T]$ must also be full column rank (by inspecting the matrix structure). Actually, one can verify that the above $\hat{x}$ is also the unique solution of $\min_x \|\ y_{\boldsymbol{E}^c} - A_{\boldsymbol{E}^c}x\|_2$, so $\hat{x} = A_{\boldsymbol{E}^c}{}^\dagger y_{\boldsymbol{E}^c} = A_{\boldsymbol{E}^c}{}^\dagger (A_{\boldsymbol{E}^c}x + \eta_{\boldsymbol{E}^c}) = x + A_{\boldsymbol{E}^c}{}^\dagger \eta_{\boldsymbol{E}^c}$. We have $\|\ \hat{x} - x\|_2 = \left\| A_{\boldsymbol{E}^c}{}^\dagger \eta_{\boldsymbol{E}^c} \right\|_2 \leq \left\| A_{\boldsymbol{E}^c}{}^\dagger \right\|_2 \|\ \eta_{\boldsymbol{E}^c}\|_2 = \frac{\|\ \eta_{\boldsymbol{E}^c}\|_2}{\sigma_{\min}(\ A_{\boldsymbol{E}^c})}$.

Next, we want to show that the bound here is better than the bound in (2.10). Let v_1 be any of the singular vectors corresponding to the smallest singular value of the $A_{\boldsymbol{E}^c}$. Since $A_{\boldsymbol{E}^c}$ is full column rank, we have $\|A_{\boldsymbol{E}^c}v_1\|_2 = \sigma_{\min}(A_{\boldsymbol{E}^c})\|v_1\|_2 > 0$.

In Definition 2.2, we let the set $\boldsymbol{L} = \boldsymbol{E}^c$, and let the set $\boldsymbol{B}$ be a subset of $\boldsymbol{E}^c$ that corresponds to the $m - q - \mathcal{m}(A)$ (according to Proposition 2.2 it must be positive) smallest entries of $|A_{\boldsymbol{E}^c}v_1|$. Then $|\boldsymbol{B} \bigcup \boldsymbol{L}^c| = m - \mathcal{m}(A)$, and $|\boldsymbol{L}\backslash\boldsymbol{B}| = \mathcal{m}(A)$. Since $c_{\mathcal{m}(A)}(A) > 0.5$, we must have (2.9) holds. So the above set $\boldsymbol{B}$ is a possibly extreme set with q, i.e., $\boldsymbol{B} \in Q_{|\boldsymbol{E}|}$. Since $\sigma_{\min}(A_{\boldsymbol{B}})\|v_1\|_2 \leq \|A_{\boldsymbol{B}}v_1\|_2 \leq \|A_{\boldsymbol{E}^c}v_1\|_2 = \sigma_{\min}(A_{\boldsymbol{E}^c})\|v_1\|_2$ (where the second inequality becomes a strict inequality as long as $\mathcal{m}(A) > 0$), we have $\sigma_{\min}(A_{\boldsymbol{B}}) \leq \sigma_{\min}(A_{\boldsymbol{E}^c})$. Then it follows that the bound in (2.10) is larger or equal to $\|\eta_{\boldsymbol{E}^c}\|_2/\sigma_{\min}(A_{\boldsymbol{E}^c})$, and is strictly larger when $\mathcal{m}(A) > 0$.

2.6.8 Proof of Theorem 2.7

a-b) By definition, we have $|\boldsymbol{P}| = t \leq \mathcal{m}(A)$, and $\boldsymbol{P} \subseteq \boldsymbol{G} \subseteq \boldsymbol{E}$. We can view e_i indexed by $\boldsymbol{E}\backslash\boldsymbol{G}$ (can be an empty set) as part of the noise, i.e., we define the new noise and corruptions as $\eta_i^{'} = \begin{cases} e_i + \eta_i, \ i \in \boldsymbol{E}\backslash\boldsymbol{G} \\ \eta_i, \ otherwise \end{cases}$, $e_i^{'} = \begin{cases} e_i, \ i \in \boldsymbol{G} \\ 0, \ otherwise \end{cases}$, then $y = Ax + e^{'} + \eta^{'}$ with $\|e^{'}\|_0 = |\boldsymbol{G}| = \mathcal{m}(A) = q_1$.

In Step 1 of the first iteration, we have $\|(e+\eta)-r^{(1)}\|_1 = \|(e^{'}+\eta^{'})-r^{(1)}\|_1 \leq \frac{\sum_{i\in\boldsymbol{G}^c}|\eta_i^{'}|}{c_{q_1}(A)-0.5}$ from Lemma 2.3. So $\|(e+\eta)-r^{(1)}\|_1 \leq \frac{\sum_{i\in\boldsymbol{G}^c}|\eta_i^{'}|}{c_{q_1}(A)-0.5} \leq \frac{\sum_{i\in\boldsymbol{G}^c}|\eta_i+e_i|}{c_{q_1}(A)-0.5} \leq \frac{\sum_{i\in\boldsymbol{G}^c}|\eta_i|+\sum_{i\in\boldsymbol{G}^c}|e_i|}{c_{q_1}(A)-0.5} = \frac{\sum_{i\in\boldsymbol{G}^c}|\eta_i|+\sum_{i\in E\backslash\boldsymbol{G}}|e_i|}{c_{q_1}(A)-0.5} \leq \frac{\sqrt{m-q_1}\|\eta\|_2+\sum_{i\in E\backslash\boldsymbol{G}}|e_i|}{c_{q_1}(A)-0.5}$.

For $\forall i \in \boldsymbol{P}$, we have $|e_i|-|\eta_i|-|r_i^{(1)}| \leq |(e+\eta)_i|-|r_i^{(1)}| \leq |(e+\eta)_i - r_i^{(1)}| \leq \|e+\eta-r^{(1)}\|_1 \leq \frac{\sqrt{m-q_1}\|\eta\|_2+\sum_{i\in\boldsymbol{E}\backslash\boldsymbol{G}}|e_i|}{c_{q_1}(A)-0.5} \leq w_2$, thus $|r_i^{(1)}| \geq |e_i| - |\eta_i| - w_2 \geq \min\{|e_i| : i \in \boldsymbol{P}\} - \|\eta\|_\infty - w_2 > \alpha$. Then $e_i^{(1)} \neq 0$ according to (2.5). Then $\boldsymbol{P} \subseteq \boldsymbol{S}_1^c := \{i : e_i^{(1)} \neq 0\}$.

For $\forall i \in \boldsymbol{E}^c$, $e_i = 0$, $|r_i^{(1)}| - |\eta_i| \leq |\eta_i - r_i^{(1)}| = |(e+\eta)_i - r_i^{(1)}| \leq \|(e+\eta) - r^{(1)}\|_1 \leq$

$\frac{\sqrt{m-q_1}\|\eta\|_2+\sum_{i\in \boldsymbol{E}\backslash \boldsymbol{G}}|e_i|}{c_{q_1}(A)-0.5} \le w_1$. So $|r_i^{(1)}| \le |\eta_i| + w_1 \le \|\eta\|_\infty + w_1 < \alpha$. Then $e_i^{(1)} = 0$ according to (2.5). Then $\boldsymbol{E}^c \subseteq \boldsymbol{S}_1 := \{i : e_i^{(1)} = 0\}$.

Next we will show for the $(k+1)$ st (e.g., $k = 1, 2, \dots$) iteration, if $\boldsymbol{E}^c \subseteq \boldsymbol{S}_k$ and $\boldsymbol{P} \subseteq \boldsymbol{S}_k^c$, then we will have $\boldsymbol{E}^c \subseteq S_{k+1}$ and $\boldsymbol{P} \subseteq S_{k+1}^c$. Thus $\boldsymbol{E}^c \subseteq \boldsymbol{S}_k$ and $\boldsymbol{P} \subseteq S_{k+1}^c$ for any $k \in Z_{\ge 0}$.

As $m(A) \ge |\boldsymbol{P}|$, from Lemma 2.1, we know that $A_{\boldsymbol{P}^c}$ is full column rank, and $m(A_{\boldsymbol{P}^c}) \ge m(A) - \lceil 0.5 \times |\boldsymbol{P}| \rceil$. So $m(A) \le m(A_{\boldsymbol{P}^c}) + \lceil 0.5 \times |\boldsymbol{P}| \rceil$. Combined with $|\boldsymbol{E}| \le m(A) + \lfloor \frac{t}{2} \rfloor = m(A) + \lfloor 0.5 \times |\boldsymbol{P}| \rfloor$, we have $|\boldsymbol{E}| \le m(A_{\boldsymbol{P}^c}) + \lceil 0.5 \times |\boldsymbol{P}| \rceil + \lfloor 0.5 \times |\boldsymbol{P}| \rfloor = m(A_{\boldsymbol{P}^c}) + |\boldsymbol{P}|$. So

$$|\boldsymbol{E}| - |\boldsymbol{P}| \le m(A_{\boldsymbol{P}^c}). \tag{2.22}$$

As $\boldsymbol{P} \subseteq \boldsymbol{S}_k^c$, so $\boldsymbol{S}_k \subseteq \boldsymbol{P}^c$. We have $\boldsymbol{E}^c \subseteq \boldsymbol{S}_k \subseteq \boldsymbol{P}^c$ and thus $|\boldsymbol{P}^c \backslash \boldsymbol{S}_k| \le |\boldsymbol{P}^c \backslash \boldsymbol{E}^c| = |\boldsymbol{E} \backslash \boldsymbol{P}| = |\boldsymbol{E}| - |\boldsymbol{P}| \le m(A_{\boldsymbol{P}^c}) = q_2$. From Lemma 2.1, we have that $A_{\boldsymbol{S}_k}$ is full column rank, $m(A_{\boldsymbol{S}_k}) \ge m(A_{\boldsymbol{P}^c}) - |\boldsymbol{P}^c \backslash \boldsymbol{S}_k| = q_2 - |\boldsymbol{P}^c \backslash \boldsymbol{S}_k|$, and

$$c_{q_2 - |\boldsymbol{P}^c \backslash \boldsymbol{S}_k|}(A_{\boldsymbol{S}_k}) \ge c_{q_2}(A_{\boldsymbol{P}^c}) > \frac{1}{2}. \tag{2.23}$$

Combined with (2.22), we have

$$\begin{aligned} m(A_{\boldsymbol{S}_k}) \ge & |\boldsymbol{E}| - |\boldsymbol{P}| - |\boldsymbol{P}^c \backslash \boldsymbol{S}_k| \\ = & |\boldsymbol{E} \backslash \boldsymbol{P}| - |\boldsymbol{P}^c \backslash \boldsymbol{S}_k| \\ = & |\boldsymbol{P}^c \backslash \boldsymbol{E}^c| - |\boldsymbol{P}^c \backslash \boldsymbol{S}_k| \\ = & (|\boldsymbol{P}^c| - |\boldsymbol{E}^c|) - (|\boldsymbol{P}^c| - |\boldsymbol{S}_k|) \\ = & |\boldsymbol{S}_k| - |\boldsymbol{E}^c| = |\boldsymbol{S}_k \backslash \boldsymbol{E}^c|. \end{aligned} \tag{2.24}$$

From Lemma 2.3, we know that

$$\|((e+\eta)_{\boldsymbol{S}_k} - r_{\boldsymbol{S}_k}^{(k+1)}\|_1 \leq \frac{\sqrt{m-q}\|(\eta\|_2}{c_{|\boldsymbol{S}_k\backslash\boldsymbol{E}^c|}(A_{\boldsymbol{S}_k}) - 0.5} = \frac{\sqrt{m-q}\|(\eta\|_2}{c_{|\boldsymbol{P}^c\backslash\boldsymbol{E}^c|-|\boldsymbol{P}^c\backslash\boldsymbol{S}_k|}(A_{\boldsymbol{S}_k}) - 0.5}. \tag{2.25}$$

As $|\boldsymbol{P}^c\backslash\boldsymbol{E}^c| \leq q_2$, so $|\boldsymbol{P}^c\backslash\boldsymbol{E}^c| - |\boldsymbol{P}^c\backslash\boldsymbol{S}_k| \leq q_2 - |\boldsymbol{P}^c\backslash\boldsymbol{S}_k|$, and from Proposition 2.1 we have $c_{|\boldsymbol{P}^c\backslash\boldsymbol{E}^c|-|\boldsymbol{P}^c\backslash\boldsymbol{S}_k|}(A_{\boldsymbol{S}_k}) \geq c_{q_2-|\boldsymbol{P}^c\backslash\boldsymbol{S}_k|}(A_{\boldsymbol{S}_k})$. Together with (2.23), we have $c_{|\boldsymbol{P}^c\backslash\boldsymbol{E}^c|-|\boldsymbol{P}^c\backslash\boldsymbol{S}_k|}(A_{\boldsymbol{S}_k})-0.5 \geq c_{q_2}(A_{\boldsymbol{P}^c}) - 0.5 > 0$. Combined with (2.25) we have $\|(e+\eta)_{\boldsymbol{S}_k} - r_{\boldsymbol{S}_k}^{(k+1)}\|_1 \leq \frac{\sqrt{m-q}\|\eta\|_2}{c_{q_2}(A_{\boldsymbol{P}^c})-0.5}$.

For $\forall i \in \boldsymbol{E}^c \subseteq \boldsymbol{S}_k$, $e_i = 0$, $|r_i^{(k+1)}| - |\eta_i| \leq |\eta_i - r_i^{(k+1)}| = |(e+\eta)_i - r_i^{(k+1)}| \leq \|(e+\eta)_{\boldsymbol{S}_k} - r_{\boldsymbol{S}_k}^{(k+1)}\|_1 \leq \frac{\sqrt{m-q}\|\eta\|_2}{c_{q_2}(A_{\boldsymbol{P}^c})-0.5} \leq w_1$. So $|r_i^{(k+1)}| \leq |\eta_i| + w_1 \leq \|\eta\|_\infty + w_1 < \alpha$. Then $e_i^{(k+1)} = 0$ according to (2.5). Then $\boldsymbol{E}^c \subseteq \boldsymbol{S}_{k+1} := \{i : e_i^{(k+1)} = 0\}$.

As $|\boldsymbol{P}^c\backslash\boldsymbol{E}^c| \leq \mathcal{m}(A_{\boldsymbol{P}^c})$ and $A_{\boldsymbol{P}^c}$ is full column rank, from Lemma 2.1, we know $A_{\boldsymbol{E}^c}$ is also full column rank, so $\sigma_{\min}(A_{\boldsymbol{E}^c}) > 0$. As $\frac{\sqrt{m-q}\|\eta\|_2}{c_{q_2}(A_{\boldsymbol{P}^c})-0.5} \geq \|(e+\eta)_{\boldsymbol{S}_k} - r_{\boldsymbol{S}_k}^{(k+1)}\|_1 = \|(y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k}x) - (y_{\boldsymbol{S}_k} - A_{\boldsymbol{S}_k}x^{(k+1)})\|_1 = \|A_{\boldsymbol{S}_k}(x - x^{(k+1)})\|_1 \geq \|A_{\boldsymbol{E}^c}(x - x^{(k+1)})\|_1 \geq \|A_{\boldsymbol{E}^c}(x - x^{(k+1)})\|_2 \geq \sigma_{\min}(A_{\boldsymbol{E}^c})\|x - x^{(k+1)}\|_2$, so we have $\|x - x^{(k+1)}\|_2 \leq \frac{\sqrt{m-q}\|\eta\|_2}{\sigma_{\min}(A_{\boldsymbol{E}^c})\times(c_{q_2}(A_{\boldsymbol{P}^c})-0.5)}$.

For$\forall i \in \boldsymbol{P}$, we have $|e_i| - |\eta_i| - \left|r_i^{(k+1)}\right| \leq \left|(e+\eta)_i\right| - \left|r_i^{(k+1)}\right| \leq \left|(e+\eta)_i - r_i^{(k+1)}\right|$

$\leq \left\|\left(e+\eta-r^{(k+1)}\right)_{\boldsymbol{P}}\right\|_2 = \left\|(y-Ax)_{\boldsymbol{P}} - \left(y - Ax^{(k+1)}\right)_{\boldsymbol{P}}\right\|_2 = \left\|A_{\boldsymbol{P}}\left(x - x^{(k+1)}\right)\right\|_2$

$\leq \sigma_{\max}(A_{\boldsymbol{P}})\left\|x - x^{(k+1)}\right\|_2 \leq \frac{\sigma_{\max}(A_{\boldsymbol{P}})\sqrt{m-q}\,\|\,\eta\|_2}{\sigma_{\min}(A_{\boldsymbol{E}^c})\times\left(c_{q_2}(A_{\boldsymbol{P}^c})-0.5\right)} \leq w_2$, so $\left|r_i^{(k+1)}\right| \geq |e_i| - |\eta_i| - w_2 \geq \min\{|e_i|:\ i \in \boldsymbol{P}\} - \|\eta\|_\infty - w_2 > \alpha$, then $e_i^{(k+1)} \neq 0$ according to (2.5). Then $\boldsymbol{P} \subseteq \boldsymbol{S}_{k+1}^{\boldsymbol{c}} := \{i : e_i^{(k+1)} \neq 0\}$.

c) As for any $k \in Z_{\geq 1}$, we have $\mathcal{m}(A_{\boldsymbol{S}_k}) \geq |\boldsymbol{S}_k\backslash\boldsymbol{E}^c| = |\boldsymbol{E} \cap \boldsymbol{S}_k|$ from (2.24). Since $A_{\boldsymbol{S}_k}$ is full column rank, the condition of Corollary 2.1 is satisfied. So $\|x^{(k+1)} - x\|_2$ is bounded.

Chapter 3

Robust PCA via ℓ_0-ℓ_1 Regularization

Robustly identifying the underlying low-rank structure in the presence of the outliers is also very challenging as the support and magnitude of the outliers are not known beforehand. In this chapter, we first propose a novel objective function where the nuclear norm captures the low-rank term, ℓ_0-'norm' addresses the sparse outlier term, and an ℓ_1-norm to deal with the additive noise term. The associated algorithm, termed Sparsity Regularized Principal Component Pursuit (SRPCP), is guaranteed to recover the underlying low-rank matrix exactly (or stably) under certain conditions. The advantage over the ℓ_1 relaxation approach will be demonstrated both theoretically and empirically. We further propose an Iterative Reweighted SRPCP method that uses log-determinant to capture the low-rank matrix instead, which leads to further performance improvement.

3.1 Introduction

Principal component analysis (PCA) is arguably one of the most widely used data analysis methods with numerous applications. However, its performance can significantly degrade if the data is corrupted by even a few outliers. As mentioned in a recent review [1],

outliers are becoming even more common in today's big data era. The goal of Robust PCA [39] is to recover the low-rank matrix $\boldsymbol{L}_0$ and sparse matrix $\boldsymbol{E}_0$ (which often models the outlier corruptions) from their composition $\boldsymbol{M}$ (possibly with additional dense noise). This problem has received a lot of interest in the past decade, with applications ranging from video analysis, face recognition, to recommendation systems. Robust PCA was first studied in the noiseless case [39]–[41], the underlying optimization problem is [41]:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \operatorname{rank}(\boldsymbol{L}) + \lambda \|\boldsymbol{E}\|_0 \quad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{E}, \tag{3.1}$$

which is known to be NP-hard. To make the problem computationally viable, [39]–[41] suggest relaxing the rank minimization to nuclear norm minimization and the ℓ_0-'norm' penalty to an ℓ_1-norm penalty, i.e.,

$$\min_{\boldsymbol{L},\boldsymbol{E}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{E}\|_1 \quad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{E}, \tag{3.2}$$

leading to a convex optimization based approach known as Principal Component Pursuit (PCP). Interestingly, one can recover both $\boldsymbol{L}_0$ and $\boldsymbol{E}_0$ exactly under certain conditions by solving this convex program. Since then, many variants have been proposed with the goal being either lower complexity or better performance. For a comprehensive review, we refer the interested readers to [42]. In this chapter, we focus on modifications to the objective function to deal with the outliers as well as possible dense inlier noise in order to achieve better performance. We provide a list of various existing regularization schemes below for comparison. The interested readers can find further details in the references therein.

As better surrogates for the original ℓ_0-'norm', the ℓ_p-norm and log-sum function on the sparse outlier term $\boldsymbol{E}$ are adopted in [43]. The corresponding optimization leads to the reweighted ℓ_1-norm utilizing the majorization minimization (MM) framework [85].

In real world applications, besides the sparse 'corruptions' $\boldsymbol{E}_0$, there is often small

magnitude dense inlier noise $\boldsymbol{N}$. The resulting model is:

$$\boldsymbol{M} = \boldsymbol{L}_0 + \boldsymbol{E}_0 + \boldsymbol{N}. \tag{3.3}$$

To address inlier noise, Zhou et al. [44] solved the following relaxed version of (3.2), known as Stable Principal Component Pursuit (SPCP):

$$\min_{\boldsymbol{L},\boldsymbol{E}} ||\boldsymbol{L}||_* + \lambda||\boldsymbol{E}||_1 \quad s.t. \quad ||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}||_F \leq \delta. \tag{3.4}$$

It was shown that the estimation error can be bounded under certain conditions.

Hsu et al. [45] analyzed the Lagrange form of (3.4):

$$\min_{\boldsymbol{L},\boldsymbol{E}} ||\boldsymbol{L}||_* + \lambda||\boldsymbol{E}||_1 + \frac{1}{2\mu}||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}||_F^2. \tag{3.5}$$

In light of the M-estimators, He et al. [36] proposed replacing $||\boldsymbol{E}||_1$ by implicit regularizers of robust M-estimators, i.e., $\varphi(\boldsymbol{E})$, and then solving the following optimization problem:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \mu||\boldsymbol{L}||_* + \varphi(\boldsymbol{E}) + \frac{1}{2}||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}||_F^2. \tag{3.6}$$

Similarly, Chartrand [46] proposed to replace the ℓ_1-norm by implicit regularizers (also called proximal p-norm [46]) of the p-Huber function.

To better approximate the ℓ_0-'norm', rather than using the ℓ_1-norm, Sun et al. [47] used the capped ℓ_1-norm on both the sparse term $\boldsymbol{E}$ and the singular values of $\boldsymbol{L}$:

$$\begin{aligned} &\min_{\boldsymbol{L},\boldsymbol{E}} \frac{1}{\theta_1}\sum_i \min\{\sigma_i(\boldsymbol{L}), \theta_1\} + \frac{1}{\theta_2}\sum_{i,j} \min\{|\boldsymbol{E}_{i,j}|, \theta_2\} \\ &s.t. \quad ||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}||_F \leq \delta. \end{aligned} \tag{3.7}$$

In [48] and [49], the following greedy approach was proposed that directly tackles the ℓ_0-'norm':

$$\min_{\boldsymbol{L},\boldsymbol{E}} ||\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}||_F^2 \quad s.t. \quad \text{rank}(\boldsymbol{L}) \leq r, ||\boldsymbol{E}||_0 \leq k. \tag{3.8}$$

Also, Ulfarsson et al. [50] proposed to use an ℓ_0 penalty to enforce both sparsity and low rank:

$$\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{E}} ||\boldsymbol{M}-\boldsymbol{A}\boldsymbol{B}^T-\boldsymbol{E}||_F^2 + h^2||\boldsymbol{E}||_0 \quad s.t. \quad \boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}_r. \tag{3.9}$$

However, these methods need to specify the rank (and sparsity), which are usually unknown in practice and hard to specify.

In the context of detecting contiguous outliers in the low-rank representation (termed DECOLOR), Zhou et al. [38] proposed an objective function whose degenerate form can be shown equivalent to the following:

$$||\boldsymbol{L}||_* + \beta||\boldsymbol{E}||_0 + \lambda||\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}||_F^2. \tag{3.10}$$

In this chapter, we first study a new objective function proposed in our recent conference paper [86]:

$$||\boldsymbol{L}||_* + \beta||\boldsymbol{E}||_0 + \lambda||\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}||_1. \tag{3.11}$$

This ℓ_0-ℓ_1 regularization framework is inspired by our robust linear regression work [87]. We will discuss the relation with our previous works in Section 3.2.A. Compared with (3.5)-(3.7), we use genuine ℓ_0-'norm' to enforce the sparseness of the outliers, and employ the ℓ_1-norm instead of the usual Frobenius norm on the noise term. Compared with (3.10), the only difference is the replacement of the Frobenius norm by the ℓ_1-norm on the noise term. But this replacement makes a big difference in that it not only significantly improves the recovery performances, but also enjoys many theoretical guarantees as we will see later.

We propose and analyze a new algorithm to minimize the objective function (3.11)

in Section 3.2. An important byproduct of our analysis is that both PCP and its missing entry version are shown to be stable to dense noise. Inspired by the superior performance of log-determinant [51], [52] in pursuing the low-rank structure, in Section 3.3, we replace the nuclear norm in (3.11) by the log-determinant, propose and analyze an algorithm to minimize the corresponding objective function. In both cases, our proposed algorithms iteratively detect and exclude suspected outlier entries and perform robust noisy matrix completion on the remaining entries. The robustness in each iteration results from the ℓ_1-norm on the noise term. Section 3.4 empirically studies the performance of the proposed methods and verify their effectiveness via two applications. Conclusions and future work are discussed in Section 3.5.

Notation: Throughout this chapter, bold capital letters denote matrices, e.g., $\boldsymbol{L}$, where $\boldsymbol{L}^{(k)}$ denotes the updated $\boldsymbol{L}$ in the kth iteration. The $\|\boldsymbol{L}\|_\infty$, $\|\boldsymbol{L}\|_1$, and $\|\boldsymbol{L}\|_0$ denote the ℓ_∞-norm, ℓ_1-norm, and ℓ_0-'norm'[1] of $\boldsymbol{L}$ seen as a long vector, respectively, while $\|\boldsymbol{L}\|_F$, $\|\boldsymbol{L}\|_*$, and $\|\boldsymbol{L}\|$ denote the Frobenius norm, nuclear norm, and the operator norm of the matrix $\boldsymbol{L}$, respectively. For a given subset $\Phi \subseteq [n_1] \times [n_2]$, $|\Phi|$ is its cardinality, $\mathcal{P}_\Phi(\boldsymbol{M})$ is the matrix obtained by setting the entries of $\boldsymbol{M}$ that are outside the index set Φ to zero. We use $\Phi_{(k)}$ to denote the updated Φ in the kth iteration.

3.2 Sparsity Regularized Principal Component Pursuit

3.2.1 Algorithm

We consider the following objective function to recover the low-rank component and sparse component:

$$J(\boldsymbol{L}, \boldsymbol{E}) = \|\boldsymbol{L}\|_* + \beta\|\boldsymbol{E}\|_0 + \lambda\|\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}\|_1. \tag{3.12}$$

To minimize the above nonconvex objective funciton, we propose an alternating

[1] ℓ_0-'norm' is not homogeneous and, hence, does not satisfy the axioms of a norm.

minimization 'like' approach that alternates between the following two steps:

Step 1: With index set $\Phi_{(k)}$ (which depends on $\boldsymbol{E}^{(k)}$), update

$$\boldsymbol{L}^{(k+1)} = \arg\min_{\boldsymbol{L}}(\|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L})\|_1);$$

Step 2: Fix $\boldsymbol{L}^{(k+1)}$, update

$$\boldsymbol{E}^{(k+1)} = \arg\min_{\boldsymbol{E}}(\beta\|\boldsymbol{E}\|_0 + \lambda\|\boldsymbol{M} - \boldsymbol{L}^{(k+1)} - \boldsymbol{E}\|_1),$$
$$\Phi_{(k+1)} = \{(i,j)|\boldsymbol{E}^{(k+1)}_{i,j} = 0\}.$$

The detailed procedure is summarized in Algorithm 1, where $\boldsymbol{L}^{(k+1)}$ and $\boldsymbol{E}^{(k+1)}$ denote the updated $\boldsymbol{L}$ and $\boldsymbol{E}$ at the $(k+1)$st iteration. $\Phi_{(k)}$ is the index set of the entries that are estimated to be free of large outliers in the kth iteration. This algorithm is a modification of our previous vanilla alternating minimization algorithm [86]. The distinguishing part w.r.t. our previous algorithm is Step 1, where we exclude the estimated outlier entries for estimating $\boldsymbol{L}$. As we will show in the numerical results, this leads to significant improvements in recovery performance.

At first glance, it seems more reasonable to use the Frobenius norm rather than the ℓ_1-norm in the third term of the objective function (3.12) and in Step 1, especially for Gaussian noise. This would become exactly the method DECOLOR with objective function (3.10). We want to point out that, in Step 1 of each iteration, though we aim to exclude outlier entries for estimating $\boldsymbol{L}$, we do not expect that all the outliers are identified by the previous iteration. It is likely that some outliers are not identified making it safer to use the ℓ_1-norm in Step 1 than the Frobenius norm, which is very sensitive to large residuals. As we will see in the numerical studies, this leads to significant improvements over DECOLOR in recovery performance. It is useful to note that in the noiseless matrix completion literature, there is a heuristic approach [88] that gradually deletes the suspected outliers in the observed entries

Algorithm 1 Sparsity Regularized Principal Component Pursuit (SRPCP)

Input: $\boldsymbol{M}$, β, λ

Initialization: $k = 0$, $\boldsymbol{E}^{(0)} = \boldsymbol{0}$, $\Phi_{(0)} = \{(i, j)|\boldsymbol{E}^{(0)}_{i,j} = 0\}$

While $J(\boldsymbol{L},\boldsymbol{E})$ **not converged DO:**

Iteration $k + 1$

Step 1: $\boldsymbol{L}^{(k+1)} = \arg\min_{\boldsymbol{L}}(\|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L})\|_1)$;

If $\|\boldsymbol{L}^{(k+1)}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L}^{(k+1)})\|_1 = \|\boldsymbol{L}^{(k)}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L}^{(k)})\|_1$, further update $\boldsymbol{L}^{(k+1)} = \boldsymbol{L}^{(k)}$.

Step 2: update

$$\boldsymbol{E}^{(k+1)}_{i,j} = \begin{cases} 0, & |(\boldsymbol{M} - \boldsymbol{L}^{(k+1)})_{i,j}| \le \frac{\beta}{\lambda} \\ (\boldsymbol{M} - \boldsymbol{L}^{(k+1)})_{i,j}, & \text{otherwise} \end{cases}$$

$$\Phi_{(k+1)} = \{(i, j)|\boldsymbol{E}^{(k+1)}_{i,j} = 0\}$$

$k := k + 1$

End While

Output: $\boldsymbol{L}$ and $\boldsymbol{E}$

and models the sparsity of remaining outliers via the ℓ_1-norm.

Our whole framework can be viewed as a 2D extension of our robust linear regression framework [87], where we use the genuine ℓ_0-'norm' to enforce the sparseness of the outliers and employ an ℓ_1-norm on the noise term. In the iterations, we delete the estimated outlier entries and still use a robust ℓ_1-norm for estimating the signal. We refer the interested readers to [87] for a detailed analysis of the benefits of using this framework under the linear regression setting.

In case there are multiple solutions for $\min_{\boldsymbol{L}} \|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M}-\boldsymbol{L})\|_1$, and $\boldsymbol{L}^{(k)}$ happens to be one of these solutions, we set $\boldsymbol{L}^{(k+1)} = \boldsymbol{L}^{(k)}$ to make the algorithm more stable.

At the beginning, we have no information about outliers except that they are sparse. So we simply initialize $\boldsymbol{E}^{(0)} = \boldsymbol{0}$ and the corresponding index set $\Phi_{(0)}$ to be all entries. Then in Step 1 of the first iteration, SRPCP solves the following:

$$\min_{\boldsymbol{L}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{M}-\boldsymbol{L}\|_1, \tag{3.13}$$

which is equivalent to PCP in (3.2).

Solutions for Each Step:

In Step 1, the subproblem is convex, which is equivalent to the following problem: $\min_{\boldsymbol{L},\boldsymbol{E}} \|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{E}\|_1, s.t.\ \boldsymbol{M} = \boldsymbol{L}+\boldsymbol{E}$. An alternating direction method of multipliers (ADMM) algorithm was proposed in [89] to solve this, and this problem was shown to be equivalent to PCP with missing entries [39], [90], [91] in terms of $\boldsymbol{L}$:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{E}\|_1,\ s.t.\ \mathcal{P}_{\Phi_{(k)}}\boldsymbol{M} = \mathcal{P}_{\Phi_{(k)}}(\boldsymbol{L}+\boldsymbol{E}) \tag{3.14}$$

In Step 2, though the subproblem is not convex, we can directly find the global optimal solution through elementwise hard thresholding [50], [86], which is detailed in Algorithm 1.

From Step 2 of SRPCP, if any entry of $|\boldsymbol{M}-\boldsymbol{L}^{(k+1)}|$ is larger than $\frac{\beta}{\lambda}$, this entry will be

considered as an outlier corrupted entry. In general, $\frac{\beta}{\lambda}$ should be set at least larger than the inlier noise level. Our analysis shows that under certain conditions on the model parameters, if $\frac{\beta}{\lambda}$ is greater than some certain threshold, we can guarantee that all the inlier entries are kept in every iteration and the removed entries are purely outliers. So our preference has been to set it much larger than the inlier noise level. In practice the parameter λ is fixed, and the adaptation to the noise level is transferred to parameter β.

Complexity: The main computational cost of SRPCP is Step 1. Assume $n_1 \leq n_2$. The complexity of ADMM [89] to solve Step 1 is $\mathcal{O}(n_1^2 n_2 \frac{1}{\epsilon})$ for achieving an error of ϵ. Our Theorem 3.1 in next subsection guarantees that SRPCP converges in a finite number of iterations, and we empirically notice that it usually converges in 10 iterations. Therefore the worst case complexity of SRPCP would be the number of iterations times $\mathcal{O}(n_1^2 n_2 \frac{1}{\epsilon})$. However, we use the previous iteration's $\boldsymbol{L}^{(k)}$ as the warm-start for the ADMM to solve current iteration's Step 1. As a result, the actual complexity is less than that. It is worth noting that, to solve large-scale problems, [89] further proposed factorizing the low-rank matrix into two much smaller matrices to approximately solve Step 1, with some theoretical guarantees.

3.2.2 Theoretical Analysis

In this subsection, we study the main properties of SRPCP. We first establish its convergence property. Then, we analyze its behaviors in both the noiseless case and the noisy case. Finally, we show that both PCP and its missing entry version are stable to dense noise.

3.2.2.1 Convergence Property

The following theorem establishes the convergence of the iterates generated by SRPCP to a local minimizer.

Theorem 3.1. ***(Convergence property)*** *SRPCP converges in a finite number of iterations to a*

fixed point, which is a local optimum. Moreover, the objective function is strictly decreasing before convergence.

The proof of the Theorem is detailed in Appendix 3.6.1.

3.2.2.2 Noiseless Case Analysis

In this subsection, we analyze the behaviors of SRPCP when there is no inlier noise, i.e., $\boldsymbol{M} = \boldsymbol{L}_0 + \boldsymbol{E}_0$. We first show the exact recovery property of SRPCP under the deterministic sparsity model. Then, we turn to the random sparsity model, aiming to show the potential for SRPCP to go beyond PCP. The analysis benefits greatly from the results in [90]. We first quote the incoherence condition with parameter μ from [39] that is needed for this discussion.

The singular value decomposition of $\boldsymbol{L}_0 \in R^{n_1 \times n_2}$ is

$$\boldsymbol{L}_0 = \boldsymbol{U}\Sigma\boldsymbol{V}^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T,$$

where r is the rank of $\boldsymbol{L}_0$, $\sigma_1, ..., \sigma_r$ are the positive singular values, and $\boldsymbol{U} = [u_1, ..., u_r]$, $\boldsymbol{V} = [v_1, ..., v_r]$ are the matrices of left- and right-singular vectors. Then, the incoherence condition with parameter μ states that

$$\max_i \|\boldsymbol{U}^T e_i\|_2^2 \leq \frac{\mu r}{n_1}, \quad \max_i \|\boldsymbol{V}^T e_i\|_2^2 \leq \frac{\mu r}{n_2}, \tag{3.15}$$

and

$$\|\boldsymbol{U}\boldsymbol{V}^T\|_\infty \leq \sqrt{\frac{\mu r}{n_1 n_2}}. \tag{3.16}$$

We also borrow some definitions from [90] for analyzing deterministic outliers/errors: 1) Let d be the maximum number of outliers on each row/column; 2) $\|X\| \leq \eta d \|X\|_\infty$ for any matrix X that is supported on the set of outlier entries; 3) Let $\alpha = \sqrt{\frac{\mu r d}{n_1}} + \sqrt{\frac{\mu r d}{n_2}} + \sqrt{\frac{\mu r d}{\max(n_1, n_2)}}$. Now we are ready to state the exact recovery property of SRPCP.

Theorem 3.2. ***(Exact recovery in noiseless case)*** *If* $\sqrt{\frac{\mu rd}{\min(n_1,n_2)}}\left(1+\sqrt{\frac{\min(n_1,n_2)}{\max(n_1,n_2)}}+\eta\sqrt{\frac{d}{\max(n_1,n_2)}}\right)\leq 0.5$, *SRPCP with* $\lambda\in[\frac{1}{1-2\alpha}\sqrt{\frac{\mu r}{n_1n_2}},\frac{1-\alpha}{\eta d}-\sqrt{\frac{\mu r}{n_1n_2}}]$ *and any* $\beta>0$ *recovers* $\boldsymbol{L}_0$ *exactly in two iterations. If additionally*[2] $\beta<\lambda\min\{|(\boldsymbol{E}_0)_{i,j}|:(\boldsymbol{E}_0)_{i,j}\neq 0\}$, *then SRPCP recovers both* $\boldsymbol{L}_0$ *and* $\boldsymbol{E}_0$ *exactly.*

The proof of the Theorem is in Appendix 3.6.2. The proof is based on Theorem 3 of [90], which studies the deterministic outliers/errors and erasures, and counts them as the same. Actually under the condition of Theorem 3.2, both PCP and SRPCP succeed. Next, we borrow a recovery guarantee for random errors and erasures from [90], and discuss the potential for SRPCP to go beyond PCP. In the random model, each entry is observed (e.g., $\in\Phi$ in (3.14)) with probability at least p_0, and each entry is an outlier/error with probability at most τ.

Theorem 3.3. ***(Simplified Theorem 2 of [90])*** *Set* $n=\min\{n_1,n_2\}$. *Assume that the signs of nonzero entries of* $\boldsymbol{E}_0$ *are symmetric* ±1 *Bernoulli random variables independent of all others. Then, there exist absolute constants* C *and* ρ_r *independent of* n, μ, *and* r *such that, with probability at least* $1-Cn^{-10}$, *(3.14) with tradeoff parameter* $\lambda=\frac{1}{32\sqrt{p_0n}}$ *recovers* $\boldsymbol{L}_0$ *exactly provided that* $p_0(1-\tau)^2\geq\rho_r\frac{\mu r\log^6 n}{n}$.

Note that the conclusion of the above theorem holds for a range of values of λ [90]. In the following discussion, we assume that the λ we set is always in the valid range of values for exact recovery. As pointed out by [90], one interesting observation is that p_0 can approach zero faster than $1-\tau$. For Step 1 of our first iteration, we have $p_0=1$. Suppose $(1-\tau)^2<\rho_r\frac{\mu r\log^6 n}{n}$, the condition for exact recovery is not satisfied. In Step 2, let the fraction of nonzero entries in the estimated $\boldsymbol{E}^{(1)}$ be $c\tau$. Then in Step 1 of the second iteration, we exclude these entries for estimating $\boldsymbol{L}$. Among these excluded entries, p percent of them are indeed outliers. For

[2]This additional constraint is not needed if we are aware that there is no inlier noise, since $\boldsymbol{E}_0$ can be recovered exactly by $\boldsymbol{M}-\boldsymbol{L}_0$.

Table 3.1. Value of $1 - \frac{(1-\tau)(1-\sqrt{1-c\tau})}{c\tau}$ for different c and τ

c \ τ	0.01	0.05	0.1	0.2	0.3	0.4
0.5	0.5044	0.5220	0.5442	0.5895	0.6358	0.6833
1	0.5038	0.5189	0.5381	0.5777	0.6189	0.6619
1.5	0.5031	0.5157	0.5317	0.5644	0.5981	0.6325

this subproblem, we can view each entry as being observed with probability $1 - c\tau$, and each entry is an outlier/error with probability $\frac{\tau - pc\tau}{1-c\tau}$. Using the above theorem with this parameter setting, we have that if

$$(1 - c\tau)\left(1 - \frac{\tau - pc\tau}{1 - c\tau}\right)^2 > (1 - \tau)^2, \tag{3.17}$$

then we take a forward step toward exact recovery. Exact recovery can be guaranteed w.h.p. if $(1 - c\tau)(1 - \frac{\tau - pc\tau}{1-c\tau})^2 \geq \rho_r \frac{\mu r \log^6 n}{n}$.

For (3.17) to hold, we require $p > 1 - \frac{(1-\tau)(1-\sqrt{1-c\tau})}{c\tau}$. The following Table 3.1 lists corresponding values for different c and τ. We can see that the requirement on p is not very demanding in order to make a forward step toward exact recovery.

For simplicity, we assume that in Step 2, the fraction of nonzero entries in the estimated $\boldsymbol{E}^{(2)}$ is also $c\tau$. If the fraction of the true outliers among those nonzero entries is larger than p, then we take a further step toward exact recovery in Step 1 of the third iteration, and so on. In our numerical experiments (Section 4.5.1), we notice that there are many cases where PCP (the first iteration of SRPCP) does not give exact recovery, while SRPCP gives exact recovery at the end of the iterations.

3.2.2.3 Analysis in the Noisy Case

In this subsection, we analyze the behaviors of SRPCP when there is dense inlier noise, i.e., $\boldsymbol{M} = \boldsymbol{L}_0 + \boldsymbol{E}_0 + \boldsymbol{N}$. To simplify analysis and presentation, we assume that the matrices are all square and write $n = n_1 = n_2$ in this subsection. The conclusions can be

extended to the general rectangular matrices. We first introduce some definitions from [40], which first establishes the deterministic guarantee for PCP.

Given a matrix pair $\boldsymbol{X}_0 = (\boldsymbol{L}_0, \boldsymbol{E}_0)$, let the space Ω be the set of all matrices that have support contained within the support of $\boldsymbol{E}_0$:

$$\Omega = \{\boldsymbol{Z} \in \mathbb{R}^{n\times n} | \text{supp}(\boldsymbol{Z}) \subseteq \text{supp}(\boldsymbol{E}_0)\} \subset \mathbb{R}^{n\times n}.$$

Let $\mathcal{P}_\Omega$ denote the orthogonal projection onto this space. Then $\mathcal{P}_\Omega(\boldsymbol{M})$ is the matrix obtained by setting the entries of $\boldsymbol{M}$ that are outside the support of $\boldsymbol{E}_0$ to zero. The subspace orthogonal to Ω is denoted Ω^c, and it consists of matrices with complementary support, i.e., supported on $\text{supp}(\boldsymbol{E}_0)^c$. The projection onto Ω^c is denoted $\mathcal{P}_{\Omega^c}$.

Let $r = \text{rank}(\boldsymbol{L}_0)$, and let $\boldsymbol{L}_0 = \boldsymbol{U}\Sigma\boldsymbol{V}^T$ denote the compact singular value decomposition of $\boldsymbol{L}_0$, with $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{n\times r}$ and $\Sigma \in \mathbb{R}^{r\times r}$. We will let T denote the subspace generated by matrices with the same column space or row space as $\boldsymbol{L}_0$:

$$T = \{\boldsymbol{U}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{V}^T | \boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times r}\} \subset \mathbb{R}^{n\times n},$$

and $\mathcal{P}_T$ be the projection operator onto this subspace under the inner product $< \boldsymbol{M}_1, \boldsymbol{M}_2 >= \text{tr}(\boldsymbol{M}_1^T\boldsymbol{M}_2)$. We have $\mathcal{P}_T(\boldsymbol{M}) = \mathcal{P}_U\boldsymbol{M} + \boldsymbol{M}\mathcal{P}_V - \mathcal{P}_U\boldsymbol{M}\mathcal{P}_V$. Here $\mathcal{P}_U = \boldsymbol{U}\boldsymbol{U}^T$ and $\mathcal{P}_V = \boldsymbol{V}\boldsymbol{V}^T$. The space orthogonal to T is denoted $T^\perp$, and the corresponding projection is denoted $\mathcal{P}_{T^\perp}(\boldsymbol{M})$. The space $T^\perp$ consists of matrices with row-space orthogonal to the row-space of $\boldsymbol{L}_0$ and column-space orthogonal to the column-space of $\boldsymbol{L}_0$. We have that $\mathcal{P}_{T^\perp}(\boldsymbol{M}) = (I_{n\times n} - \mathcal{P}_U)\boldsymbol{M}(I_{n\times n} - \mathcal{P}_V)$.

Define $\xi \doteq \max_{\boldsymbol{Z}\in T, \|\boldsymbol{Z}\|\leq 1} \|\boldsymbol{Z}\|_\infty$, and $\upsilon \doteq \max_{\boldsymbol{Z}\in\Omega, \|\boldsymbol{Z}\|_\infty \leq 1} \|\boldsymbol{Z}\|$. Theorem 3.4 states that the estimation error of SRPCP is bounded when $\upsilon\xi < \frac{1}{8}$. As discussed in [90], this condition implies the condition in Theorem 3.2 holds. The analysis benefits greatly from the analysis of PCP in the noiseless case [40] and the analysis of SPCP [44] for the noisy case.

Theorem 3.4. *Define* $C(\xi,\upsilon)=\sqrt{\frac{11}{12}}\frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{\xi^2\upsilon}$. *Suppose* $\upsilon\xi<\frac{1}{8}$, $\lambda\in(\frac{\xi}{1-5\upsilon\xi},\frac{1-4\xi\upsilon}{\upsilon})$, *fix any* $\beta>\lambda[\frac{2+\sqrt{2}}{2}+C(\xi,\upsilon)(\sqrt{n}+3\lambda n)]\|\boldsymbol{N}\|_F$, *SRPCP with input* $\boldsymbol{M}=\boldsymbol{L}_0+\boldsymbol{E}_0+\boldsymbol{N}$ *guarantees that:*

(a) Significant outlier entries at iteration i *denoted by* $G_{(i)}$ *and defined as*

$$G_{(1)}:=\{(i,j):|(\boldsymbol{E}_0)_{i,j}|>\frac{\beta}{\lambda}+[\frac{2+\sqrt{2}}{2}+C(\xi,\upsilon)(\sqrt{n}+3\lambda n)]\|\boldsymbol{N}\|_F\}$$

$$G_{(k+1)}:=\{(i,j):|(\boldsymbol{E}_0)_{i,j}|>\frac{\beta}{\lambda}+[\frac{2+\sqrt{2}}{2}+C(\xi,\upsilon)(\sqrt{n}+3\lambda\sqrt{n^2-|G_{(k)}|})]\|\boldsymbol{N}\|_F\}$$

satisfy $G_{(k)}\subseteq supp(\boldsymbol{E}^{(k)})\subseteq supp(\boldsymbol{E}_0)$, *and* $G_{(1)}\subseteq G_{(2)}\subseteq ...$

(b) $$\|\boldsymbol{L}^{(1)}-\boldsymbol{L}_0\|_F\leq[\frac{\sqrt{2}}{2}+C(\xi,\upsilon)(\sqrt{n}+3\lambda n)]\|\boldsymbol{N}\|_F$$

$$\|\boldsymbol{L}^{(k+1)}-\boldsymbol{L}_0\|_F\leq[\frac{\sqrt{2}}{2}+C(\xi,\upsilon)(\sqrt{n}+3\lambda\sqrt{n^2-|G_{(k)}|})]\|\boldsymbol{N}\|_F$$

The proof of the Theorem is in Appendix 3.6.3.

Note that we have defined $\Phi_{(k)}=\{(i,j)|\boldsymbol{E}^{(k)}_{i,j}=0\}$ in Algorithm 1. So we have $\text{supp}(\boldsymbol{E}^{(k)})=\Phi^c_{(k)}$, and $G_{(k)}\subseteq\Phi^c_{(k)}$.

Remark 3.1 *First, (a) guarantees that there is no false alarm when identifying outlier entries in the noisy case. Further, all the significant outliers (e.g., indexed by* $G_{(1)}$*), which are usually the most troublesome ones, are guaranteed to be identified and excluded for the next iteration. Note that if the magnitudes of the nonzero entries of* $\boldsymbol{E}_0$ *are all large enough, e.g.,* $G_{(2)}=supp(\boldsymbol{E}_0)$*, we can even guarantee the exact support recovery of the outliers.*

Second, $G_{(1)}\subseteq G_{(2)}\subseteq ...$ *implies* $|G_{(1)}|\leq|G_{(2)}|\leq ...$*, so more and more outliers can be* guaranteed *to be identified and excluded for the next iteration. Then, we immediately see that the error bound for* $\|\boldsymbol{L}^{(k)}-\boldsymbol{L}_0\|_F$ *is decreasing with iterations. This agrees with the intuition that correcting erasures (e.g., our excluded entries) with known locations is easier than correcting errors with unknown locations [90]. We remind the reader that* $G_{(1)}\subseteq G_{(2)}\subseteq ...$ *does not imply* $\Phi^c_{(1)}\subseteq\Phi^c_{(2)}\subseteq ...$

Third, the bound on $\|\boldsymbol{L}^{(k)}-\boldsymbol{L}_0\|_F$ *shows that the widely used PCP (our first iteration) and its missing entry version in (3.14) (which is equivalent to our Step 1 in terms of* $\boldsymbol{L}$*) are actually stable to dense inlier noise! Note that they were both designed and analyzed for the*

noiseless case. The detailed bound for (3.14) can be found in (3.31) in Appendix 3.6.3.

Lastly, note that the parameters λ and β we required do not depend on the magnitudes of the outliers.

3.3 Iterative Reweighted Sparsity Regularized Principal Component Pursuit

3.3.1 Algorithm

Recall that in the objective function (3.12), the nuclear norm is the convex surrogate for the rank. In this section, inspired by the efficacy of nonconvex log-determinant heuristic [51], [52] for promoting low-rank, we propose the minimization of the following nonconvex objective function:

$$\begin{aligned} J(\boldsymbol{L},\boldsymbol{E}) =& \gamma \log\det\left((\boldsymbol{L}\boldsymbol{L}^T)^{\frac{1}{2}} + \varepsilon I\right) + \beta\|\boldsymbol{E}\|_0 + \lambda_0\|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}\|_1 \\ =& \gamma\sum_i \log(\sigma_i(\boldsymbol{L}) + \varepsilon) + \beta\|\boldsymbol{E}\|_0 + \lambda_0\|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}\|_1 \end{aligned} \tag{3.18}$$

where λ_0 is fixed to be $\frac{1}{\sqrt{n}}$, and ε is a small numerical constant.

To minimize the above objective function, we use the same framework as SRPCP and alternate between the following two steps:

Step 1: With index set $\Phi_{(k)}$ (which depends on $\boldsymbol{E}^{(k)}$), update $\boldsymbol{L}^{(k+1)}$ such that

$$\begin{aligned} &\gamma\sum_i \log(\sigma_i(\boldsymbol{L}^{(k+1)}) + \varepsilon) + \lambda_0\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M}-\boldsymbol{L}^{(k+1)})\|_1 \\ \leq& \gamma\sum_i \log(\sigma_i(\boldsymbol{L}^{(k)}) + \varepsilon) + \lambda_0\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M}-\boldsymbol{L}^{(k)})\|_1; \end{aligned}$$

Step 2: Fix $\boldsymbol{L}^{(k+1)}$, update

$$\boldsymbol{E}^{(k+1)} = \arg\min_{\boldsymbol{E}}(\beta\|\boldsymbol{E}\|_0 + \lambda_0\|\boldsymbol{M} - \boldsymbol{L}^{(k+1)} - \boldsymbol{E}\|_1),$$
$$\Phi_{(k+1)} = \{(i,j)|\boldsymbol{E}^{(k+1)}_{i,j} = 0\}.$$

The subproblem in Step 1 is difficult to solve. Using the majorization minimization (MM) framework, we construct the upper bounding surrogate function for $\gamma\sum_i \log(\sigma_i(\boldsymbol{L})+\varepsilon)+\lambda_0\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M}-\boldsymbol{L})\|_1$ at $\boldsymbol{L}^{(k)}$ as $\gamma\sum_i[\log(\sigma_i(\boldsymbol{L}^{(k)})+\varepsilon)+\frac{1}{\sigma_i(\boldsymbol{L}^{(k)})+\varepsilon}(\sigma_i(\boldsymbol{L})-\sigma_i(\boldsymbol{L}^{(k)}))]+\lambda_0\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M}-\boldsymbol{L})\|_1$, and update $\boldsymbol{L}^{(k+1)}$ such that:

$$\|\boldsymbol{L}^{(k+1)}\|_{w,*} + \lambda_0\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L}^{(k+1)})\|_1 \leq \|\boldsymbol{L}^{(k)}\|_{w,*} + \lambda_0\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L}^{(k)})\|_1 \tag{3.19}$$

where $\|\cdot\|_{w,*}$ is the weighted nuclear norm with the weights $w_i = \frac{\gamma}{\sigma_i(\boldsymbol{L}^{(k)})+\varepsilon}$.

Step 2 is similar to step 2 in SRPCP and has a closed form solution.

Similar to SRPCP, we initialize $\boldsymbol{E}^{(0)} = \boldsymbol{0}$ and the corresponding index set $\Phi_{(0)}$ to be all entries. The detailed procedure is summarized in Algorithm 2. Compared with SRPCP in Algorithm 1, the main difference is Step 1. Here we use the weighted nuclear norm and iteratively update the weight. So we call it Iterative Reweighted Sparsity Regularized Principal Component Pursuit (IR-SRPCP).

Solving the subproblem (3.19)**:** Note that in Step 1 of the first iteration, we have the initial weight vector $w = (1, ..., 1)$, but the $\boldsymbol{L}^{(0)}$ is not defined. However, we can directly get the optimal solution of the convex problem $\boldsymbol{L}^{(1)} = \arg\min_{\boldsymbol{L}}(\|\boldsymbol{L}\|_* + \lambda_0\|\mathcal{P}_{\Phi_{(0)}}(\boldsymbol{M} - \boldsymbol{L})\|_1) = \arg\min_{\boldsymbol{L}}(\|\boldsymbol{L}\|_* + \lambda_0\|\boldsymbol{M} - \boldsymbol{L}\|_1)$. In the following iterations, the subproblem is nonconvex in general, due to the ascending nature of the weight vector, i.e., $w_1 \leq w_2 \leq ... \leq w_d$, where $d = \min(n_1, n_2)$. Notice that $\|\boldsymbol{L}\|_{w,*} = \sum_i w_i\sigma_i = \sum_i w_d\sigma_i - \sum_i(w_d - w_i)\sigma_i = w_d\|\boldsymbol{L}\|_* - \sum_i(w_d - w_i)\sigma_i$. Since $(w_d - w_1) \geq (w_d - w_2) \geq ... \geq (w_d - w_d) = 0$, the term $\sum_i(w_d - w_i)\sigma_i$ is convex [51]. Therefore the objective in (3.19) can be expressed as the difference of two convex functions:

Algorithm 2 Iterative Reweighted Sparsity Regularized Principal Component Pursuit (IR-SRPCP)

Input: $\boldsymbol{M}$, γ, β

Initialization: $k = 0$, $w = (1, ..., 1)$, $\boldsymbol{E}^{(0)} = \boldsymbol{0}$, $\Phi_{(0)} = \{(i, j)|\boldsymbol{E}^{(0)}_{i,j} = 0\}$

While $J(\boldsymbol{L},\boldsymbol{E})$ **not converged DO:**

Iteration $k + 1$

Step 1: fix $\boldsymbol{E}^{(k)}$ and $\Phi_{(k)}$, update $\boldsymbol{L}^{(k+1)}$ such that

$$||\boldsymbol{L}^{(k+1)}||_{w,*} + \lambda_0||\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L}^{(k+1)})||_1 \leq \ ||\boldsymbol{L}^{(k)}||_{w,*} + \lambda_0||\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L}^{(k)})||_1,$$

update $w_i = \frac{\gamma}{\sigma_i(\boldsymbol{L}^{(k+1)})+\varepsilon}$.

Step 2: fix $\boldsymbol{L}^{(k+1)}$, update

$$\boldsymbol{E}^{(k+1)}_{i,j} = \begin{cases} 0, & |(\boldsymbol{M} - \boldsymbol{L}^{(k+1)})_{i,j}| \leq \frac{\beta}{\lambda_0} \\ (\boldsymbol{M} - \boldsymbol{L}^{(k+1)})_{i,j}, & \textit{otherwise} \end{cases}$$

$$\Phi_{(k+1)} = \{(i, j)|\boldsymbol{E}^{(k+1)}_{i,j} = 0\}$$

$k := k + 1$

End While

Output: $\boldsymbol{L}$ and $\boldsymbol{E}$

$w_d\|\boldsymbol{L}\|_* + \lambda_0\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M}-\boldsymbol{L})\|_1$ and $\sum_i(w_d - w_i)\sigma_i$. Thus (3.19) can be guaranteed if we apply the convex-concave procedure [92], [93] to solve this Difference of Convex (DC) problem. However, it is known to have a slow convergence rate. We adapt the ADMM approach proposed in [94] instead to minimize $\|\boldsymbol{L}\|_{w,*} + \lambda_0\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M}-\boldsymbol{L})\|_1$, which is equivalent to $\min_{\boldsymbol{L},\boldsymbol{E}} \|\boldsymbol{L}\|_{w,*} + \lambda_0\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{E}\|_1$, *s.t.* $\boldsymbol{M} = \boldsymbol{L} + \boldsymbol{E}$. Its augmented Lagrange function is

$$\Gamma(\boldsymbol{L},\boldsymbol{E},\boldsymbol{Y},\theta) = \|\boldsymbol{L}\|_{w,*} + \lambda_0\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{E}\|_1 + <\boldsymbol{Y},\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}> + \frac{\theta}{2}\|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}\|_F^2.$$

The detailed ADMM procedure is summarized in Algorithm 3. We call this Weighted Nuclear Norm Minimization for Robust Matrix Completion (WNNM-RMC). In Algorithm 3, $\mathcal{S}_{\frac{\lambda_0}{\theta^{(k)}}}(\text{scalar}) = \text{sign}(\text{scalar}) \times \max(|\text{scalar}| - \frac{\lambda_0}{\theta^{(k)}}, 0)$ is the soft-thresholding operator, while $\mathcal{S}_{\frac{w}{\theta^{(k)}}}(\Sigma)$ is the generalized soft-thresholding operator with weight vector w [94]: $[\mathcal{S}_{\frac{w}{\theta^{(k)}}}(\Sigma)]_{i,i} = \max(\Sigma_{i,i} - \frac{w_i}{\theta^{(k)}}, 0)$. It is worth noting that when we update $\boldsymbol{L}$ with others fixed in Algorithm 3, the solution is globally optimal [51], [94].

Complexity: Assume $n_1 \leq n_2$. The per-iteration complexity of WNNM-RMC is $\mathcal{O}(n_1^2 n_2)$, but it only requires $\mathcal{O}(\log(\frac{1}{\epsilon}))$ number of iterations to achieve an error of ϵ, as we will see in Theorem 3.6. For the overall method IR-SRPCP, the worst case complexity would be the number of outer-iterations times $\mathcal{O}(n_1^2 n_2 \log(\frac{1}{\epsilon}))$. However, we use the previous iteration's $\boldsymbol{L}^{(k)}$ as the warm-start for WNNM-RMC to solve current iteration's Step 1. Hence the actual complexity is less than that. Empirically, we notice that IR-SRPCP usually converges in around 10 iterations. For large-scale problems, it is quite promising to use the same bilinear factorization strategy [89] to approximately solve the Step 1 of IR-SRPCP, which is our future work.

Algorithm 3 Weighted Nuclear Norm Minimization for Robust Matrix Completion (WNNM-RMC)

Input: $\boldsymbol{M}$, non-descending weight w, Φ

Initialization: $\theta^{(0)} > 0, k = 0, \rho > 1, \boldsymbol{L}^{(0)}, \boldsymbol{Y}^{(0)}$

DO:

Iteration $k + 1$

//update $\boldsymbol{E}^{(k+1)} = \arg\min_{\boldsymbol{E}} \Gamma(\boldsymbol{L}^{(k)}, \boldsymbol{E}, \boldsymbol{Y}^{(k)}, \theta^{(k)})$

$$\boldsymbol{E}_{i,j}^{(k+1)} = \begin{cases} \mathcal{S}_{\frac{\lambda_0}{\theta^{(k)}}}((\boldsymbol{M} + \boldsymbol{Y}^{(k)}/\theta^{(k)} - \boldsymbol{L}^{(k)})_{i,j}), (i,j) \in \Phi \\ (\boldsymbol{M} + \boldsymbol{Y}^{(k)}/\theta^{(k)} - \boldsymbol{L}^{(k)})_{i,j}, \quad otherwise \end{cases}$$

//update $\boldsymbol{L}^{(k+1)} = \arg\min_{\boldsymbol{L}} \Gamma(\boldsymbol{L}, \boldsymbol{E}^{(k+1)}, \boldsymbol{Y}^{(k)}, \theta^{(k)})$

$(\boldsymbol{U}, \Sigma, \boldsymbol{V})$=svd$(\boldsymbol{M} + \boldsymbol{Y}^{(k)}/\theta^{(k)} - \boldsymbol{E}^{(k+1)})$,

$\boldsymbol{L}^{(k+1)} = \boldsymbol{U}\mathcal{S}_{\frac{w}{\theta^{(k)}}}(\Sigma)\boldsymbol{V}^T$,

//update $\boldsymbol{Y}$

$\boldsymbol{Y}^{(k+1)} = \boldsymbol{Y}^{(k)} + \theta^{(k)}(\boldsymbol{M} - \boldsymbol{L}^{(k+1)} - \boldsymbol{E}^{(k+1)})$

//update θ

$\theta^{(k+1)} = \rho\theta^{(k)}$

$k := k + 1$

While $||\boldsymbol{M} - \boldsymbol{L}^{(k+1)} - \boldsymbol{E}^{(k+1)}||_F / ||\boldsymbol{M}||_F >$ tolerance

Output: $\boldsymbol{L}$

3.3.2 Theoretical Analysis

In this section, we establish the convergence properties of both IR-SRPCP and WNNM-RMC, and provide some justifications for IR-SRPCP.

Theorem 3.5. ***(Convergence property of IR-SRPCP)*** *The value of the objective function is non-increasing in each iteration of IR-SRPCP and it will always converge.*

The proof is similar to part a) of the Appendix 3.6.1, and can be found in the supple-

mental material.

Theorem 3.6. ***(Convergence property of WNNM-RMC)*** *The sequences* $\{\boldsymbol{L}^{(k)}\}$ *and* $\{\boldsymbol{E}^{(k)}\}$ *generated by WNNM-RMC converge Q-linearly. Further,* $\lim_{k\to\infty} \|\boldsymbol{M} - \boldsymbol{L}^{(k)} - \boldsymbol{E}^{(k)}\|_F = 0$.

The proof is inspired by the proof in [94], and can be found in the supplemental material.

Theorem 3.6 shows that the iterates $\boldsymbol{L}^{(k)}$ and $\boldsymbol{E}^{(k)}$ of WNNM-RMC approach feasibility, i.e., $\boldsymbol{L}^{(k)} + \boldsymbol{E}^{(k)} \to \boldsymbol{M}$.

Justifications for IR-SRPCP: Note that the first iteration of IR-SRPCP is the same as the first iteration of SRPCP. Hence all the guarantees for the first iteration of SRPCP also apply here. In the following iterations, the only difference between them is the utilization of the weighted nuclear norm in IR-SRPCP, where the weight is proportional to $\frac{1}{\sigma_i(\boldsymbol{L}^{(k)})+\varepsilon}$. Intuitively, for the estimated small singular values, it uses large weights in the next iteration to encourage them to be 0, thus promoting low-rank. While for the estimated large singular values, it uses small weights in the next iteration to allow them to be nonzero. This reweighting mechanism is the same as that of reweighted ℓ_1 [95] in sparse recovery, which often achieves superior performance than the unweighted one. This has also been adopted in [51], [94] for promoting low-rank, and outperforms the unweighted nuclear norm.

3.4 Empirical Studies

In this section, we first empirically study and compare the proposed methods with the state-of-the-art methods on the simulated data. Then we demonstrate the effectiveness of the proposed methods on two applications. The methods we compared are AltProj [96], IR-PCP [51], [94], Capped Norm [97], PCP [39]–[41], SPCP [44], [98], DECOLOR [38], our previous SRPCP method [86], and He's implicit regularizer (GAPG_Welsch) [36] that corre-

sponds to Welsch M-estimation. We also compare a variant of PCP, termed PCP_LogSum, which replaces the ℓ_1-norm by the Log-sum function as in [43]. Note that the last six methods all use nuclear norm for the low-rank term. So the comparison of our newly proposed SRPCP with them can clearly demonstrate the effectiveness of various regularizations in dealing with the outliers. We additionally compare with the nuclear norm based matrix completion [99], [100], where the locations of the outliers are known, and purely observe outlier-free entries. This serves as an oracle solution, and can be viewed as a performance upper bound for nuclear norm based robust PCA methods. IR-PCP replaces the nuclear norm in PCP by reweighted nuclear norm (the same as IR-SRPCP) in the iterations. The comparison of our proposed IR-SRPCP with the IR-PCP method can also demonstrate the benefits of using our framework.

Complexity comparision: Assume $n_1 \leq n_2$. The fast Robust PCA method AltProj has complexity $\mathcal{O}(n_1 n_2 r_{max}^2 \log(\frac{1}{\epsilon}))$, where r_{max} is an upper bound for the true rank that needs to be specified. The complexities of PCP, SPCP, and the M-estimator are all $\mathcal{O}(n_1^2 n_2 \frac{1}{\epsilon})$, which is the per-iteration complexity of SRPCP, SRPCP_previous, DECOLOR, Capped Norm, and PCP_LogSum. Both IR-PCP and IR-SRPCP have complexity $\mathcal{O}(n_1^2 n_2 \log(\frac{1}{\epsilon}))$ per iteration.

3.4.1 Comparison on Simulated Data

Our first experimental setup is similar to [97], [98], which is as follows:

1) Given the rank r, the low-rank component $\boldsymbol{L}_0$ is built as $\boldsymbol{L}_0 = \boldsymbol{A}\boldsymbol{B}^T$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are randomly generated $n \times r$ standard Gaussian matrices;

2) Given the fraction ρ (corruption rate) of non-zero entries in $\boldsymbol{E}_0$, the support of $\boldsymbol{E}_0$ is chosen uniformly at random with size ρn^2, and the value of each non-zero entry is independently drawn from a uniform distribution over the interval $[-100, 100]$;

3) Each entry of the noise $\boldsymbol{N}$ is independently drawn from a Gaussian distribution with mean 0 and variance σ^2.

4) Finally, generate $\boldsymbol{M} = \boldsymbol{L}_0 + \boldsymbol{E}_0 + \boldsymbol{N}$. $\boldsymbol{L}_0$ and $\boldsymbol{E}_0$ from $\boldsymbol{M}$ are estimated using different methods.

For each $r \in \{1 : 40\}$, and each $\rho \in \{0.01 : 0.01 : 0.50\}$, we repeat the above procedure 10 times for $\sigma = 0.01$ and 0.1 respectively, and repeat 100 times for the noiseless case ($\sigma = 0$). We fix $n = 100$, $\lambda = 1/\sqrt{n}$, and set $\gamma = 40$ for both IR-SRPCP and IR-PCP, $\beta = 2$ for SRPCP_previous, SRPCP, and IR-SRPCP, such that the threshold $\frac{\beta}{\lambda} = 20$. The parameters of other methods are carefully tuned. For evaluation, the estimated $\hat{\boldsymbol{L}}$ is compared with the ground truth via the Relative Error $\frac{\|\hat{\boldsymbol{L}}-\boldsymbol{L}_0\|_F}{\|\boldsymbol{L}_0\|_F}$. We report the average Relative Error over all trials in the log scale as in [94], i.e., $2\log(\text{Average}(\frac{\|\hat{\boldsymbol{L}}-\boldsymbol{L}_0\|_F}{\|\boldsymbol{L}_0\|_F}))$. Additionally, since we are interested in exact recovery in the noiseless case, we measure the percentage of exact recovery over 100 trials. Due to the limitation of precision, we consider exact recovery to be when the Relative Error is less than 10^{-5}.

a) Noisy case

Fig. 3.1 shows the average Relative Error of different methods in the log scale when $\sigma = 0.1$. Note that in the color scale bar, 0 means $2\log(\text{Average}(\frac{\|\hat{\boldsymbol{L}}-\boldsymbol{L}_0\|_F}{\|\boldsymbol{L}_0\|_F})) = 0$, i.e., the average Relative Error is 1. The red color therefore indicates very poor recovery. Similarly, -2 means the average Relative Error is 10^{-1}, indicated by the green color. First, note that PCP does show its stability against the noise, as we proved. Capped Norm and SRPCP_previous have performance very similar to PCP. Compared with these three methods, PCP_LogSum and DECOLOR can tolerate slightly more fractions of outliers when the rank is low, while tolerate less outliers when the rank is high. On the contrary, the M-estimator performs worse than PCP when the rank is low, only slightly better when the rank is high. SPCP and AltProj do not perform better than PCP. Compared with these above mentioned methods, our newly proposed SRPCP has better performance and can tolerate more outliers. Its performance is relatively closer to the oracle solution, i.e., Matrix Completion. The superior performance of Matrix Completion agrees with the intuition that correcting erasures with known locations

is easier than correcting errors with unknown locations [90]. In Robust PCA, there is no knowledge of the locations and values of the outliers/errors. While in matrix completion, the locations of the erasures are known. Finally, comparing IR-SRPCP with IR-PCP, we can clearly see the improvement achieved using our regularization framework.

The conclusion for the case $\sigma = 0.01$ is very similar, except that the advantage of utilizing iterative reweighted nuclear norm becomes clear, which can be found in the supplemental material. We have also tested smaller magnitude of outlier corruptions (e.g., drawn from U$[-20, 20]$), and the difference in performance between the methods becomes smaller, but the relative performance remains unchanged.

In the above benchmark setting, the generated outlier corruptions ($\sim U[-100, 100]$) are balanced on either side of the low-rank subspace to some extent[3]. We further test the case where the outlier corruptions are single-sided, i.e., drawn from $U[0, 100]$, and noise $\sigma = 0.1$. The corresponding results are shown in Fig. 3.2. We can see that the performances of most methods degrade in this more challenging case. The superiority of the proposed methods becomes more clear. Note that DECOLOR uses the same optimization scheme as SRPCP, except that it uses Frobenius norm instead of the ℓ_1-norm in Step 1 of Algorithm 1. This is exactly the ℓ_1-norm that makes our SRPCP much more robust than DECOLOR to the unidentified outliers.

b) Noiseless case

Fig. 3.3 shows the percentage of exact recovery over 100 trials for each method when outlier corruptions drawn from $U[-100, 100]$. Recall that a trial is declared success if the Relative Error is less than 10^{-5}. First, we find that DECOLOR, PCP_LogSum, and the M-estimator never have 100 percent exact recovery in the experiments. For other methods, there is a large region in which the recovery is exact. SRPCP_previous has larger success region than PCP, while it is smaller than the new SRPCP method proposed in this chapter. The

[3]We thank a reviewer for this observation and suggestion to use single-sided outliers.

IR-PCP has much larger success region than PCP, owing to the iterative reweighted nuclear norm. Our proposed IR-SRPCP has even larger success region than IR-PCP by tolerating more fraction of outliers. When the rank is 5, PCP has 100 percent exact recovery up to 4% corruption rate, and it is 4%, 10%, 17%, 31%, 46% for Capped Norm, SRPCP_previous, SRPCP, IR-PCP, and IR-SRPCP, respectively. The oracle solution has much larger success region than all the nuclear norm based robust PCA methods, as observed in [39].

3.4.2 Comparison on Text Removal

In this subsection, we follow [57] to conduct a text removal image processing simulation, where the results are directly visible. The ground truth low-rank clean image is a 256×256 matrix with rank equal to 10, whose values are between -1 and 1. We embed black text in the image, where the values of the text are randomly drawn from $\mathrm{U}[-1, 0]$. The text can be viewed as sparse outliers. We fix $\lambda = 1/\sqrt{256}$, and set $\gamma = 10$ for both IR-SRPCP and IR-PCP, $\beta = \lambda/4$ for both SRPCP and IR-SRPCP, such that the threshold $\frac{\beta}{\lambda} = 0.25$. For evaluation, we compare the recovered low-rank matrix with the ground truth via ℓ_2 error, i.e., $\|\hat{\boldsymbol{L}} - \boldsymbol{L}_0\|_F$. As the support (mask) of the text is also of interest, the mask of the text is usually obtained by thresholding the estimated $\hat{\boldsymbol{E}}$. We vary the threshold as in [57] to find the maximum F-measure for each method, where the F-measure is commonly used in pattern recognition and is defined as: 2(precision $\cdot$ recall)/(precision+recall). Fig. 4.3 shows the results of each method, where we additionally compare with the BRMF method proposed in [57]. It can be seen that most methods failed to return a clean low-rank image. Our proposed SRPCP is able to recover a relatively clean low-rank image and performs best in terms of F-measure and ℓ_2 error. DECOLOR also performs well on this task. Our proposed IR-SRPCP performs better than the IR-PCP method.

3.4.3 Comparison on Real Data

Lastly, we compare the performance of the methods on first 200 frames of a surveillance video[4], where each frame is converted to a column vector, and the integer pixel values are scaled to the range [-1,1]. The background over the frames is the low-rank component and the moving objects over the frames can be considered as the sparse component. We fix $\lambda = 1/\sqrt{\max(n_1, n_2)}$, and set $\gamma = 40$ for both IR-SRPCP and IR-PCP, $\beta = \lambda/10$ for both SRPCP and IR-SRPCP, such that the threshold $\frac{\beta}{\lambda} = 0.1$. Fig. 4.4 shows the recovered background (left) and foreground (right) in the first frame. SRPCP and IR-SRPCP successfully separate the foreground with the background. For other methods, we can see that there are some ghosting effects in the recovered backgrounds. Note that the lighting at the top of the video changes over the frames. SRPCP, IR-SRPCP, and the M-estimator consider this as the foreground.

3.5 Conclusions and Future Work

In this chapter, we proposed SRPCP and IR-SRPCP to recover the low-rank component and sparse component from possibly noisy observations. Both methods use the ℓ_0-ℓ_1 regularization framework to deal with the outliers and noise. Theoretical results are provided to support the methods. SRPCP has more performance guarantees than IR-SRPCP, e.g., exact recovery in the noiseless case, while IR-SRPCP demonstrates superior performance in the simulation studies. Empirical studies show that IR-SRPCP actually has much larger exact recovery region than SRPCP. One of future directions is to explore the exact recovery property of IR-SRPCP. Applications on text removal and background modeling further support the efficacy and advantage of the proposed methods.

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

As mentioned before, another future work is to use the same bilinear factorization strategy [89] to approximately solve the Step 1 of IR-SRPCP for large-scale problems. It's also very promising to build the connections between the original solution and the solution obtained via using bilinear factorization.

Chapter 3, in part, is a reprint of the material as it appears in the paper: J. Liu and B. D. Rao, "Robust PCA via ℓ_0-ℓ_1 Regularization," in *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 535-549, 15 Jan.15, 2019. The dissertation author was the primary investigator and author of this paper.

3.6 Appendices

3.6.1 Proof of Theorem 3.1

Proof. The proof is divided into the following three parts: a) monotonic decrease in the objective function prior to convergence, b) convergence in a finite number of steps, and c) local optimality of the cluster point.

a) Strictly decreasing behavior of $J(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$ before convergence

We first define following terms: $\Phi^c_{(k)} = \{(i, j)|\boldsymbol{E}^{(k)}_{i,j} \neq 0\}$,

$$J_{\Phi_{(k)}}(\boldsymbol{L}, \boldsymbol{E}) \triangleq ||\boldsymbol{L}||_* + \sum_{(i,j)\in\Phi_{(k)}} \beta||\boldsymbol{E}_{i,j}||_0 + \lambda|(\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E})_{i,j}|,$$

$$J_{\Phi^c_{(k)}}(\boldsymbol{L}, \boldsymbol{E}) \triangleq \sum_{(i,j)\in\Phi^c_{(k)}} \beta||\boldsymbol{E}_{i,j}||_0 + \lambda|(\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E})_{i,j}|,$$

So $J(\boldsymbol{L}, \boldsymbol{E}) = J_{\Phi_{(k)}}(\boldsymbol{L}, \boldsymbol{E}) + J_{\Phi^c_{(k)}}(\boldsymbol{L}, \boldsymbol{E})$.

For any $(i, j) \in \Phi_{(k)}, \boldsymbol{E}^{(k)}_{i,j} = 0$. Hence $J_{\Phi_{(k)}}(\boldsymbol{L}, \boldsymbol{E}^{(k)}) = ||\boldsymbol{L}||_* + \sum_{(i,j)\in\Phi_{(k)}}(\beta||\boldsymbol{E}^{(k)}_{i,j}||_0 + \lambda|(\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}^{(k)})_{i,j}|) = ||\boldsymbol{L}||_* + \sum_{(i,j)\in\Phi_{(k)}} \lambda|(\boldsymbol{M} - \boldsymbol{L})_{i,j}| = ||\boldsymbol{L}||_* + \lambda||\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L})||_1$.

In Step 1, since $\boldsymbol{L}^{(k+1)} \in \arg\min_{\boldsymbol{L}}(||\boldsymbol{L}||_* + \lambda||\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M}-\boldsymbol{L})||_1)$, we have

$$J_{\Phi_{(k)}}(\boldsymbol{L}^{(k+1)}, \boldsymbol{E}^{(k)}) \leq J_{\Phi_{(k)}}(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)}) \tag{3.20}$$

where the equality holds if and only if

$$||\boldsymbol{L}^{(k+1)}||_* + \lambda||\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M}-\boldsymbol{L}^{(k+1)})||_1 = ||\boldsymbol{L}^{(k)}||_* + \lambda||\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M}-\boldsymbol{L}^{(k)})||_1. \tag{3.21}$$

In Step 2, $J_{\Phi_{(k)}}(\boldsymbol{L}^{(k+1)}, \boldsymbol{E}^{(k+1)}) \doteq ||\boldsymbol{L}^{(k+1)}||_* + \sum_{(i,j)\in\Phi_{(k)}}(\beta||\boldsymbol{E}^{(k+1)}_{i,j}||_0 + \lambda|(\boldsymbol{M}-\boldsymbol{L}^{(k+1)}-\boldsymbol{E}^{(k+1)})_{i,j}|) \leq ||\boldsymbol{L}^{(k+1)}||_* + \sum_{(i,j)\in\Phi_{(k)}}(\beta||\boldsymbol{E}^{(k)}_{i,j}||_0 + \lambda|(\boldsymbol{M}-\boldsymbol{L}^{(k+1)}-\boldsymbol{E}^{(k)})_{i,j}|) \doteq J_{\Phi_{(k)}}(\boldsymbol{L}^{(k+1)}, \boldsymbol{E}^{(k)})$, where the inequality is due to $\boldsymbol{E}^{(k+1)}_{i,j} \in \arg\min_{\boldsymbol{E}_{i,j}}(\beta||\boldsymbol{E}_{i,j}||_0 + \lambda|(\boldsymbol{M}-\boldsymbol{L}^{(k+1)}-\boldsymbol{E})_{i,j}|)$ in Step 2.

Utilizing (3.20) we have

$$J_{\Phi_{(k)}}(\boldsymbol{L}^{(k+1)}, \boldsymbol{E}^{(k+1)}) \leq J_{\Phi_{(k)}}(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)}). \tag{3.22}$$

For any $(i,j) \in \Phi^c_{(k)}, \boldsymbol{E}^{(k)}_{i,j} \neq 0$. From Step 2, we know that the upper bound for $J_{\Phi^c_{(k)}}(\boldsymbol{L}^{(t)}, \boldsymbol{E}^{(t)}), t = 1, 2, ...$ is $\beta \times |\Phi^c_{(k)}|$. Hence

$$J_{\Phi^c_{(k)}}(\boldsymbol{L}^{(k+1)}, \boldsymbol{E}^{(k+1)}) \leq \beta \times |\Phi^c_{(k)}| = J_{\Phi^c_{(k)}}(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)}). \tag{3.23}$$

In summary, we have $J(\boldsymbol{L}^{(k+1)}, \boldsymbol{E}^{(k+1)}) \leq J(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$, indicating that the value of the objective function is non-increasing in each iteration. As the objective function is non-negative, it will always converge.

If $J(\boldsymbol{L}^{(k+1)}, \boldsymbol{E}^{(k+1)}) = J(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$, we must have equality hold in (3.20), which implies $\boldsymbol{L}^{(k+1)} = \boldsymbol{L}^{(k)}$ according to (3.21) and Step 1. $\boldsymbol{L}^{(k+1)} = \boldsymbol{L}^{(k)}$ ensures $\boldsymbol{E}^{(k+1)} = \boldsymbol{E}^{(k)}$ and $\Phi_{(k+1)} = \Phi_{(k)}$. Similarly $\Phi_{(k+1)} = \Phi_{(k)}$ implies $\boldsymbol{L}^{(k+2)} = \boldsymbol{L}^{(k+1)}$, and further $\boldsymbol{E}^{(k+2)} = \boldsymbol{E}^{(k+1)}$ and $\Phi_{(k+2)} = \Phi_{(k+1)}$ and so on. So $(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)}) = (\boldsymbol{L}^{(k+1)}, \boldsymbol{E}^{(k+1)}) = (\boldsymbol{L}^{(k+2)}, \boldsymbol{E}^{(k+2)}) = \cdots$, which is a fixed point

of SRPCP.

Thus it follows that the objective function is strictly decreasing before convergence.

b) Convergence in a finite number of iterations

Now, we show that the objective function must converge in a finite number of iterations. As the number of different index sets $\Phi_{(k)}$ is finite (less than $2^{n_1 \times n_2}$), it suffices to show that the same index set will not appear again before the objective function converges. Note that the value of the objective function $J(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$ is determined by $\boldsymbol{L}^{(k)}$ (as $\boldsymbol{E}^{(k)}$ is also determined by $\boldsymbol{L}^{(k)}$ according to Step 2).

We first show that the same index set can not reappear in consecutive iterations before convergence. Suppose $\Phi_{(p)} = \Phi_{(p-1)}$, as $\boldsymbol{L}^{(p)} = \arg\min_{\boldsymbol{L}}(\|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(p-1)}}(\boldsymbol{M} - \boldsymbol{L})\|_1) = \arg\min_{\boldsymbol{L}}(\|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(p)}}(\boldsymbol{M} - \boldsymbol{L})\|_1)$, and $\boldsymbol{L}^{(p+1)} = \arg\min_{\boldsymbol{L}}(\|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(p)}}(\boldsymbol{M} - \boldsymbol{L})\|_1)$, we must have $\|\boldsymbol{L}^{(p+1)}\|_* + \lambda\|\mathcal{P}_{\Phi_{(p)}}(\boldsymbol{M} - \boldsymbol{L}^{(p+1)})\|_1 = \|\boldsymbol{L}^{(p)}\|_* + \lambda\|\mathcal{P}_{\Phi_{(p)}}(\boldsymbol{M} - \boldsymbol{L}^{(p)})\|_1$. So the algorithm sets $\boldsymbol{L}^{(p+1)} = \boldsymbol{L}^{(p)}$ in Step 1. Then we must have convergence of the objective function.

Then it remains to show the same index set can not reappear in non-consecutive iterations before convergence. Before convergence, we have $J(\boldsymbol{L}^{(1)}, \boldsymbol{E}^{(1)}) > \cdots > J(\boldsymbol{L}^{(p+1)}, \boldsymbol{E}^{(p+1)}) > \cdots > J(\boldsymbol{L}^{(r)}, \boldsymbol{E}^{(r)}) > J(\boldsymbol{L}^{(r+1)}, \boldsymbol{E}^{(r+1)}) > \cdots$ The corresponding index sets in Step 1 of each iteration are $\Phi_{(0)}, \cdots, \Phi_{(p)}, \cdots, \Phi_{(r-1)}, \Phi_{(r)}, \cdots$ We only need to show that $\Phi_{(r)} \neq \Phi_{(p)}$ for any $r > p + 1$. As proved earlier, any $\boldsymbol{L}^{(r+1)} \in \arg\min_{\boldsymbol{L}}(\|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(r)}}(\boldsymbol{M} - \boldsymbol{L})\|_1)$ ensures $J(\boldsymbol{L}^{(r+1)}, \boldsymbol{E}^{(r+1)}) \leq J(\boldsymbol{L}^{(r)}, \boldsymbol{E}^{(r)})$, see (3.20 − 3.23). Suppose $\Phi_{(r)} = \Phi_{(p)}$, then for any $\boldsymbol{L}^{(p+1)} \in \arg\min_{\boldsymbol{L}}(\|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(p)}}(\boldsymbol{M} - \boldsymbol{L})\|_1), \boldsymbol{L}^{(p+1)} \in \arg\min_{\boldsymbol{L}} \|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(r)}}(\boldsymbol{M} - \boldsymbol{L})\|_1$, thus $J(\boldsymbol{L}^{(p+1)}, \boldsymbol{E}^{(p+1)}) \leq J(\boldsymbol{L}^{(r)}, \boldsymbol{E}^{(r)})$, which is contradictory to $J(\boldsymbol{L}^{(p+1)}, \boldsymbol{E}^{(p+1)}) > J(\boldsymbol{L}^{(r)}, \boldsymbol{E}^{(r)})$.

c) Convergence to a local optimum

We now prove that when $J(\boldsymbol{L}, \boldsymbol{E})$ converges ($J(\boldsymbol{L}^{(k+1)}, \boldsymbol{E}^{(k+1)}) = J(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$), $(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$ is a local optimum.

Let $(\Delta \boldsymbol{L}, \Delta \boldsymbol{E})$ be a small deformation matrix around $(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$. Then

$$J(\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}, \boldsymbol{E}^{(k)} + \Delta \boldsymbol{E}) = ||\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}||_* + \beta ||\boldsymbol{E}^{(k)} + \Delta \boldsymbol{E}||_0 + \lambda ||\boldsymbol{M} - (\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}) - (\boldsymbol{E}^{(k)} + \Delta \boldsymbol{E})||_1 \tag{3.24}$$

Next, we will show that $J(\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}, \boldsymbol{E}^{(k)} + \Delta \boldsymbol{E}) \geq J(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$ as long as $||\Delta \boldsymbol{E}||_1$ is small enough, thus $J(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$ is a local optimum.

For the term $\beta ||\boldsymbol{E}^{(k)} + \Delta \boldsymbol{E}||_0$, notice that when $||\Delta \boldsymbol{E}||_1$ is small enough,

$$\beta ||\boldsymbol{E}_{i,j}^{(k)} + \Delta \boldsymbol{E}_{i,j}||_0 = \begin{cases} \beta ||\Delta \boldsymbol{E}_{i,j}||_0, & (i, j) \in \Phi_{(k)} \\ \beta ||\boldsymbol{E}_{i,j}^{(k)}||_0, & \text{otherwise} \end{cases} \tag{3.25}$$

So $\beta ||\boldsymbol{E}^{(k)} + \Delta \boldsymbol{E}||_0 = \beta ||\boldsymbol{E}^{(k)}||_0 + \beta \sum_{(i,j) \in \Phi_{(k)}} ||\Delta \boldsymbol{E}_{i,j}||_0$.

Plug in (3.24), we have

$$\begin{aligned}
& J(\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}, \boldsymbol{E}^{(k)} + \Delta \boldsymbol{E}) \\
=& ||\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}||_* + \beta ||\boldsymbol{E}^{(k)}||_0 + \beta \sum_{(i,j) \in \Phi_{(k)}} ||\Delta \boldsymbol{E}_{i,j}||_0 + \lambda ||\boldsymbol{M} - (\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}) - (\boldsymbol{E}^{(k)} + \Delta \boldsymbol{E})||_1 \\
\geq& ||\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}||_* + \beta ||\boldsymbol{E}^{(k)}||_0 + \beta \sum_{(i,j) \in \Phi_{(k)}} ||\Delta \boldsymbol{E}_{i,j}||_0 + \lambda \sum_{(i,j) \in \Phi_{(k)}} |(\boldsymbol{M} - \boldsymbol{L}^{(k)} - \Delta \boldsymbol{L} - \boldsymbol{E}^{(k)} - \Delta \boldsymbol{E})_{i,j}| \\
\overset{(a)}{=}& ||\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}||_* + \beta ||\boldsymbol{E}^{(k)}||_0 + \beta \sum_{(i,j) \in \Phi_{(k)}} ||\Delta \boldsymbol{E}_{i,j}||_0 + \lambda \sum_{(i,j) \in \Phi_{(k)}} |(\boldsymbol{M} - \boldsymbol{L}^{(k)} - \Delta \boldsymbol{L} - \Delta \boldsymbol{E})_{i,j}| \\
\geq& ||\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}||_* + \beta ||\boldsymbol{E}^{(k)}||_0 + \beta \sum_{(i,j) \in \Phi_{(k)}} ||\Delta \boldsymbol{E}_{i,j}||_0 + \lambda \sum_{(i,j) \in \Phi_{(k)}} |(\boldsymbol{M} - \boldsymbol{L}^{(k)} - \Delta \boldsymbol{L})_{i,j}| - \lambda \sum_{(i,j) \in \Phi_{(k)}} |\Delta \boldsymbol{E}_{i,j}| \\
=& ||\boldsymbol{L}^{(k)} + \Delta \boldsymbol{L}||_* + \beta ||\boldsymbol{E}^{(k)}||_0 + \lambda ||\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L}^{(k)} - \Delta \boldsymbol{L})||_1 + \beta \sum_{(i,j) \in \Phi_{(k)}} (||\Delta \boldsymbol{E}_{i,j}||_0 - \frac{\lambda}{\beta} |\Delta \boldsymbol{E}_{i,j}|) \\
\overset{(b)}{\geq}& ||\boldsymbol{L}^{(k+1)}||_* + \beta ||\boldsymbol{E}^{(k)}||_0 + \lambda ||\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L}^{(k+1)})||_1 + \beta \sum_{(i,j) \in \Phi_{(k)}} (||\Delta \boldsymbol{E}_{i,j}||_0 - \frac{\lambda}{\beta} |\Delta \boldsymbol{E}_{i,j}|) \\
\overset{(c)}{=}& ||\boldsymbol{L}^{(k)}||_* + \beta ||\boldsymbol{E}^{(k)}||_0 + \lambda ||\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L}^{(k)})||_1 + \beta \sum_{(i,j) \in \Phi_{(k)}} (||\Delta \boldsymbol{E}_{i,j}||_0 - \frac{\lambda}{\beta} |\Delta \boldsymbol{E}_{i,j}|)
\end{aligned}$$

$$\overset{(d)}{=}||\boldsymbol{L}^{(k)}||_* + \beta||\boldsymbol{E}^{(k)}||_0 + \lambda||\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{M} - \boldsymbol{L}^{(k)} - \boldsymbol{E}^{(k)})||_1 + \beta \sum_{(i,j)\in\Phi_{(k)}} (||\Delta\boldsymbol{E}_{i,j}||_0 - \frac{\lambda}{\beta}|\Delta\boldsymbol{E}_{i,j}|)$$

$$\overset{(e)}{=}||\boldsymbol{L}^{(k)}||_* + \beta||\boldsymbol{E}^{(k)}||_0 + \lambda||\boldsymbol{M} - \boldsymbol{L}^{(k)} - \boldsymbol{E}^{(k)}||_1 + \beta \sum_{(i,j)\in\Phi_{(k)}} (||\Delta\boldsymbol{E}_{i,j}||_0 - \frac{\lambda}{\beta}|\Delta\boldsymbol{E}_{i,j}|)$$

$$= J(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)}) + \beta \sum_{(i,j)\in\Phi_{(k)}} (||\Delta\boldsymbol{E}_{i,j}||_0 - \frac{\lambda}{\beta}|\Delta\boldsymbol{E}_{i,j}|),$$

where step (a) and (d) follow from the fact that $\boldsymbol{E}^{(k)}_{i,j} = 0, \forall(i,j) \in \Phi_{(k)}$, step (b) is from our Step 1. Step (c) is from (3.21), since we must have equality holds in (3.20) upon convergence. Step (e) is from our Step 2.

As long as $||\Delta\boldsymbol{E}||_1$ is small enough (then $|\Delta\boldsymbol{E}_{i,j}|$ is also small enough), we will have $\sum_{(i,j)\in\Phi_{(k)}}(||\Delta\boldsymbol{E}_{i,j}||_0 - \frac{\lambda}{\beta}|\Delta\boldsymbol{E}_{i,j}|) \geq 0$, and thus $J(\boldsymbol{L}^{(k)} + \Delta\boldsymbol{L}, \boldsymbol{E}^{(k)} + \Delta\boldsymbol{E}) \geq J(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$. So $(\boldsymbol{L}^{(k)}, \boldsymbol{E}^{(k)})$ is a local optimum.

3.6.2 Proof of Theorem 3.2

Proof. First, as we have mentioned in Section II, our Step 1 gives the same solution of $\boldsymbol{L}$ as the following problem: $\min_{\boldsymbol{L},\boldsymbol{E}} ||\boldsymbol{L}||_* + \lambda||\boldsymbol{E}||_1$, $s.t.\ \mathcal{P}_{\Phi_{(k)}}\boldsymbol{M} = \mathcal{P}_{\Phi_{(k)}}(\boldsymbol{L} + \boldsymbol{E})$, which is analyzed in [90].

In Step 1 of the first iteration, SRPCP solves $\min_{\boldsymbol{L},\boldsymbol{E}} ||\boldsymbol{L}||_* + \lambda||\boldsymbol{E}||_1$, $s.t.\ \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{E}$. Under the given condition, it recovers $\boldsymbol{L}_0$ exactly according to Theorem 3 of [90], i.e., $\boldsymbol{L}^{(1)} = \boldsymbol{L}_0$. Then in Step 2, since $\boldsymbol{M} - \boldsymbol{L}^{(1)} = \boldsymbol{M} - \boldsymbol{L}_0 = \boldsymbol{E}_0$, for any $\beta > 0$, we have $\text{supp}(\boldsymbol{E}^{(1)}) \subseteq \text{supp}(\boldsymbol{E}_0)$ and $\Phi^c_{(1)} = \{(i,j) : |(\boldsymbol{E}_0)_{i,j}| > \frac{\beta}{\lambda}\}$.

In Step 1 of the second iteration, SRPCP is equivalent to solving $\min_{\boldsymbol{L},\boldsymbol{E}} ||\boldsymbol{L}||_* + \lambda||\boldsymbol{E}||_1$, $s.t.\ \mathcal{P}_{\Phi_{(1)}}\boldsymbol{M} = \mathcal{P}_{\Phi_{(1)}}(\boldsymbol{L} + \boldsymbol{E})$. Where the large outlier entries indexed by $\Phi^c_{(1)}$ become erased entries, but the condition for exact recovery required by Theorem 3 of [90] still holds. So it recovers $\boldsymbol{L}_0$ exactly again, i.e., $\boldsymbol{L}^{(2)} = \boldsymbol{L}_0$. Then SRPCP converges in 2 iterations and finds $\boldsymbol{L}_0$ exactly.

If additionally $\beta < \lambda \min\{|(\boldsymbol{E}_0)_{i,j}| : (\boldsymbol{E}_0)_{i,j} \neq 0\}$, $\boldsymbol{L}^{(1)} = \boldsymbol{L}^{(2)} = \boldsymbol{L}_0$ will ensure $\boldsymbol{E}^{(1)} = \boldsymbol{E}^{(2)} = \boldsymbol{E}_0$ according to Step 2. So SRPCP recovers both $\boldsymbol{L}_0$ and $\boldsymbol{E}_0$ exactly.

3.6.3 Proof of Theorem 3.4

Proof. As mentioned before, in Step 1, the subproblem is equivalent to the following problem:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{E}\|_1, \, s.t. \, \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{E}, \tag{3.26}$$

where in the first iteration, the index set $\Phi_{(0)}$ is all the entries. To make the proof brief, let us first assume $\text{supp}(\boldsymbol{E}_0)^c \subseteq \Phi_{(k)}$ for any k, and use $\Phi_{(k)}$ in the proof. In the end we will show this assumption holds.

We first introduce some additional notations. For any pair $\boldsymbol{X} = (\boldsymbol{L}, \boldsymbol{E})$, define $\|\boldsymbol{X}\|_\diamond \doteq \|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{E}\|_1$, $\|\boldsymbol{X}\|_F \doteq (\|\boldsymbol{L}\|_F^2 + \|\boldsymbol{E}\|_F^2)^{1/2}$, and define the projection operator $\mathcal{P}_T \times \mathcal{P}_\Omega : (\boldsymbol{L}, \boldsymbol{E}) \mapsto (\mathcal{P}_T\boldsymbol{L}, \mathcal{P}_\Omega\boldsymbol{E})$. Define the subspaces $\Gamma \doteq \{(\boldsymbol{W}, \boldsymbol{W}) | \boldsymbol{W} \in \mathbb{R}^{n\times n}\}$ and $\Gamma^\perp \doteq \{(\boldsymbol{W}, -\boldsymbol{W}) | \boldsymbol{W} \in \mathbb{R}^{n\times n}\}$, and let $\mathcal{P}_\Gamma$ and $\mathcal{P}_{\Gamma^\perp}$ denote their respective projection operators. Finally, for any linear operator $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, we use $\|\mathcal{A}\|$ to denote the operator norm $\sup_{\|\boldsymbol{X}\|_F=1} \|\mathcal{A}\boldsymbol{X}\|_F$.

Firstly, note that $\upsilon\xi < \frac{1}{8}$ guarantees that $\frac{\xi}{1-5\upsilon\xi} < \frac{1-4\xi\upsilon}{\upsilon}$.

Our proof uses two crucial properties of $(\hat{\boldsymbol{L}}, \hat{\boldsymbol{E}})$, which is the optimal solution to (3.26). First, since $\hat{\boldsymbol{L}} + \hat{\boldsymbol{E}} = \boldsymbol{M} = \boldsymbol{L}_0 + \boldsymbol{E}_0 + \boldsymbol{N}$, we have $\hat{\boldsymbol{L}} + \hat{\boldsymbol{E}} - \boldsymbol{L}_0 - \boldsymbol{E}_0 = \boldsymbol{N}$. Second, as $(\boldsymbol{L}_0, \boldsymbol{E}_0 + \boldsymbol{N})$ is also a feasible solution to (3.26) with input $\boldsymbol{M} = \boldsymbol{L}_0 + \boldsymbol{E}_0 + \boldsymbol{N}$, we have

$$\|\hat{\boldsymbol{L}}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}\hat{\boldsymbol{E}}\|_1 \leq \|\boldsymbol{L}_0\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{E}_0 + \boldsymbol{N})\|_1 \leq \|\boldsymbol{L}_0\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{E}_0\|_1 + \lambda\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{N}\|_1. \tag{3.27}$$

Denote $\hat{\boldsymbol{X}} = \boldsymbol{X}_0 + \boldsymbol{H}$, where $\hat{\boldsymbol{X}} = (\hat{\boldsymbol{L}}, \hat{\boldsymbol{E}})$, $\boldsymbol{X}_0 = (\boldsymbol{L}_0, \boldsymbol{E}_0)$, $\boldsymbol{H} = (\boldsymbol{H}_L, \boldsymbol{H}_S) = (\hat{\boldsymbol{L}} - \boldsymbol{L}_0, \hat{\boldsymbol{E}} - \boldsymbol{E}_0)$.

We have

$$\boldsymbol{H}_L + \boldsymbol{H}_S = (\hat{\boldsymbol{L}} - \boldsymbol{L}_0) + (\hat{\boldsymbol{E}} - \boldsymbol{E}_0) = \boldsymbol{N}. \tag{3.28}$$

Write $\boldsymbol{H}^{\Gamma} = \mathcal{P}_{\Gamma}(\boldsymbol{H})$, $\boldsymbol{H}^{\Gamma^{\perp}} = \mathcal{P}_{\Gamma^{\perp}}(\boldsymbol{H})$ for short. We want to bound $\|\boldsymbol{H}\|_F^2$, which can be expanded as

$$\|\boldsymbol{H}\|_F^2 = \|\boldsymbol{H}^{\Gamma}\|_F^2 + \|\boldsymbol{H}^{\Gamma^{\perp}}\|_F^2 = \|\boldsymbol{H}^{\Gamma}\|_F^2 + \|(\mathcal{P}_T \times \mathcal{P}_{\Omega})(\boldsymbol{H}^{\Gamma^{\perp}})\|_F^2 + \|(\mathcal{P}_{T^{\perp}} \times \mathcal{P}_{\Omega^c})(\boldsymbol{H}^{\Gamma^{\perp}})\|_F^2 \tag{3.29}$$

Since (3.28) gives us $\|\boldsymbol{H}^{\Gamma}\|_F = (\|(\boldsymbol{H}_L + \boldsymbol{H}_S)/2\|_F^2 + \|(\boldsymbol{H}_L + \boldsymbol{H}_S)/2\|_F^2)^{1/2} = (\|\boldsymbol{N}/2\|_F^2 + \|\boldsymbol{N}/2\|_F^2)^{1/2} = \|\boldsymbol{N}\|_F/\sqrt{2}$, it suffices to bound the second and third terms on the right-hand-side of (3.29).

A. Bound on the third term of (3.29).

First, since $\mathrm{supp}(\boldsymbol{E}_0)^c \subseteq \Phi_{(k)}$, we have that $\boldsymbol{Q}$ is the subgradient of $\lambda\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{E}\|_1$ at $\boldsymbol{E}_0$ if and only if $\mathcal{P}_{\Omega}(\boldsymbol{Q}) = \lambda\mathcal{P}_{\Phi_{(k)}} sign(\boldsymbol{E}_0)$, $\|\mathcal{P}_{\Omega^c}(\boldsymbol{Q})\|_{\infty} \leq \lambda$. Also, $\boldsymbol{Q} \in \partial\|\boldsymbol{L}_0\|_*$ if and only if [40] $\mathcal{P}_T(\boldsymbol{Q}) = \boldsymbol{U}\boldsymbol{V}^T$, $\|\mathcal{P}_{T^{\perp}}(\boldsymbol{Q})\| \leq 1$.

Following the proof of Theorem 2 of [40] (simply replace $sign(\cdot)$ with $\mathcal{P}_{\Phi_{(k)}} sign(\cdot)$ in their proof), $\upsilon\xi < \frac{1}{8}$ guarantees that there exists a dual $\boldsymbol{Q}$ satisfies

$$\boldsymbol{Q} = \lambda\mathcal{P}_{\Phi_{(k)}} sign(\boldsymbol{E}_0) + \mathcal{P}_{\Omega^c}(\boldsymbol{Q}) = \boldsymbol{U}\boldsymbol{V}^T + \mathcal{P}_{T^{\perp}}(\boldsymbol{Q}), \tag{3.30}$$

and

$$\|\mathcal{P}_{T^{\perp}}(\boldsymbol{Q})\| \leq \upsilon(\frac{\lambda + \xi}{1 - 2\xi\upsilon}) < \upsilon(\frac{\frac{1-4\xi\upsilon}{\upsilon} + \xi}{1 - 2\xi\upsilon}) = 1 - \frac{\upsilon\xi}{1 - 2\xi\upsilon},$$

$$\|\mathcal{P}_{\Omega^c}(\boldsymbol{Q})\|_{\infty} \leq \frac{\xi - \lambda(1 - 4\xi\upsilon)}{1 - 2\xi\upsilon} + \lambda < \frac{\xi - \frac{\xi(1-4\xi\upsilon)}{1-5\xi\upsilon}}{1 - 2\xi\upsilon} + \lambda = \lambda - \frac{\xi^2\upsilon}{(1 - 2\xi\upsilon)(1 - 5\xi\upsilon)}.$$

Given $\upsilon\xi < \frac{1}{8}$, we have $\frac{\upsilon\xi}{1-2\xi\upsilon} > 0$ and $\frac{\xi^2\upsilon}{(1-2\xi\upsilon)(1-5\xi\upsilon)} > 0$, thus $\|\mathcal{P}_{T^{\perp}}(\boldsymbol{Q})\| < 1$ and $\|\mathcal{P}_{\Omega^c}(\boldsymbol{Q})\|_{\infty} < \lambda$.

Following the proof of Proposition 2 of [40], we have that for *any* subgradient $(\boldsymbol{Q}_L, \boldsymbol{Q}_S)$ of the function $\|\boldsymbol{L}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{E}\|_1$ at $(\boldsymbol{L}_0, \boldsymbol{E}_0)$,

$$
\begin{aligned}
&\|\boldsymbol{X}_0 + \boldsymbol{H}^{\Gamma^\perp}\|_\diamond - \|\boldsymbol{X}_0\|_\diamond \doteq \|\boldsymbol{L}_0 + \boldsymbol{H}_L^{\Gamma^\perp}\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}(\boldsymbol{E}_0 + \boldsymbol{H}_S^{\Gamma^\perp})\|_1 - \|\boldsymbol{L}_0\|_* - \lambda\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{E}_0\|_1 \\
&\geq \ < \boldsymbol{Q}_L, \boldsymbol{H}_L^{\Gamma^\perp} > + < \boldsymbol{Q}_S, \boldsymbol{H}_S^{\Gamma^\perp} > \\
&= < \boldsymbol{U}\boldsymbol{V}^T + \mathcal{P}_{T^\perp}(\boldsymbol{Q}_L), \boldsymbol{H}_L^{\Gamma^\perp} > + < \lambda\mathcal{P}_{\Phi_{(k)}} sign(\boldsymbol{E}_0) + \mathcal{P}_{\Omega^c}(\boldsymbol{Q}_S), \boldsymbol{H}_S^{\Gamma^\perp} > \\
&\overset{(a)}{=} < \boldsymbol{Q} - \mathcal{P}_{T^\perp}(\boldsymbol{Q}) + \mathcal{P}_{T^\perp}(\boldsymbol{Q}_L), \boldsymbol{H}_L^{\Gamma^\perp} > + < \boldsymbol{Q} - \mathcal{P}_{\Omega^c}(\boldsymbol{Q}) + \mathcal{P}_{\Omega^c}(\boldsymbol{Q}_S), \boldsymbol{H}_S^{\Gamma^\perp} > \\
&\overset{(b)}{=} < -\mathcal{P}_{T^\perp}(\boldsymbol{Q}) + \mathcal{P}_{T^\perp}(\boldsymbol{Q}_L), \boldsymbol{H}_L^{\Gamma^\perp} > + < -\mathcal{P}_{\Omega^c}(\boldsymbol{Q}) + \mathcal{P}_{\Omega^c}(\boldsymbol{Q}_S), \boldsymbol{H}_S^{\Gamma^\perp} > \\
&= < \mathcal{P}_{T^\perp}(\boldsymbol{Q}_L) - \mathcal{P}_{T^\perp}(\boldsymbol{Q}), \mathcal{P}_{T^\perp}(\boldsymbol{H}_L^{\Gamma^\perp}) > + < \mathcal{P}_{\Omega^c}(\boldsymbol{Q}_S) - \mathcal{P}_{\Omega^c}(\boldsymbol{Q}), \mathcal{P}_{\Omega^c}(\boldsymbol{H}_S^{\Gamma^\perp}) >
\end{aligned}
$$

where step (a) is due to (3.30), step (b) is due to $\boldsymbol{H}_L^{\Gamma^\perp} + \boldsymbol{H}_S^{\Gamma^\perp} = 0$. Now pick $\boldsymbol{Q}_L$ such that $\mathcal{P}_{T^\perp}(\boldsymbol{Q}_L) = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{V}}^T$, where $\mathcal{P}_{T^\perp}(\boldsymbol{H}_L^{\Gamma^\perp}) = \tilde{\boldsymbol{U}}\tilde{\Sigma}\tilde{\boldsymbol{V}}^T$, and pick $\boldsymbol{Q}_S$ such that $\mathcal{P}_{\Omega^c}(\boldsymbol{Q}_S) = \lambda sign(\mathcal{P}_{\Omega^c}(\boldsymbol{H}_S^{\Gamma^\perp}))$ as in [40], we have

$$
\begin{aligned}
\|\boldsymbol{X}_0 + \boldsymbol{H}^{\Gamma^\perp}\|_\diamond - \|\boldsymbol{X}_0\|_\diamond \geq & (1 - \|\mathcal{P}_{T^\perp}(\boldsymbol{Q})\|)\|\mathcal{P}_{T^\perp}(\boldsymbol{H}_L^{\Gamma^\perp})\|_* + (\lambda - \|\mathcal{P}_{\Omega^c}(\boldsymbol{Q})\|_\infty)\|\mathcal{P}_{\Omega^c}(\boldsymbol{H}_S^{\Gamma^\perp})\|_1 \\
&\geq \frac{\upsilon\xi}{1 - 2\xi\upsilon}\|\mathcal{P}_{T^\perp}(\boldsymbol{H}_L^{\Gamma^\perp})\|_* + \frac{\xi^2\upsilon\|\mathcal{P}_{\Omega^c}(\boldsymbol{H}_S^{\Gamma^\perp})\|_1}{(1 - 2\xi\upsilon)(1 - 5\xi\upsilon)} \\
&\overset{(c)}{\geq} \frac{\xi^2\upsilon}{(1 - 2\xi\upsilon)(1 - 5\xi\upsilon)}(\|\mathcal{P}_{T^\perp}(\boldsymbol{H}_L^{\Gamma^\perp})\|_* + \|\mathcal{P}_{\Omega^c}(\boldsymbol{H}_S^{\Gamma^\perp})\|_1) \\
&\overset{(d)}{\geq} \frac{\xi^2\upsilon}{(1 - 2\xi\upsilon)(1 - 5\xi\upsilon)}(\|\mathcal{P}_{T^\perp}(\boldsymbol{H}_L^{\Gamma^\perp})\|_F + \|\mathcal{P}_{\Omega^c}(\boldsymbol{H}_S^{\Gamma^\perp})\|_F) \\
&\geq \frac{\xi^2\upsilon}{(1 - 2\xi\upsilon)(1 - 5\xi\upsilon)}(\|\mathcal{P}_{T^\perp}(\boldsymbol{H}_L^{\Gamma^\perp})\|_F^2 + \|\mathcal{P}_{\Omega^c}(\boldsymbol{H}_S^{\Gamma^\perp})\|_F^2)^{1/2} \\
&= \frac{\xi^2\upsilon}{(1 - 2\xi\upsilon)(1 - 5\xi\upsilon)}\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^c})(\boldsymbol{H}^{\Gamma^\perp})\|_F
\end{aligned}
$$

where step (c) uses the fact that $\upsilon \geq 1$ and $\upsilon\xi < \frac{1}{8}$, so $\frac{\xi^2\upsilon}{(1-2\xi\upsilon)(1-5\xi\upsilon)} \leq \frac{\upsilon\xi}{1-2\xi\upsilon}$. Step (d) uses the

fact that $\|\boldsymbol{M}\|_* \geq \|\boldsymbol{M}\|_F$ and $\|\boldsymbol{M}\|_1 \geq \|\boldsymbol{M}\|_F$ for any matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$.

$$
\begin{aligned}
&\text{So, } \|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^c})(\boldsymbol{H}^{\Gamma^\perp})\|_F \\
&\leq \frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{\xi^2\upsilon}(\|\boldsymbol{X}_0 + \boldsymbol{H}^{\Gamma^\perp}\|_\diamond - \|\boldsymbol{X}_0\|_\diamond) \\
&\overset{(e)}{\leq} \frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{\xi^2\upsilon}(\|\boldsymbol{H}^{\Gamma}\|_\diamond + \|\boldsymbol{X}_0 + \boldsymbol{H}\|_\diamond - \|\boldsymbol{X}_0\|_\diamond) \\
&\overset{(f)}{\leq} \frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{\xi^2\upsilon}(\|\boldsymbol{H}^{\Gamma}\|_\diamond + \lambda\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{N}\|_1) \\
&\leq \frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{\xi^2\upsilon}(\|\boldsymbol{H}^{\Gamma}\|_\diamond + \lambda\sqrt{|\Phi_{(k)}|}\,\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{N}\|_F) \\
&= \frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{\xi^2\upsilon}(\|\boldsymbol{H}^{\Gamma}_L\|_* + \lambda\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{H}^{\Gamma}_S\|_1 + \lambda\sqrt{|\Phi_{(k)}|}\,\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{N}\|_F) \\
&\overset{(g)}{\leq} \frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{\xi^2\upsilon}(\sqrt{n}\|\boldsymbol{H}^{\Gamma}_L\|_F + \lambda\sqrt{|\Phi_{(k)}|}\,\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{H}^{\Gamma}_S\|_F + \lambda\sqrt{|\Phi_{(k)}|}\,\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{N}\|_F) \\
&= \frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{\xi^2\upsilon}(0.5\sqrt{n}\|\boldsymbol{N}\|_F + 0.5\lambda\sqrt{|\Phi_{(k)}|}\,\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{N}\|_F + \lambda\sqrt{|\Phi_{(k)}|}\,\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{N}\|_F) \\
&= \frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{2\xi^2\upsilon}(\sqrt{n}\|\boldsymbol{N}\|_F + 3\lambda\sqrt{|\Phi_{(k)}|}\,\|\mathcal{P}_{\Phi_{(k)}}\boldsymbol{N}\|_F) \\
&\leq \frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{2\xi^2\upsilon}(\sqrt{n} + 3\lambda\sqrt{|\Phi_{(k)}|})\|\boldsymbol{N}\|_F
\end{aligned}
$$

where step (e) is due to $\|\boldsymbol{X}_0 + \boldsymbol{H}\|_\diamond \geq \|\boldsymbol{X}_0 + \boldsymbol{H}^{\Gamma^\perp}\|_\diamond - \|\boldsymbol{H}^{\Gamma}\|_\diamond$, step (f) is from (3.27). Step (g) uses the fact that $\|\boldsymbol{M}\|_* \leq \sqrt{n}\|\boldsymbol{M}\|_F$ for any matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$.

B. Bound on the second term of (3.29).

Let us first show that $\|\mathcal{P}_\Omega\mathcal{P}_T\| \leq \frac{1}{4}$. For any matrix $\boldsymbol{Z} \in \mathbb{R}^{n\times n}$,

$$
\|\mathcal{P}_\Omega\mathcal{P}_T\boldsymbol{Z}\| = \max_{\|\boldsymbol{Y}\|_F=1} \|\mathcal{P}_\Omega\mathcal{P}_T\boldsymbol{Z}\boldsymbol{Y}\|_F \leq \max_{\|\boldsymbol{Y}\|_F=1} \|\mathcal{P}_\Omega\mathcal{P}_T\|\|\boldsymbol{Z}\boldsymbol{Y}\|_F = \|\mathcal{P}_\Omega\mathcal{P}_T\|\|\boldsymbol{Z}\|.
$$

So $\|\mathcal{P}_\Omega\mathcal{P}_T\| \geq \frac{\|\mathcal{P}_\Omega\mathcal{P}_T\boldsymbol{Z}\|}{\|\boldsymbol{Z}\|}$ for any nonzero matrix $\boldsymbol{Z}$. Since the equality can be achieved for $\boldsymbol{Z} = I$, we have

$$
\|\mathcal{P}_\Omega\mathcal{P}_T\| = \max_{\boldsymbol{Z}\neq\boldsymbol{0}} \frac{\|\mathcal{P}_\Omega\mathcal{P}_T\boldsymbol{Z}\|}{\|\boldsymbol{Z}\|} = \max_{\boldsymbol{Z}\neq\boldsymbol{0},\mathcal{P}_T\boldsymbol{Z}\neq\boldsymbol{0}} \frac{\|\mathcal{P}_\Omega\mathcal{P}_T\boldsymbol{Z}\|}{\|\boldsymbol{Z}\|} \overset{(a)}{\leq} \max_{\mathcal{P}_T\boldsymbol{Z}\neq\boldsymbol{0}} \frac{\|\mathcal{P}_\Omega\mathcal{P}_T\boldsymbol{Z}\|}{0.5\|\mathcal{P}_T\boldsymbol{Z}\|} = \max_{\boldsymbol{J}\in T,\boldsymbol{J}\neq\boldsymbol{0}} \frac{\|\mathcal{P}_\Omega\boldsymbol{J}\|}{0.5\|\boldsymbol{J}\|}
$$

$$= \max_{\boldsymbol{J}\in T, \|\boldsymbol{J}\|\le 1} 2\|\mathcal{P}_\Omega \boldsymbol{J}\| \overset{(b)}{\le} 2\upsilon\xi < \frac{1}{4},$$

where step (a) is from $\|\mathcal{P}_T \boldsymbol{Z}\| \le 2\|\boldsymbol{Z}\|$, and step (b) is from the proof of proposition 1 in the Appendix B of [40].

Following the proof of Lemma 6 in [44], we have for *any* pair $\boldsymbol{X} = (\boldsymbol{L}, \boldsymbol{E})$, $\|\mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\Omega)(\boldsymbol{X})\|_F^2 \ge 0.5(\|\mathcal{P}_T(\boldsymbol{L})\|_F^2 + \|\mathcal{P}_\Omega(\boldsymbol{E})\|_F^2 - 2\|\mathcal{P}_\Omega\mathcal{P}_T\|\|\mathcal{P}_T(\boldsymbol{L})\|_F\|\mathcal{P}_\Omega(\boldsymbol{E})\|_F)$. Plug in $\|\mathcal{P}_\Omega\mathcal{P}_T\| < \frac{1}{4}$, and use the inequality $(a^2 + b^2 - 0.5ab) \ge \frac{3}{4}(a^2 + b^2)$, we have $\|\mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\Omega)(\boldsymbol{X})\|_F^2 \ge \frac{3}{8}(\|\mathcal{P}_T(\boldsymbol{L})\|_F^2 + \|\mathcal{P}_\Omega(\boldsymbol{E})\|_F^2) = \frac{3}{8}\|(\mathcal{P}_T \times \mathcal{P}_\Omega)(\boldsymbol{X})\|_F^2$. Putting in Section IV.b of [44], we finally have $\|(\mathcal{P}_T \times \mathcal{P}_\Omega)(\boldsymbol{H}^{\Gamma^\perp})\|_F^2 \le \frac{8}{3}\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^c})(\boldsymbol{H}^{\Gamma^\perp})\|_F^2$.

Combined with (3.29), we have

$$\begin{aligned}
\|\boldsymbol{H}^{\Gamma^\perp}\|_F^2 &\le \frac{11}{3}\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^c})(\boldsymbol{H}^{\Gamma^\perp})\|_F^2 \\
\|\boldsymbol{H}^{\Gamma^\perp}\|_F &\le \sqrt{\frac{11}{3}}\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^c})(\boldsymbol{H}^{\Gamma^\perp})\|_F \\
&\le \sqrt{\frac{11}{12}}\frac{(1-5\xi\upsilon)(1-2\xi\upsilon)}{\xi^2\upsilon}(\sqrt{n} + 3\lambda\sqrt{|\Phi_{(k)}|})\|\boldsymbol{N}\|_F \\
&\doteq C(\xi,\upsilon)(\sqrt{n} + 3\lambda\sqrt{|\Phi_{(k)}|})\|\boldsymbol{N}\|_F
\end{aligned}$$

So, $\|\hat{\boldsymbol{L}} - \boldsymbol{L}_0\|_F \le (\|\hat{\boldsymbol{L}} - \boldsymbol{L}_0\|_F^2 + \|\hat{\boldsymbol{E}} - \boldsymbol{E}_0\|_F^2)^{1/2} = \|\boldsymbol{H}\|_F \le \|\boldsymbol{H}^\Gamma\|_F + \|\boldsymbol{H}^{\Gamma^\perp}\|_F$

$\le \frac{\sqrt{2}}{2}\|\boldsymbol{N}\|_F + C(\xi,\upsilon)(\sqrt{n} + 3\lambda\sqrt{|\Phi_{(k)}|})\|\boldsymbol{N}\|_F = [\frac{\sqrt{2}}{2} + C(\xi,\upsilon)(\sqrt{n} + 3\lambda\sqrt{|\Phi_{(k)}|})]\|\boldsymbol{N}\|_F$.

So in Step 1, we must have

$$\|\boldsymbol{L}^{(k+1)} - \boldsymbol{L}_0\|_F \le [\frac{\sqrt{2}}{2} + C(\xi,\upsilon)(\sqrt{n} + 3\lambda\sqrt{|\Phi_{(k)}|})]\|\boldsymbol{N}\|_F. \tag{3.31}$$

In Step 2, $\boldsymbol{E}^{(k+1)}$ will be a trimmed version of $\boldsymbol{M} - \boldsymbol{L}^{(k+1)}$.

In the first iteration, $|\Phi_{(0)}| = n^2$, $\|\boldsymbol{L}^{(1)} - \boldsymbol{L}_0\|_F \le [\frac{\sqrt{2}}{2} + C(\xi,\upsilon)(\sqrt{n} + 3\lambda n)]\|\boldsymbol{N}\|_F$.

For $\forall (i,j) \in G_{(1)}$, we have

$$|(\boldsymbol{M}-\boldsymbol{L}^{(1)})_{i,j}| = |(\boldsymbol{E}_0+\boldsymbol{N}+\boldsymbol{L}_0-\boldsymbol{L}^{(1)})_{i,j}|$$

$$= |(\boldsymbol{E}_0)_{i,j}+(\boldsymbol{N})_{i,j}+(\boldsymbol{L}_0-\boldsymbol{L}^{(1)})_{i,j}|$$

$$\geq |(\boldsymbol{E}_0)_{i,j}| - ||\boldsymbol{N}||_\infty - ||\boldsymbol{L}_0-\boldsymbol{L}^{(1)}||_\infty$$

$$\geq |(\boldsymbol{E}_0)_{i,j}| - ||\boldsymbol{N}||_F - ||\boldsymbol{L}_0-\boldsymbol{L}^{(1)}||_F$$

$$> \frac{\beta}{\lambda} + [\frac{2+\sqrt{2}}{2} + C(\xi,\upsilon)(\sqrt{n}+3\lambda n)]||\boldsymbol{N}||_F - ||\boldsymbol{N}||_F - ||\boldsymbol{L}_0-\boldsymbol{L}^{(1)}||_F$$

$$\geq \tfrac{\beta}{\lambda}.$$

Then $\boldsymbol{E}^{(1)}_{i,j} \neq 0$ for $\forall(i,j) \in G_{(1)}$ according to Step 2 of SRPCP, thus $G_{(1)} \subseteq \text{supp}(\boldsymbol{E}^{(1)})$.

For $\forall(i,j) \in \text{supp}(\boldsymbol{E}_0)^c$, we have $(\boldsymbol{E}_0)_{i,j} = 0$,

$$|(\boldsymbol{M}-\boldsymbol{L}^{(1)})_{i,j}| = |(\boldsymbol{N})_{i,j}+(\boldsymbol{L}_0-\boldsymbol{L}^{(1)})_{i,j}|$$

$$\leq ||\boldsymbol{N}||_\infty + ||\boldsymbol{L}_0-\boldsymbol{L}^{(1)}||_\infty$$

$$\leq ||\boldsymbol{N}||_F + ||\boldsymbol{L}_0-\boldsymbol{L}^{(1)}||_F$$

$$\leq ||\boldsymbol{N}||_F + [\tfrac{\sqrt{2}}{2} + C(\xi,\upsilon)(\sqrt{n}+3\lambda n)]||\boldsymbol{N}||_F < \tfrac{\beta}{\lambda}.$$

Then $\boldsymbol{E}^{(1)}_{i,j} = 0$ for $\forall(i,j) \in \text{supp}(\boldsymbol{E}_0)^c$ according to Step 2 of SRPCP. So $\text{supp}(\boldsymbol{E}^{(1)}) \subseteq \text{supp}(\boldsymbol{E}_0)$. In sum, we have $G_{(1)} \subseteq \text{supp}(\boldsymbol{E}^{(1)}) \subseteq \text{supp}(\boldsymbol{E}_0)$, which implies $|\Phi_{(1)}| \leq n^2 - |G_{(1)}|$ and $\text{supp}(\boldsymbol{E}_0)^c \subseteq \Phi_{(1)}$.

For the second iteration, since $\text{supp}(\boldsymbol{E}_0)^c \subseteq \Phi_{(1)}$, we can plug $|\Phi_{(1)}| \leq n^2 - |G_{(1)}|$ in (3.31) to get

$$||\boldsymbol{L}^{(2)}-\boldsymbol{L}_0||_F \leq [\frac{\sqrt{2}}{2} + C(\xi,\upsilon)\left(\sqrt{n}+3\lambda\sqrt{n^2-|G_{(1)}|}\right)]||\boldsymbol{N}||_F. \tag{3.32}$$

For $\forall(i,j) \in G_{(2)}$, similar to above, we have

$$|(\boldsymbol{M}-\boldsymbol{L}^{(2)})_{i,j}| = |(\boldsymbol{E}_0+\boldsymbol{N}+\boldsymbol{L}_0-\boldsymbol{L}^{(2)})_{i,j}|$$

$$\geq |(\boldsymbol{E}_0)_{i,j}| - ||\boldsymbol{N}||_F - ||\boldsymbol{L}_0-\boldsymbol{L}^{(2)}||_F$$

$$> \frac{\beta}{\lambda} + [\frac{2+\sqrt{2}}{2} + C(\xi,\upsilon)(\sqrt{n}+3\lambda\sqrt{n^2-|G_{(1)}|})]||\boldsymbol{N}||_F - ||\boldsymbol{N}||_F - ||\boldsymbol{L}_0-\boldsymbol{L}^{(2)}||_F$$

$$\geq \tfrac{\beta}{\lambda}.$$

Then $\boldsymbol{E}^{(2)}_{i,j} \neq 0$ for $\forall(i,j) \in G_{(2)}$ according to Step 2 of SRPCP, thus $G_{(2)} \subseteq \text{supp}(\boldsymbol{E}^{(2)})$.

For $\forall(i,j) \in \text{supp}(\boldsymbol{E}_0)^c$, we have

$|(\boldsymbol{M} - \boldsymbol{L}^{(2)})_{i,j}| = |(\boldsymbol{N})_{i,j} + (\boldsymbol{L}_0 - \boldsymbol{L}^{(2)})_{i,j}|$

$\leq ||\boldsymbol{N}||_F + ||\boldsymbol{L}_0 - \boldsymbol{L}^{(2)}||_F$

$\leq ||\boldsymbol{N}||_F + [\frac{\sqrt{2}}{2} + C(\xi,\upsilon)(\sqrt{n} + 3\lambda\sqrt{n^2 - |G_{(1)}|})]||\boldsymbol{N}||_F$

$< \frac{\beta}{\lambda}$.

Then $\boldsymbol{E}^{(2)}_{i,j} = 0$ for $\forall(i,j) \in \text{supp}(\boldsymbol{E}_0)^c$ according to Step 2 of SRPCP. So $\text{supp}(\boldsymbol{E}^{(2)}) \subseteq \text{supp}(\boldsymbol{E}_0)$. In sum, we have $G_{(2)} \subseteq \text{supp}(\boldsymbol{E}^{(2)}) \subseteq \text{supp}(\boldsymbol{E}_0)$, which implies $|\Phi_{(2)}| \leq n^2 - |G_{(2)}|$ and $\text{supp}(\boldsymbol{E}_0)^c \subseteq \Phi_{(2)}$.

If not converged in the second iteration, in the following iterations, recursively using $|\Phi_{(k)}| \leq n^2 - |G_{(k)}|$ and $\text{supp}(\boldsymbol{E}_0)^c \subseteq \Phi_{(k)}$, like in the second iteration, we get $||\boldsymbol{L}^{(k+1)} - \boldsymbol{L}_0||_F \leq [\frac{\sqrt{2}}{2} + C(\xi,\upsilon)\ (\sqrt{n} + 3\lambda\sqrt{n^2 - |G_{(k)}|})]||\boldsymbol{N}||_F$, $G_{(k+1)} \subseteq \text{supp}(\boldsymbol{E}^{(k+1)}) \subseteq \text{supp}(\boldsymbol{E}_0)$, $|\Phi_{(k+1)}| \leq n^2 - |G_{(k+1)}|$, and $\text{supp}(\boldsymbol{E}_0)^c \subseteq \Phi_{(k+1)}$ for $k = 2, 3, \cdots$.

Finally, it is easy to see that $G_{(1)} \subseteq G_{(2)} \subseteq \cdots$, which implies $|G_{(1)}| \leq |G_{(2)}| \leq \cdots$.

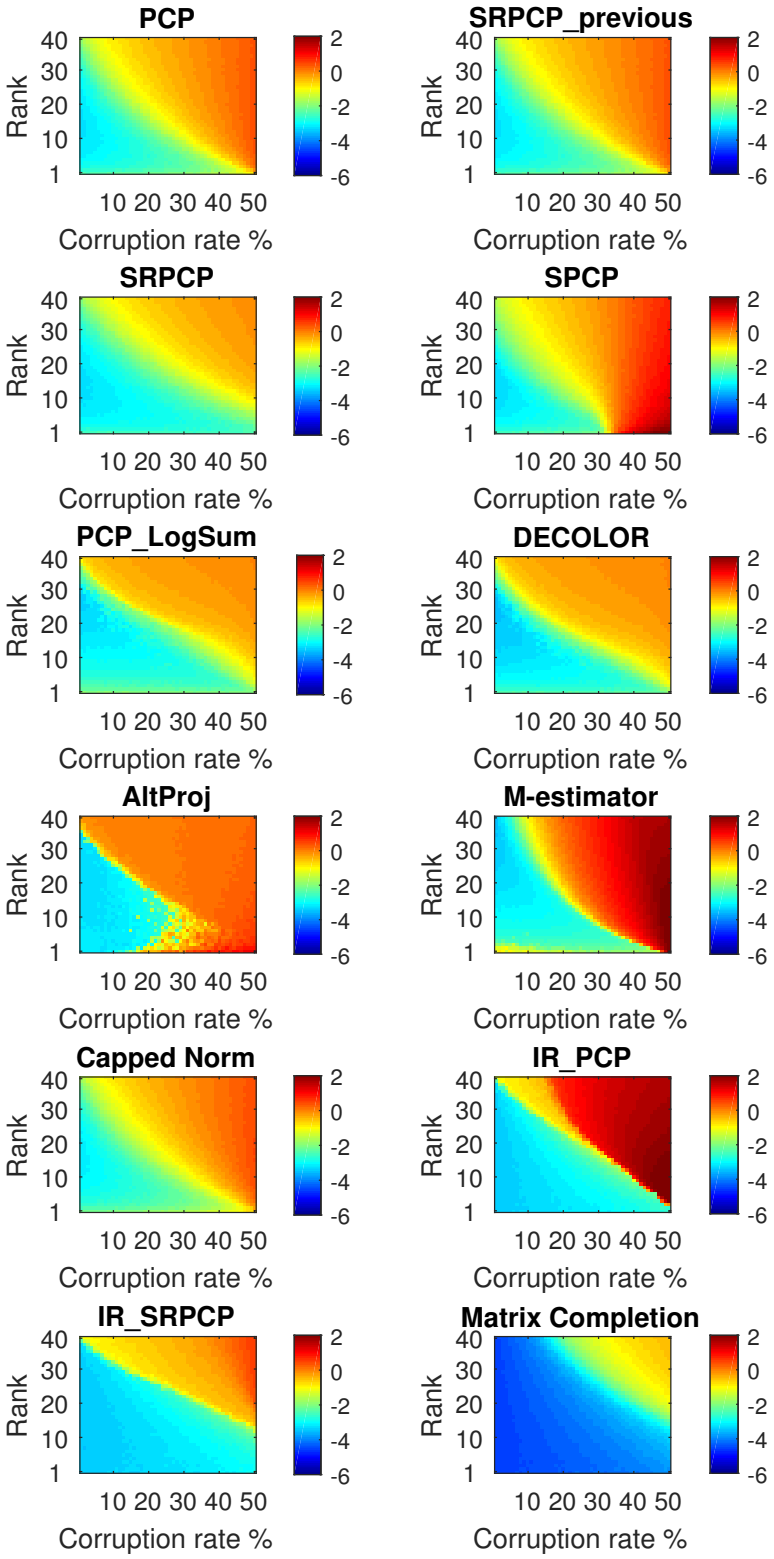

Figure 3.1. Average Relative Error in log scale w.r.t. different rank and corruption rate (corruptions$\sim U[-100, 100]$).

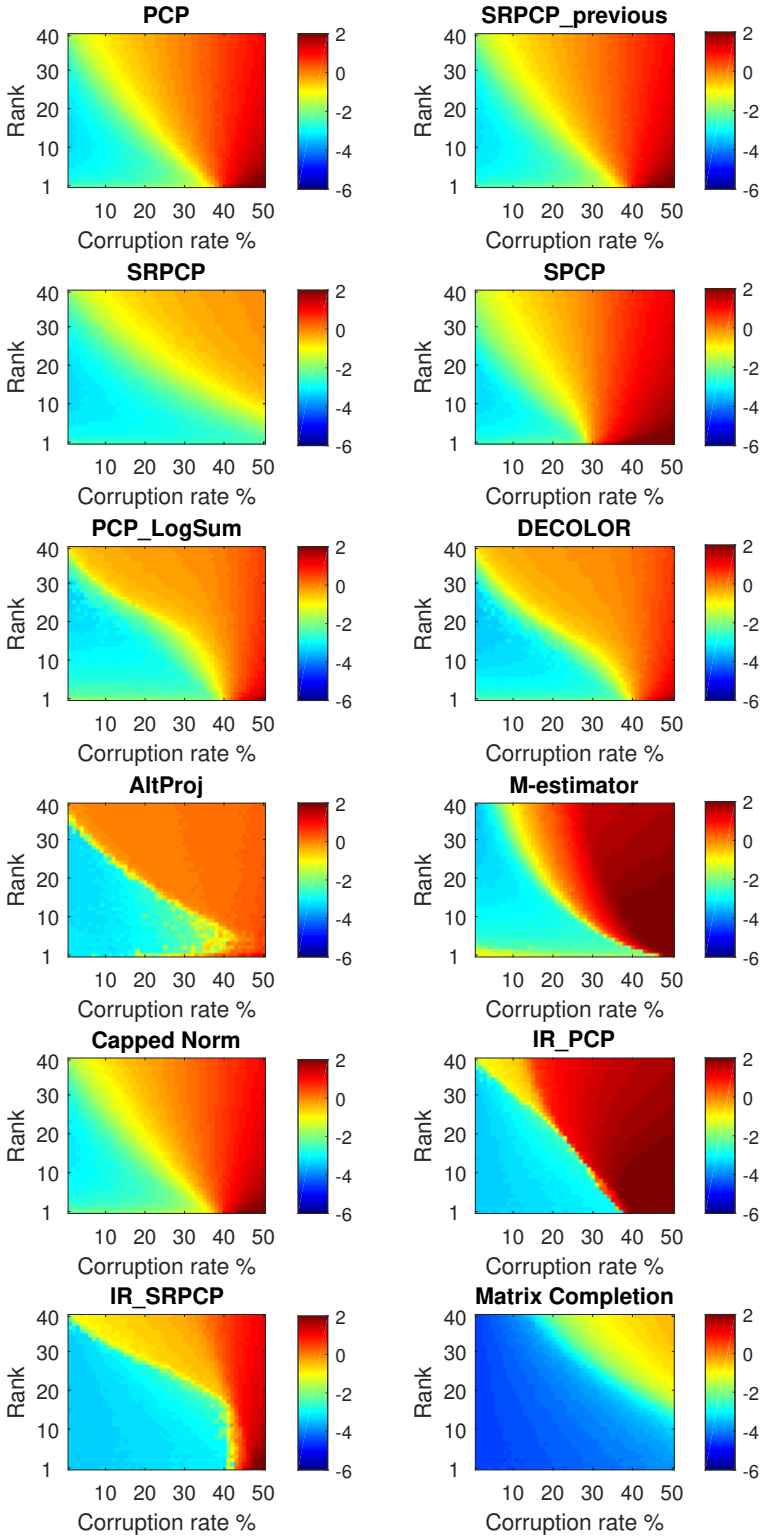

Figure 3.2. Average Relative Error in log scale w.r.t. different rank and corruption rate (corruptions$\sim U[0, 100]$).

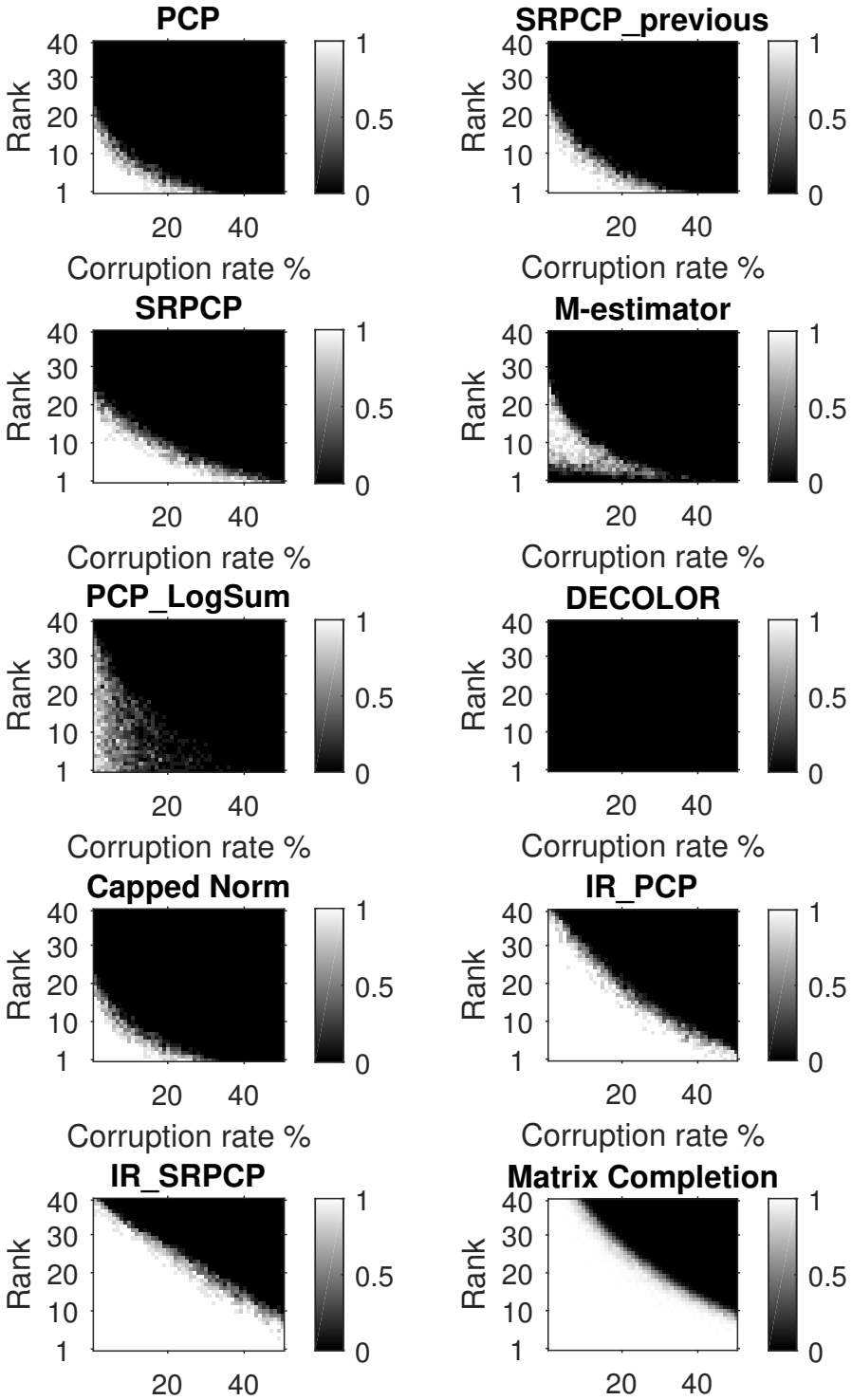

Figure 3.3. Percentage of exact recovery over 100 trials w.r.t. different rank and corruption rate ($\sigma = 0$).

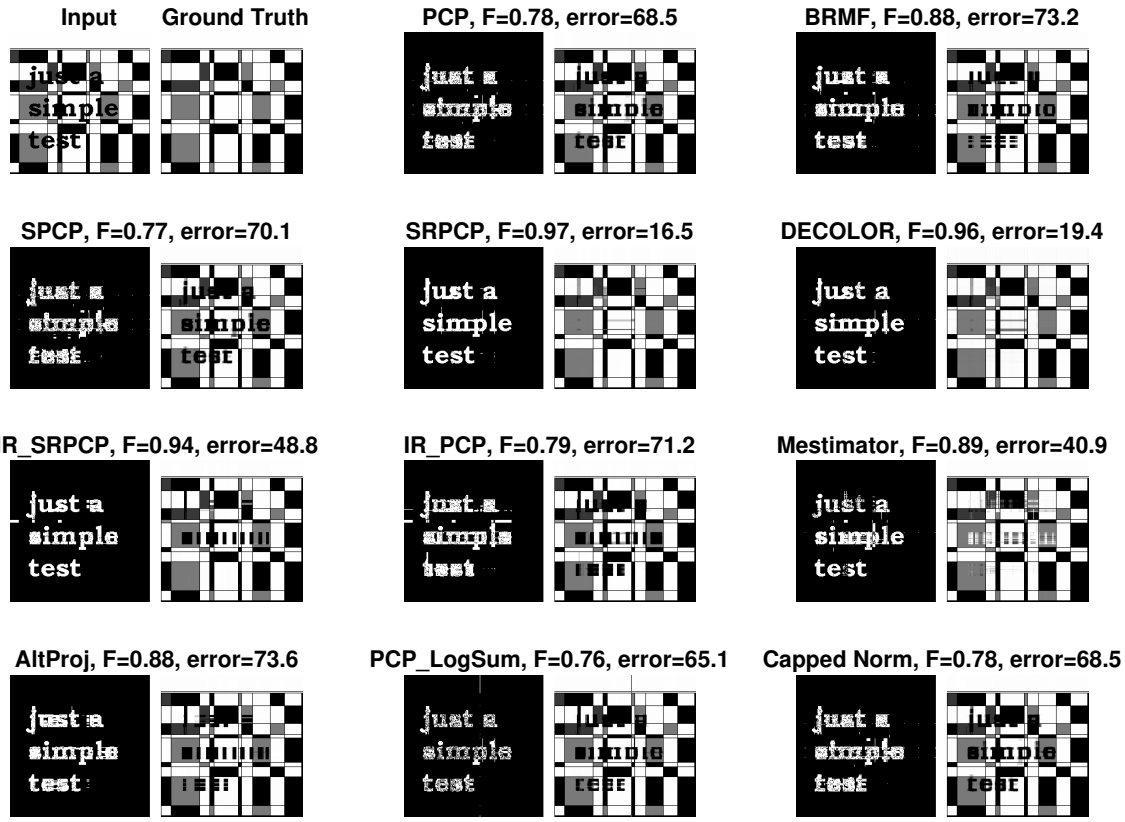

Figure 3.4. Recovered text mask (left, measured by F-measure) and low-rank matrix (right, measured by ℓ_2 error) by each method.

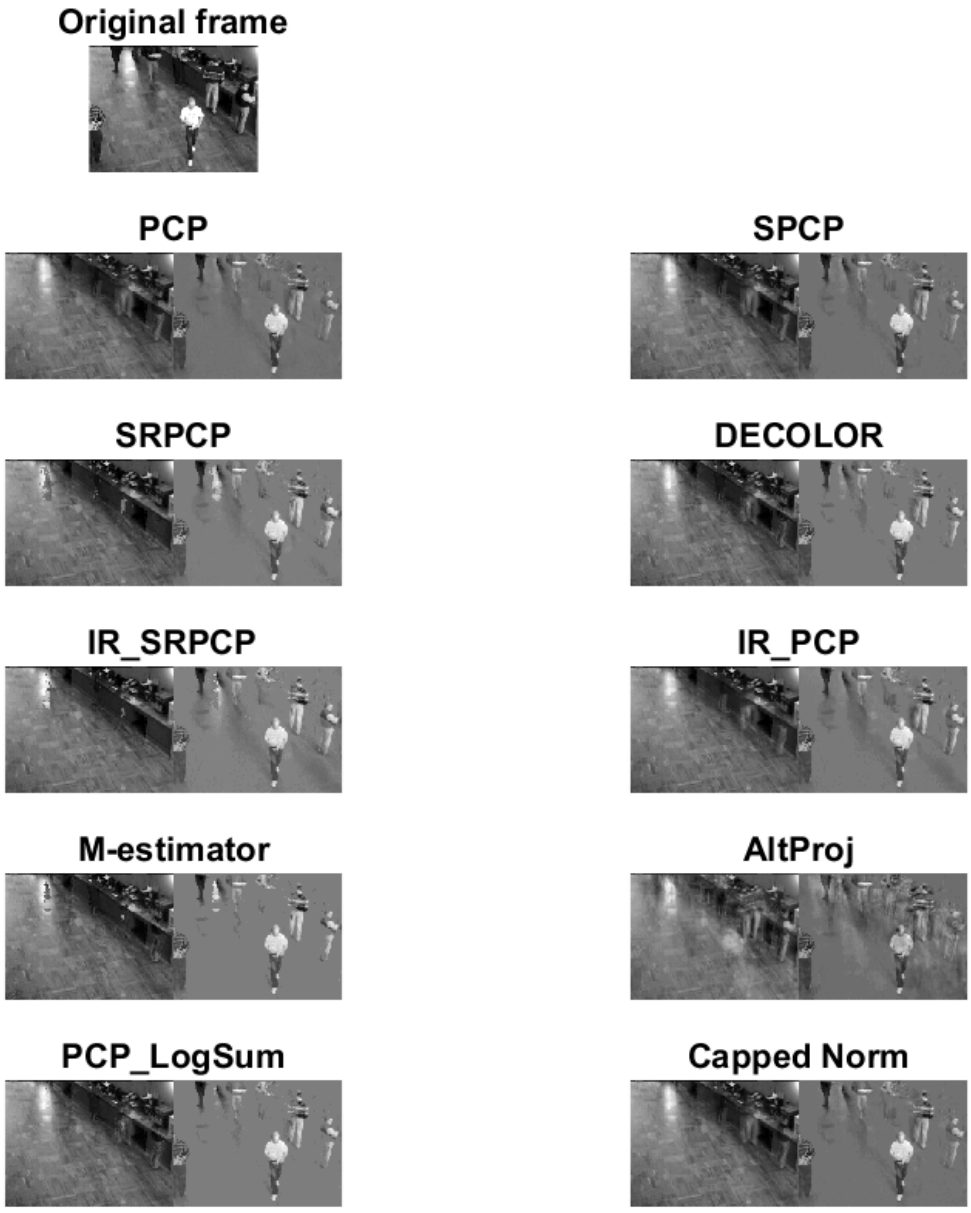

Figure 3.5. Recovered background (left) and foreground (right) by each method.

Chapter 4

Sparse Bayesian Learning for Robust PCA

In this chapter, 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 requirements of the problem. The concise modeling allows simple and effective Empirical Bayesian inference. To further utilize the sparsity pattern information of the outliers in Robust PCA problem, a modification of the above Bayesian method is proposed.

4.1 Introduction

Principal component analysis (PCA) is arguably one of the most widely used data analysis methods with numerous applications. However, its performance can significantly

degrade if the data is corrupted by even a few outliers. As mentioned in a recent review [1], outliers are becoming even more common in today's big data era. The goal of Robust PCA [39] is to recover the low-rank matrix $\boldsymbol{L} \in \mathbb{R}^{n_1 \times n_2}$ and sparse matrix $\boldsymbol{E} \in \mathbb{R}^{n_1 \times n_2}$ (which often models the outlier corruptions) from their composition $\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$ (possibly with additional dense inlier noise). This problem has received a lot of interest in the past decade, with applications ranging from video analysis, face recognition, to recommendation systems. Robust PCA was first studied in the noiseless case[1] [39]–[41]. The optimization problem underlying Robust PCA is [41]:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \text{rank}(\boldsymbol{L}) + \lambda ||\boldsymbol{E}||_0 \quad s.t. \; ||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}||_F \leq \delta, \tag{4.1}$$

where δ is the parameter that is determined by the inlier noise variance. 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 (4.1) is equivalent to the following with $d = \min(n_1, n_2)$:

$$\begin{aligned} \min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{s}\geq 0,\boldsymbol{E}} ||\boldsymbol{s}||_0 + \lambda ||\boldsymbol{E}||_0 \quad & s.t. \; ||\boldsymbol{M} - \boldsymbol{U}\text{diag}(\boldsymbol{s})\boldsymbol{V}^T - \boldsymbol{E}||_F \leq \delta, \\ & \boldsymbol{U} \in \mathbb{R}^{n_1 \times d} \text{ and } \boldsymbol{V} \in \mathbb{R}^{n_2 \times d} \text{ orthonormal.} \end{aligned} \tag{4.2}$$

Further denoting $\boldsymbol{m} = \text{vec}(\boldsymbol{M})$, $\boldsymbol{e} = \text{vec}(\boldsymbol{E})$, 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, (4.2) can be written in the following vector form:

$$\begin{aligned} \min_{\boldsymbol{A},\boldsymbol{s}\geq 0,\boldsymbol{e}} ||\boldsymbol{s}||_0 + \lambda ||\boldsymbol{e}||_0 \; s.t. \; ||\boldsymbol{m} - \boldsymbol{A}\boldsymbol{s} - \boldsymbol{e}||_2 \leq \delta, \; \boldsymbol{A}_i = \text{vec}(\boldsymbol{U}_i\boldsymbol{V}_i^T), \; \forall i, \\ \boldsymbol{U} \in \mathbb{R}^{n_1 \times d} \text{ and } \boldsymbol{V} \in \mathbb{R}^{n_2 \times d} \text{ orthonormal.} \end{aligned} \tag{4.3}$$

[1] Noiseless in this context refers to the absence of inlier noise, standard perturbations, whose density function does not have heavy tails and is usually modeled as standard additive Gaussian noise.

It is known that the optimization problem in (4.1) is NP-hard. To make the problem computationally viable, [39]–[41], [44] 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) [39] in the noiseless case, and Stable Principal Component Pursuit (SPCP)[44] in the noisy case:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{E}\|_1 \quad s.t.\ \|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{E}\|_F \le \delta, \tag{4.4}$$

which is equivalent to

$$\min_{\boldsymbol{A},\boldsymbol{s}\ge 0,\boldsymbol{e}} \|\boldsymbol{s}\|_1 + \lambda\|\boldsymbol{e}\|_1 \quad s.t.\ \|\boldsymbol{m}-\boldsymbol{A}\boldsymbol{s}-\boldsymbol{e}\|_2 \le \delta,\ \boldsymbol{A}_i = \mathrm{vec}(\boldsymbol{U}_i\boldsymbol{V}_i^T),\ \forall i, \tag{4.5}$$
$$\boldsymbol{U} \in \mathbb{R}^{n_1\times \min(n_1,n_2)} \text{ and } \boldsymbol{V} \in \mathbb{R}^{n_2\times \min(n_1,n_2)} \text{ orthonormal.}$$

Interestingly, one can recover both the low-rank matrix and the sparse matrix exactly (or stably) *under certain conditions* by solving (4.4) [39]–[41], [44].

However, viewed from a robust linear regression viewpoint (dealing with the sparse outliers $\boldsymbol{e}$), there is room for improvement. Recent progress [16], [22] in this area shows that the Sparse Bayesian Learning (SBL) [17] approach is quite effective and often 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 [18], [53], [54]. So the question is: can we leverage the advantage of SBL to solve the Robust PCA problem?

In Chapter 3, a genuine ℓ_0-'norm' is used on the sparse matrix $\boldsymbol{E}$ and an algorithm termed Sparsity Regularized Principal Component Pursuit (SRPCP) was proposed to minimize the following objective function:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \|\boldsymbol{L}\|_* + \beta\|\boldsymbol{E}\|_0 + \lambda\|\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{E}\|_1. \tag{4.6}$$

Moreover, it is proved that SRPCP can recover both the low-rank matrix and the sparse matrix exactly in the noiseless case, and stably in the noisy case, under the same conditions required by that of PCP. Empirical results also demonstrate that SRPCP has much better performance than the convex PCP. Though SRPCP manages to use the genuine ℓ_0-'norm' to enforce the sparseness of $\boldsymbol{E}$, it still has to relax the rank minimization objective to the nuclear norm on the low-rank matrix $\boldsymbol{L}$. Recall that the nuclear norm is equivalent to the ℓ_1-norm of the singular values, while the rank function is equivalent to the ℓ_0-'norm' of the singular values. Nevertheless, SRPCP can be served as a good initialization for our Sparse Bayesian Learning method, due to its strong theoretical guarantees. In the following, we will focus on the Bayesian methods. For a comprehensive review on Robust PCA approaches, we refer the interested reader to [42].

There have already been several Sparse Bayesian Learning methods proposed for solving the Robust PCA problem. The earliest work [55] proposed to model the low-rank matrix as $\boldsymbol{L} = \boldsymbol{D}(\text{diag}(\boldsymbol{z})\text{diag}(\boldsymbol{s}))\boldsymbol{W}$, and the sparse matrix as $\boldsymbol{E} = \boldsymbol{B} \circ \boldsymbol{X}$, i.e., $\boldsymbol{M} = \boldsymbol{D}(\text{diag}(\boldsymbol{z})\text{diag}(\boldsymbol{s}))\boldsymbol{W} + \boldsymbol{B} \circ \boldsymbol{X} + \boldsymbol{N}$, 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. [56] 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. Some improvement has been

shown compared to the previous Bayesian approach [55]. However, it is still inferior to the convex PCP approach. Note that this probabilistic model of the low-rank matrix is also adopted in some later works [57]–[59].

Recently, Wipf [60] proposed a even simpler model that directly assumes the columns of $\boldsymbol{L}$ are independent zero-mean Gaussian vectors which share the same covariance matrix, while the sparse matrix is modeled similar to Babacan's work [56]. Slight improvement over the convex PCP method has been empirically demonstrated. In [61], Jansson et al. assume $\text{vec}(\boldsymbol{L})$ is zero-mean Gaussian and its covariance matrix is the Kronecker product of two Wishart distributed matrices. It also demonstrated a slight performance improvement over the PCP method, but the complexity of the inference is very high. Wipf et al. [62] further proposed a modification to the model in [60], which assumes $\text{vec}(\boldsymbol{L})$ is zero-mean Gaussian with covariance matrix obeying Kronecker-sum structure. 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. Nevertheless, the resulting method demonstrates much better performance than the convex PCP method and Bayesian approaches. We will discuss this approach in detail, and show that its objective function implicitly uses additional information of the sparsity pattern of the outliers.

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 has to be used.

In this chapter, we first provide a simple model and derive a concise SBL approach in Section 4.2. The proposed method has minimum assumptions and effectively deals with the

requirements of the problem. It also admits a simple and exact inference. In Section 4.3, we analyze this approach in detail. Motivated by the success of [62], in Section 4.4, we further propose a modified SBL approach that utilizes the sparsity pattern information of the outliers, in a more principled way. Empirical comparisons with the existing methods are in Section 4.5 and the conclusions are made in Section 4.6.

Notation: Throughout this chapter, bold lowercase letters denote vectors, e.g., $\boldsymbol{s}$, while $\boldsymbol{s}_i$ denotes its ith element. Bold capital letters denote matrices, e.g., $\boldsymbol{A}$, where $\boldsymbol{A}^{(k)}$ denotes the updated $\boldsymbol{A}$ in the kth iteration, and $\boldsymbol{A}_i$ denotes the ith column of $\boldsymbol{A}$. Besides that, $\text{vec}(\boldsymbol{A}) \in \mathbb{R}^{n_1 n_2 \times 1}$ is a long vector obtained by stacking the columns of matrix $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_2}$, whereas $\text{Mat}(\boldsymbol{h}) \in \mathbb{R}^{n_1 \times n_2}$ is a matrix obtained by the reverse operation on the vector $\boldsymbol{h} \in \mathbb{R}^{n_1 n_2 \times 1}$. We sometimes use $\langle \cdot \rangle$ to stand for the posterior expectation, and the posterior density involved should be clear from the context.

4.2 Sparse Bayesian Learning Approach

4.2.1 Objective

Before going into the details of our SBL formulation, let us first consider the fundamental problem that our Bayesian approach attempts to solve:

$$\min_{\boldsymbol{A},\boldsymbol{s},\boldsymbol{e}} \|\boldsymbol{m} - \boldsymbol{A}\boldsymbol{s} - \boldsymbol{e}\|_2^2 + \lambda_1 \|\boldsymbol{s}\|_0 + \lambda_2 \|\boldsymbol{e}\|_0 \; s.t. \; \boldsymbol{A}_i = \text{vec}(\boldsymbol{U}_i \boldsymbol{V}_i^T), \; \|\boldsymbol{U}_i\|_2 = \|\boldsymbol{V}_i\|_2 = 1, \forall i, \\ \boldsymbol{U} \in \mathbb{R}^{n_1 \times d}, \boldsymbol{V} \in \mathbb{R}^{n_2 \times d}. \tag{4.7}$$

which is the Lagrange form of

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

$$\boldsymbol{U} \in \mathbb{R}^{n_1 \times d}, \boldsymbol{V} \in \mathbb{R}^{n_2 \times d}. \tag{4.8}$$

Compared with (4.3), 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[2]. More importantly, the following proposition guarantees that this modification does not change the optimal solution in terms of $\boldsymbol{L}(= \boldsymbol{U}\text{diag}(\boldsymbol{s})\boldsymbol{V}^T)$ and $\boldsymbol{E}$.

Proposition 4.1 *Set $d = \min(n_1, n_2)$ in (4.2) and (4.8). Then the optimization problems in (4.1), (4.2) and (4.8) 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 (4.2) and (4.8).*

Proof. Proved as a special case of Proposition 4.2.

As we will see in Section 4.3.4, the complexity of the proposed SBL approach scales with d^2. The following proposition justifies that d can be set to the same order of the rank of the low-rank matrix, which is usually much less than $\min(n_1, n_2)$.

Proposition 4.2 *Set $d \in [rank(\boldsymbol{L}_{opt}),\ \min(n_1, n_2)]$ in (4.2) and (4.8), where $\boldsymbol{L}_{opt}$ is the global optimal solution(s) of (4.1). Then the optimization problems in (4.1), (4.2) and (4.8) 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 (4.2) and (4.8).*

The proof can be found in the Appendix 4.7.2.

The proposed objective function (4.8) and the above propositions establish a connection between Robust PCA and Robust Sparse Linear Regression, and offer a new viewpoint for the Robust PCA problem. Many existing methods and analyses in Robust Sparse Lin-

[2]Note that the SVD can be generalized to make the entries in $\boldsymbol{s}$ real rather than restricting them to be positive. One can deal with the sign by modifying the singular vectors appropriately.

ear Regression (e.g., [63]–[66]) can be leveraged to solve and understand the Robust PCA problem, and vice versa.

4.2.2 SBL Model

Now we present our SBL approach to tackle (4.7). Our observation model is

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

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

Thanks to the removal of the non-negative constraint on $\boldsymbol{s}$ in (4.7), 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[3] (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

$$\boldsymbol{F} = (\boldsymbol{\Gamma}^{-1} + \boldsymbol{A}^T\boldsymbol{D}\boldsymbol{A})^{-1}. \tag{4.9}$$

We have that $\boldsymbol{m}$ is zero mean Gaussian with covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{m}} = \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \boldsymbol{\Lambda} + \beta\boldsymbol{I}$,

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

whose inverse is given by

$$\boldsymbol{\Sigma_m^{-1}} = (\boldsymbol{A\Gamma A}^T + \boldsymbol{\Lambda} + \beta\boldsymbol{I})^{-1} = \boldsymbol{D} - \boldsymbol{DAFA}^T\boldsymbol{D}. \tag{4.10}$$

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

$$\begin{aligned}\boldsymbol{\mu}_{e|m} &= \boldsymbol{\mu}_e + \boldsymbol{\Sigma}_{em}\boldsymbol{\Sigma_m^{-1}}(\boldsymbol{m} - \boldsymbol{\mu}_m) = \boldsymbol{\Lambda\Sigma_m^{-1}m} \\ &= \boldsymbol{\Lambda Dm} - \boldsymbol{\Lambda D}\left(\boldsymbol{A}\left(\boldsymbol{F}\left(\boldsymbol{A}^T(\boldsymbol{Dm})\right)\right)\right).\end{aligned} \tag{4.11}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{e|m} &= \boldsymbol{\Sigma}_e - \boldsymbol{\Sigma}_{em}\boldsymbol{\Sigma_m^{-1}}\boldsymbol{\Sigma}_{me} = \boldsymbol{\Lambda} - \boldsymbol{\Lambda\Sigma_m^{-1}\Lambda} \\ &= \boldsymbol{\Lambda} - \boldsymbol{\Lambda D\Lambda} + \boldsymbol{\Lambda DAFA}^{\boldsymbol{T}}\boldsymbol{D\Lambda}.\end{aligned} \tag{4.12}$$

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

$$\begin{aligned}\boldsymbol{\mu}_{s|m} &= \boldsymbol{\mu}_s + \boldsymbol{\Sigma}_{sm}\boldsymbol{\Sigma_m^{-1}}(\boldsymbol{m} - \boldsymbol{\mu}_m) = \boldsymbol{\Gamma A}^T\boldsymbol{\Sigma_m^{-1}m} \\ &= \boldsymbol{\Gamma A}^T\boldsymbol{Dm} - \boldsymbol{\Gamma}(\boldsymbol{A}^T\boldsymbol{DA})\left(\boldsymbol{F}\left(\boldsymbol{A}^T(\boldsymbol{Dm})\right)\right).\end{aligned} \tag{4.13}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{s|m} =& \boldsymbol{\Sigma}_s - \boldsymbol{\Sigma}_{sm}\boldsymbol{\Sigma_m^{-1}}\boldsymbol{\Sigma}_{ms} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma A}^T\boldsymbol{\Sigma_m^{-1}A\Gamma} \\ =& \boldsymbol{\Gamma} - \boldsymbol{\Gamma}(\boldsymbol{A}^T\boldsymbol{DA})\boldsymbol{\Gamma} + \boldsymbol{\Gamma}(\boldsymbol{A}^T\boldsymbol{DA})\boldsymbol{F}(\boldsymbol{A}^T\boldsymbol{DA})\boldsymbol{\Gamma}.\end{aligned} \tag{4.14}$$

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

$$\boldsymbol{\Sigma}_{se|m} = \boldsymbol{FA}^T(\boldsymbol{I} - \beta\boldsymbol{D}). \tag{4.15}$$

Note that the term $(\boldsymbol{A}^T\boldsymbol{DA})$ in (4.13) and (4.14) has already been calculated in (4.9). If $\boldsymbol{\gamma}, \boldsymbol{\alpha}$

are random, as assumed in the next section, the above statistics can be viewed as conditional statistics, conditioned on $\boldsymbol{\gamma}, \boldsymbol{\alpha}$.

4.2.3 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 $p(\Psi|\boldsymbol{m}) \propto p(\boldsymbol{m}|\Psi)p(\Psi)$. Here we restrict $\boldsymbol{A} \in \mathcal{A}$ 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 [101] procedure to optimize $p(\Psi|\boldsymbol{m})$.

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

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

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

In M-step, the objective function to minimize is $[Q(\Psi|\Psi^{(k)}) - \log p(\Psi)]$. More specifically,

$$
\begin{aligned}
&\min_{\boldsymbol{\gamma},\boldsymbol{\alpha},\boldsymbol{A}\in\mathcal{A}} Q(\boldsymbol{A}, \boldsymbol{\gamma}, \boldsymbol{\alpha}|\boldsymbol{A}^{(k)}, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\alpha}^{(k)}) - \log p(\boldsymbol{\gamma}) \\
=&\min_{\boldsymbol{\gamma},\boldsymbol{\alpha},\boldsymbol{A}\in\mathcal{A}} \frac{1}{2\beta}\|\boldsymbol{m} - \boldsymbol{A}\langle\boldsymbol{s}\rangle - \langle\boldsymbol{e}\rangle\|_2^2 + 2\,\text{Trace}(\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{s}\boldsymbol{e}|\boldsymbol{m}}) + \text{Trace}(\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}\boldsymbol{A}^T) + \text{Trace}(\boldsymbol{\Sigma}_{\boldsymbol{e}|\boldsymbol{m}})
\end{aligned}
$$

$$+\frac{1}{2}\sum_i(\log\boldsymbol{\gamma}_i+\frac{\langle \boldsymbol{s}_i^2\rangle}{\boldsymbol{\gamma}_i})+\frac{1}{2}\sum_i(\log\boldsymbol{\alpha}_i+\frac{\langle \boldsymbol{e}_i^2\rangle}{\boldsymbol{\alpha}_i})+\sum_i((a+1)\log\boldsymbol{\gamma}_i)+const. \tag{4.16}$$

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 $\boldsymbol{\gamma}$: $\boldsymbol{\gamma}_i=\langle \boldsymbol{s}_i^2\rangle/(2a+3)=(\boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}^2(i)+\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(i,i))/(2a+3),\forall i.$

Directly updating the whole matrix $\boldsymbol{A}$ under the constraints $\boldsymbol{A}\in\mathcal{A}$ is difficult. However, we can update each column of $\boldsymbol{A}$ with other columns fixed and still obey the constraints $\boldsymbol{A}\in\mathcal{A}$. To simplify the presentation, the following discussion *assumes* that $|\langle \boldsymbol{s}_1\rangle|\geq|\langle \boldsymbol{s}_2\rangle|\geq\cdots$, and we first update the first column of $\boldsymbol{A}$. The actual order is detailed in Algorithm 1 and will be discussed in Section 4.3.1.

Update $\boldsymbol{A}_1$: Given $\boldsymbol{A}_2^{(k)},\boldsymbol{A}_3^{(k)},\cdots,\boldsymbol{A}_d^{(k)}$,

$$\begin{aligned}\boldsymbol{A}_1^{(k+1)}=\arg\min_{\substack{\boldsymbol{A}_1=\mathrm{vec}(\boldsymbol{U}_1\boldsymbol{V}_1^T)\\ \|\boldsymbol{U}_1\|_2=1\\ \|\boldsymbol{V}_1\|_2=1}}&\|\boldsymbol{m}-\langle \boldsymbol{e}\rangle-\sum_{i=2}^d\langle \boldsymbol{s}_i\rangle\boldsymbol{A}_i^{(k)}-\langle \boldsymbol{s}_1\rangle\boldsymbol{A}_1\|_2^2+2\,\mathrm{Trace}(\boldsymbol{A}_1\boldsymbol{\Sigma}_{\boldsymbol{se}|\boldsymbol{m}}(1,:))\\ &+\mathrm{Trace}\left(\boldsymbol{A}_1\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(1,1)\boldsymbol{A}_1^T+2\sum_{i=2}^d\boldsymbol{A}_i^{(k)}\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{m}}(1,i)\boldsymbol{A}_1^T\right)\end{aligned} \tag{4.17}$$

$$=\arg\min_{\substack{\boldsymbol{A}_1=\mathrm{vec}(\boldsymbol{U}_1\boldsymbol{V}_1^T)\\ \|\boldsymbol{U}_1\|_2=1\\ \|\boldsymbol{V}_1\|_2=1}}\|\boldsymbol{h}-\boldsymbol{A}_1\|_2^2 \tag{4.18}$$

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

At first glance, (4.18) still seems hard to solve. However, utilizing the structure of $\boldsymbol{A}_1$, we can transform this problem to the equivalent matrix form:

$$(\boldsymbol{U}_1^{(k+1)},\boldsymbol{V}_1^{(k+1)})=\arg\min_{\substack{\boldsymbol{U}_1,\boldsymbol{V}_1\\ \|\boldsymbol{U}_1\|_2=1\\ \|\boldsymbol{V}_1\|_2=1}}\|\mathrm{Mat}(\boldsymbol{h})-\boldsymbol{U}_1\boldsymbol{V}_1^T\|_F^2. \tag{4.19}$$

The *optimal* solution is given by the first singular vector pair of $\mathrm{Mat}(\boldsymbol{h})$, and the proof

can be found in the Appendix 4.7.3. Updating $\boldsymbol{U}_1$ and $\boldsymbol{V}_1$ as a pair is inspired by the success of K-SVD [102]. But note that in K-SVD, $\boldsymbol{V}_1$ is not restricted to be unit length.

To update the jth column with other columns fixed, the derivation is similar to updating $\boldsymbol{A}_1$, except that we use the latest updates of the other columns, e.g., $\boldsymbol{A}_1^{(k+1)}$ instead of $\boldsymbol{A}_1^{(k)}$.

The whole algorithm is summarized in Algorithm 1. Note that in Step 1, we fix $\boldsymbol{A}^{(k)}$ and update $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ to certain precision.

4.2.4 Why the Need for an Extra Prior on γ ?

We now discuss the need for as well as the important role the Inverse-gamma prior on $\boldsymbol{\gamma}$, and non-informative prior on $\boldsymbol{\alpha}$ play. The reason becomes clear if we reformulate our observation model as $\boldsymbol{m} = [\boldsymbol{A} \quad \boldsymbol{I}] \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{e} \end{bmatrix} + \boldsymbol{n}$. 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 concatenated vector $\begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{e} \end{bmatrix}$. Note that $\boldsymbol{s}$ has much smaller dimension than $\boldsymbol{e}$. Then there is always a trivial sparse solution with $\boldsymbol{e} = \boldsymbol{0}$ and dense $\boldsymbol{s}$. This is similar to setting $\lambda = 1$ in (4.1). Putting a prior on $\boldsymbol{\gamma}$ is analogous to setting the weight parameter λ in (4.1). In [62], λ is set as $1/\max(n_1, n_2)$ to ensure that both the low-rank matrix term and the sparse matrix term scale between 0 and $\min(n_1, n_2)$. However, this fixed setting appears to only work for certain range of rank and sparsity. Suppose there is knowledge of the true rank and sparsity, a better choice seems to be $\lambda = \min\left(\text{rank}(\boldsymbol{L}_0)/\|\boldsymbol{E}_0\|_0, 1\right)$. For example, if $\text{rank}(\boldsymbol{L}_0)$ is very small and $\|\boldsymbol{E}_0\|_0$ is relative large, the objective function (4.1) with this choice of λ will also encourage the solution $\hat{\boldsymbol{L}}$ to have small rank and the solution $\hat{\boldsymbol{E}}$ to have large sparsity. This motivates our setting of the parameter of the Inverse-gamma prior on $\boldsymbol{\gamma}$, which will be discussed next.

Algorithm 1 Sparse Bayesian Learning for Robust PCA

Input: $\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$, noise variance $\beta > 0$, and Inverse-gamma prior parameter a

Initialize: $\boldsymbol{U}^{(0)} \in \mathbb{R}^{n_1 \times d}$, $\boldsymbol{V}^{(0)} \in \mathbb{R}^{n_2 \times d}$, $\boldsymbol{\gamma}_i^{(0)} = 1, \boldsymbol{\alpha}_i^{(0)} = 1, \forall i,\ k = 0$

While not converged Do

Step 1. Fix $\boldsymbol{A}^{(k)}$, repeat updating $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$ to certain precision:

Calculate $\boldsymbol{\mu}_{s|m}, \boldsymbol{\mu}_{e|m}$, diag($\boldsymbol{\Sigma}_{s|m}$), and diag($\boldsymbol{\Sigma}_{e|m}$) use (4.11),(4.12),(4.13),(4.14);

$\boldsymbol{\alpha}_i = \boldsymbol{\mu}_{e|m}^2(i) + \boldsymbol{\Sigma}_{e|m}(i,i)$;

$\boldsymbol{\gamma}_i = (\boldsymbol{\mu}_{s|m}^2(i) + \boldsymbol{\Sigma}_{s|m}(i,i))/(2a+3)$.

Step 2. Fix $\boldsymbol{\gamma}^{(k+1)}$ and $\boldsymbol{\alpha}^{(k+1)}$, update $\boldsymbol{A}$:

Calculate $\boldsymbol{\Sigma}_{se|m}, \boldsymbol{\mu}_{s|m}, \boldsymbol{\mu}_{e|m}$, and $\boldsymbol{\Sigma}_{s|m}$ use (4.11),(4.13),(4.14),(4.15);

$\langle \boldsymbol{s} \rangle \triangleq \boldsymbol{\mu}_{s|m}, \langle \boldsymbol{e} \rangle \triangleq \boldsymbol{\mu}_{e|m}$;

$\boldsymbol{index}$=sort([$|\langle \boldsymbol{s}_1^{(k+1)} \rangle|, |\langle \boldsymbol{s}_2^{(k+1)} \rangle|, \cdots, |\langle \boldsymbol{s}_d^{(k+1)} \rangle|$], 'descending');

for j=1:d

//update $\boldsymbol{A}_{\boldsymbol{index}(j)}^{(k+1)}$ using $\boldsymbol{A}_{\boldsymbol{index}(i)}^{(k+1)}, i = 1, \cdots, j-1$, and $\boldsymbol{A}_{\boldsymbol{index}(i)}^{(k)}, i = j+1, \cdots, d$.

$j' \triangleq \boldsymbol{index}(j)$;

$$\boldsymbol{h} = \frac{1}{\langle \boldsymbol{s}_{j'} \rangle^2 + \boldsymbol{\Sigma}_{s|m}(j',j')} [\langle \boldsymbol{s}_{j'} \rangle \boldsymbol{m} - \langle \boldsymbol{s}_{j'} \rangle \langle \boldsymbol{e} \rangle - \boldsymbol{\Sigma}_{se|m}^T(j',:) - \sum_{l \in \{\boldsymbol{index}(i): i<j\}} [\langle \boldsymbol{s}_{j'} \rangle \langle \boldsymbol{s}_l \rangle + \boldsymbol{\Sigma}_{s|m}(j',l)] \boldsymbol{A}_l^{(k+1)} - \sum_{l \in \{\boldsymbol{index}(i): i>j\}} [\langle \boldsymbol{s}_{j'} \rangle \langle \boldsymbol{s}_l \rangle + \Sigma_{s|m}(j',l)] \boldsymbol{A}_l^{(k)}].$$

$(\boldsymbol{U}_{j'}^{(k+1)}, \boldsymbol{V}_{j'}^{(k+1)})$ = first singular vector pair of 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: $\boldsymbol{E} = \text{Mat}(\langle \boldsymbol{e} \rangle), \boldsymbol{L} = \text{Mat}(\hat{\boldsymbol{A}} \langle \boldsymbol{s} \rangle)$

4.2.5 Parameter Setting/Initialization and Dimension Trimming

Recall that we assume an Inverse-gamma prior on each element of $\boldsymbol{\gamma}$, i.e., $p(\gamma_i) = \text{IG}(a, b)$, with $b \to 0$. The main question is how to set the parameter a. Motivated by the objective function (4.16) in the M-step and the previous discussion, we set a such that $(2a+3) = \max\Big(\sqrt{\|\boldsymbol{E}_0\|_0/\text{rank}(\boldsymbol{L}_0)}, 1\Big)$. Note that here is a square root. Also note that in (4.16), the log function is used to encourage sparseness instead of the ℓ_0-'norm'. Since usually there is no knowledge of the true rank and sparsity, we estimate them from the data by thresholding $\boldsymbol{\gamma}^{(k)}$ and $\boldsymbol{\alpha}^{(k)}$ at the end of Step 1. For the input parameter β, we recommend to set it larger than the true noise variance, e.g., $(3\sigma)^2$ or even larger, to accommodate any modeling errors (interference) especially at the beginning of the iterations.

The initialization of $\boldsymbol{\gamma}^{(0)}$ and $\boldsymbol{\alpha}^{(0)}$ directly follows the standard SBL. One may simply initialize them to be a vector of ones if there is no prior knowledge of the scale of the outliers or singular values. Otherwise, it is recommended to scale the corresponding one vector accordingly.

Now we discuss the initialization of $\boldsymbol{U}^{(0)} \in \mathbb{R}^{n_1 \times d}$, $\boldsymbol{V}^{(0)} \in \mathbb{R}^{n_2 \times d}$, and the associated dimension d. For small size problems, we initialize $d = \min(n_1, n_2)$. However, for large-scale problems, in light of Proposition 4.2, we initialize d as 2 times the rank of some pre-estimated low-rank matrix, or as maximal target rank. So d is usually on the same order of the rank r, which greatly reduces the complexity of the proposed method. Note that it is a common practice to specify the maximal target rank in solving large-scale problems, e.g., [56], [57], [96]. Since a good initialization of $\boldsymbol{U}^{(0)}$ and $\boldsymbol{V}^{(0)}$ can help accelerate the convergence and avoid some local minimas. We initialize them as the singular vectors of some pre-estimated low-rank matrix $\hat{\boldsymbol{L}} \in \mathbb{R}^{n_1 \times n_2}$. Here the dimension of $\boldsymbol{U}^{(0)}$ is $n_1 \times d$, *not* $n_1 \times \text{rank}(\hat{\boldsymbol{L}})$. As in traditional SBL for sparse signal recovery [17], [18], one can prune the columns of $\boldsymbol{A}$ when the corresponding γ_i is smaller than a predefined threshold (e.g., 1×10^{-5}), for efficiency only (since d is reduced). We recommend to prune $\boldsymbol{\gamma}$ upon the convergence of Step 1 in

Algorithm 1.

4.3 Analysis of SBL Approach

4.3.1 Analysis and Support of the Updating Procedure for A

To simplify presentation, the following discussion again *assumes* that $|\langle \boldsymbol{s}_1 \rangle| \geq |\langle \boldsymbol{s}_2 \rangle| \geq \cdots$, and we first update $\boldsymbol{A}_1$, then $\boldsymbol{A}_2$.

We first provide some insight on why updating $\boldsymbol{A}_1$ works:

Let $\boldsymbol{A}^{true} = \boldsymbol{A}^{(k)} + \Delta\boldsymbol{A}^{(k)}$, $\boldsymbol{e}^{true} = \langle \boldsymbol{e} \rangle + \Delta\boldsymbol{e}$, $\boldsymbol{s}^{true} = \langle \boldsymbol{s} \rangle + \Delta\boldsymbol{s}$, then

$$\boldsymbol{m} - \langle \boldsymbol{e} \rangle = \boldsymbol{A}^{true}\langle \boldsymbol{s} \rangle + \boldsymbol{\eta} = \boldsymbol{A}^{(k)}\langle \boldsymbol{s} \rangle + \Delta\boldsymbol{A}^{(k)}\langle \boldsymbol{s} \rangle + \boldsymbol{\eta} \tag{4.20}$$

where $\boldsymbol{\eta} = \boldsymbol{A}^{true}\Delta\boldsymbol{s} + \Delta\boldsymbol{e} + \boldsymbol{n}$ that captures the original noise $\boldsymbol{n}$ and the additional modeling noise due to the estimation error in $\langle \boldsymbol{s} \rangle$ and $\langle \boldsymbol{e} \rangle$.

Now let us look at the first term in (4.17), which is the dominant term. From (4.20) we have

$$\begin{aligned} \boldsymbol{m} - \langle \boldsymbol{e} \rangle - \sum_{i=2}^{d} \langle \boldsymbol{s}_i \rangle \boldsymbol{A}_i^{(k)} &= \boldsymbol{A}_1^{(k)}\langle \boldsymbol{s}_1 \rangle + \Delta\boldsymbol{A}^{(k)}\langle \boldsymbol{s} \rangle + \boldsymbol{\eta} \\ &= (\boldsymbol{A}_1^{(k)} + \Delta\boldsymbol{A}_1^{(k)})\langle \boldsymbol{s}_1 \rangle + \sum_{i=2}^{d} \Delta\boldsymbol{A}_i^{(k)}\langle \boldsymbol{s}_i \rangle + \boldsymbol{\eta} \end{aligned} \tag{4.21}$$

The term $\sum_{i=2}^{d} \Delta\boldsymbol{A}_i^{(k)}\langle \boldsymbol{s}_i \rangle$ can be viewed as interference. Recall that $\boldsymbol{A}_1^{(k)} + \Delta\boldsymbol{A}_1^{(k)} = \boldsymbol{A}_1^{true}$, which has a nice rank-one structure, i.e., $\boldsymbol{A}_1^{true} = \mathrm{vec}(\boldsymbol{U}_1^{true}\boldsymbol{V}_1^{true\,T})$, and can be roughly found through rank-one SVD approximation in (4.19) if other interferences and $\boldsymbol{\eta}$ are relatively small. While other combinations, e.g., $\boldsymbol{A}_1^{(k)} + \Delta\boldsymbol{A}_2^{(k)} = \boldsymbol{A}_1^{(k)} - \boldsymbol{A}_2^{(k)} + \boldsymbol{A}_2^{true} = \mathrm{vec}(\boldsymbol{U}_1^{(k)}\boldsymbol{V}_1^{(k)^T} - \boldsymbol{U}_2^{(k)}\boldsymbol{V}_2^{(k)^T} + \boldsymbol{U}_2\boldsymbol{V}_2^{T})$, usually do not have such nice rank-one structure.

When updating $\boldsymbol{A}_2$ with other columns fixed, we use the latest update of $\boldsymbol{A}_1^{(k+1)}$. If $\boldsymbol{A}_1^{(k+1)} \approx \boldsymbol{A}_1^{true} = \boldsymbol{A}_1^{(k)} + \Delta\boldsymbol{A}_1^{(k)}$, we would have from (4.20) that:

$$\boldsymbol{m} - \langle \boldsymbol{e} \rangle - \sum_{i=3}^{d} \langle \boldsymbol{s}_i \rangle \boldsymbol{A}_i^{(k)} - \boldsymbol{A}_1^{(k+1)} \langle \boldsymbol{s}_1 \rangle \approx (\boldsymbol{A}_2^{(k)} + \Delta\boldsymbol{A}_2^{(k)}) \langle \boldsymbol{s}_2 \rangle + \sum_{i=3}^{d} \Delta\boldsymbol{A}_i^{(k)} \langle \boldsymbol{s}_i \rangle + \boldsymbol{\eta} \tag{4.22}$$

Note that the interference term $\Delta\boldsymbol{A}_1^{(k)}\langle \boldsymbol{s}_1 \rangle$ has disappeared from the right-hand-side of (4.22), compared with (4.21). Then finding the rank-one structure $(\boldsymbol{A}_2^{(k)} + \Delta\boldsymbol{A}_2^{(k)})$ will be relatively easier, since the large interference term $\Delta\boldsymbol{A}_1^{(k)}\langle \boldsymbol{s}_1 \rangle$ is almost cancelled.

Similar analysis applies for the updating of the other columns. This motivates our updating orders for the columns of $\boldsymbol{A}$, which is according to the decreasing order of the magnitudes of the elements in $\langle \boldsymbol{s} \rangle$. Also from the right-hand-side of (4.21), we can see that if $|\langle \boldsymbol{s}_1 \rangle|$ is much larger than $|\langle \boldsymbol{s}_2 \rangle|, |\langle \boldsymbol{s}_3 \rangle|, \cdots$, it is expected to be easier to identify the desired rank-one structure $(\boldsymbol{A}_1^{(k)} + \Delta\boldsymbol{A}_1^{(k)})$. In addtion, from (4.22), it is desirable to have the large interference term $\Delta\boldsymbol{A}_1^{(k)}\langle \boldsymbol{s}_1 \rangle$ first cancelled from its right-hand-side. The proposed updating order is similar to the Successive Interference Cancellation (SIC) strategy that is widely used in communication systems.

4.3.2 Algorithm Guarantee

Since the proposed algorithm is a MAP-EM algorithm [101], we have the following guarantee.

Theorem 4.1. *Algorithm 1 guarantees that* $p(\Psi^{(k+1)}|\boldsymbol{m}) \geq p(\Psi^{(k)}|\boldsymbol{m})$ *in each iteration.*

Proof (***Sketch***): Recall that $\Psi = (\boldsymbol{A}, \boldsymbol{\gamma}, \boldsymbol{\alpha})$. By Theorem 7 of [101], the update of $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ in Step 1 of Algorithm 1 guarantees that $p((\boldsymbol{A}^{(k)}, \boldsymbol{\gamma}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)})|\boldsymbol{m}) \geq p((\boldsymbol{A}^{(k)}, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\alpha}^{(k)})|\boldsymbol{m})$. Also, the update of matrix $\boldsymbol{A}$ in Step 2 further guarantees $p((\boldsymbol{A}^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)})|\boldsymbol{m}) \geq p((\boldsymbol{A}^{(k)}, \boldsymbol{\gamma}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)})|\boldsymbol{m})$.

4.3.3 Underlying SBL Objective Function

The procedure employed is Type-II MAP, i.e., maximize $p(\Psi|\boldsymbol{m}) \propto p(\boldsymbol{m}|\Psi)p(\gamma)$, which is guaranteed the ascent properties at each step by Theorem 4.1. After applying $-2\log(\cdot)$ transformation, we can show

$$\min_{\gamma,\boldsymbol{\alpha},\boldsymbol{A}\in\mathbb{A}} -2\log[p(\boldsymbol{m}|\Psi)p(\gamma)] = \min_{\gamma,\boldsymbol{\alpha},\boldsymbol{A}\in\mathbb{A}} m^T\Sigma_{\boldsymbol{m}}^{-1}m + \log|\Sigma_{\boldsymbol{m}}| + 2(a+1)\log|\boldsymbol{\Gamma}| + const$$

$$= \min_{\gamma,\boldsymbol{\alpha},\boldsymbol{A}\in\mathbb{A}}\{\min_{\boldsymbol{s},\boldsymbol{e}}[\tfrac{1}{\beta}\|\boldsymbol{m}-\boldsymbol{A}\boldsymbol{s}-\boldsymbol{e}\|_2^2+\boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s}+\boldsymbol{e}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{e}]+\log|\Sigma_{\boldsymbol{m}}|+2(a+1)\log|\boldsymbol{\Gamma}|\}+const$$

$$= \min_{\boldsymbol{s},\boldsymbol{e},\boldsymbol{A}\in\mathbb{A}}\{\tfrac{1}{\beta}\|\boldsymbol{m}-\boldsymbol{A}\boldsymbol{s}-\boldsymbol{e}\|_2^2+\underbrace{\min_{\gamma,\boldsymbol{\alpha}}[\boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s}+\boldsymbol{e}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{e}+\log|\Sigma_{\boldsymbol{m}}|+2(a+1)\log|\boldsymbol{\Gamma}|]}_{g_{SBL}(\boldsymbol{A},\boldsymbol{s},\boldsymbol{e})}\}+const$$

The first term is the data-fidelity term, while the remaining $g_{SBL}(\boldsymbol{A},\boldsymbol{s},\boldsymbol{e})$ is our underlying penalty term. Recall that $\Sigma_{\boldsymbol{m}} = \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T+\boldsymbol{\Lambda}+\beta\boldsymbol{I}$. It is known that log-determinant encourages low-rank [52]. So $\log|\Sigma_{\boldsymbol{m}}|$ 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. So, we can see that the proposed model indeed *leads to sparse solutions*.

4.3.4 Complexity Analysis

In Step 1, thanks to the matrix inversion lemma used in (4.10), the complexity of calculating $\boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}$, $\boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}$, $\mathrm{diag}(\Sigma_{\boldsymbol{s}|\boldsymbol{m}})$, and $\mathrm{diag}(\Sigma_{\boldsymbol{e}|\boldsymbol{m}})$ is only $O(d^2n_1n_2)$. In Step 2, the complexity is $O(d^2n_1n_2)$. As mentioned in Section 4.2.5, for large-scale problems, we initialize d to the same order of the rank r, rather than $\min(n_1,n_2)$. Then the complexity significantly reduces to $O(r^2n_1n_2)$ in each iteration.

4.4 Modified SBL Approach

Reflecting on the algorithm at a high level, Algorithm 1 updates $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ with $\boldsymbol{A}$ fixed, and then updates $\boldsymbol{A}$ with $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ fixed, iteratively. As analyzed before, the proposed SBL cost function does lead to sparse solutions. However, there is additional information

that could be further utilized. More specifically, in Robust PCA problem (i.e., Sparse and Low-rank decomposition), usually the outliers are assumed to be spread out, i.e., sparse in each row and each column. To utilize this information, in this section, we modify the SBL cost function when we update $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ with $\boldsymbol{A}$ fixed, which leads to further performance improvement.

4.4.1 Algorithm

Recall that our original SBL cost function is $\boldsymbol{m}^T\boldsymbol{\Sigma}_{\boldsymbol{m}}^{-1}\boldsymbol{m} + \log|\boldsymbol{\Sigma}_{\boldsymbol{m}}| + 2(a+1)\log|\boldsymbol{\Gamma}|$, where $\boldsymbol{\Sigma}_{\boldsymbol{m}} = \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \boldsymbol{\Lambda} + \beta\boldsymbol{I}$ is the covariance matrix of $\boldsymbol{m}$, and $\log|\boldsymbol{\Sigma}_{\boldsymbol{m}}|$ pushes both $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$ to be sparse. In order to use the information that the outliers are sparse in each row and each column, we replace the term $\log|\boldsymbol{\Sigma}_{\boldsymbol{m}}|$ by its sub-blocks, i.e., $0.5\sum_j \log|\boldsymbol{\Sigma}_{\boldsymbol{M}_j}| + 0.5\sum_i \log|\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}|$, where $\boldsymbol{\Sigma}_{\boldsymbol{M}_j}$ is the covariance matrix of the *jth* column of $\boldsymbol{M}$, and $\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}$ is the covariance matrix of the *ith* row of $\boldsymbol{M}$. As we will show next, this encourages the outliers being not only sparse, but also sparse in each row and each column.

We have $\boldsymbol{\Sigma}_{\boldsymbol{M}_j} = \boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T + \boldsymbol{\Lambda}_{\cdot j} + \beta\boldsymbol{I}$, and $\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i} = \boldsymbol{A}_{i\cdot}\boldsymbol{\Gamma}\boldsymbol{A}_{i\cdot}^T + \boldsymbol{\Lambda}_{i\cdot} + \beta\boldsymbol{I}$, where the $\boldsymbol{A}_{\cdot j} \triangleq \boldsymbol{A}(1+(j-1)n_1 : jn_1, :)$ is a sub-matrix of $\boldsymbol{A}$ that corresponds to *jth* column of the low-rank matrix, i.e., $\boldsymbol{L}_j = \boldsymbol{A}_{\cdot j}\boldsymbol{s}$. Similarly, $\boldsymbol{A}_{i\cdot} \triangleq \boldsymbol{A}(i : n_1 : i+(n_2-1)n_1, :)$ that corresponds to *ith* row of the low-rank matrix $\boldsymbol{L}$. Also, $\boldsymbol{\Lambda}_{\cdot j} \triangleq \text{diag}(\boldsymbol{\alpha}(1+(j-1)n_1 : jn_1))$ which corresponds to *jth* column of Mat($\boldsymbol{\alpha}$). Similarly, $\boldsymbol{\Lambda}_{i\cdot} \triangleq \text{diag}(\boldsymbol{\alpha}(i : n_1 : i+(n_2-1)n_1))$ that corresponds to *ith* row of Mat($\boldsymbol{\alpha}$). It's easy to see that $\boldsymbol{\Sigma}_{\boldsymbol{M}_j}$ is a diagonal block of $\boldsymbol{\Sigma}_{\boldsymbol{M}}$. By Hadamard-Fischer Inequality, we have that $\log|\boldsymbol{\Sigma}_{\boldsymbol{m}}| \leq \sum_j \log|\boldsymbol{\Sigma}_{\boldsymbol{M}_j}|$. Similarly, one can show that $\log|\boldsymbol{\Sigma}_{\boldsymbol{m}}| \leq \sum_i \log|\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}|$. So, it turns out that $0.5\sum_j \log|\boldsymbol{\Sigma}_{\boldsymbol{M}_j}| + 0.5\sum_i \log|\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}|$ is an upper bound of $\log|\boldsymbol{\Sigma}_{\boldsymbol{m}}|$.

Intuitively, the term $\log|\boldsymbol{\Sigma}_{\boldsymbol{M}_j}| = \log|\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T + \boldsymbol{\Lambda}_{\cdot j} + \beta\boldsymbol{I}|$ encourages $\boldsymbol{\Lambda}_{\cdot j}$ to be low-rank, thus encouraging the outliers in *jth* column to be sparse. Similarly, the term $\log|\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}|$ encourages the outliers in *ith* row to be sparse.

Now the question is how to optimize the following modified SBL cost function w.r.t. $\boldsymbol{\alpha} \geq 0$ and $\boldsymbol{\gamma} \geq 0$ when $\boldsymbol{A}$ is fixed, which is an upper bound of the original SBL cost function.

$$\boldsymbol{m}^T \boldsymbol{\Sigma}_{\boldsymbol{m}}^{-1} \boldsymbol{m} + 0.5 \sum_j \log |\boldsymbol{\Sigma}_{\boldsymbol{M}_j}| + 0.5 \sum_i \log |\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}| + 2(a+1) \log |\boldsymbol{\Gamma}| \tag{4.23}$$

Following the variational bounding method used in [62], we construct the upper bounds for the first three terms and minimize the overall upper bound w.r.t. $\boldsymbol{\alpha} \geq 0$ and $\boldsymbol{\gamma} \geq 0$. The details can be found in the Appendix.

Combining these upper bounds, we have

$$\begin{aligned}(4.23) \leq & (\boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}^{(t)})^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}^{(t)} + (\boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}^{(t)})^T \boldsymbol{\Lambda}^{-1} \boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}^{(t)} + (0.5n_1 + 0.5n_2 + 2a + 2) \log |\boldsymbol{\Gamma}| + \log |\boldsymbol{\Lambda}| \\ & + \sum_j [0.5 \mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{M}_j}^{(t)})^T \boldsymbol{\gamma}^{-1} + \mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{E}_j|\boldsymbol{M}_j}^{(t)})^T \mathrm{diag}(\boldsymbol{\Lambda}_{\cdot j}^{-1})] \\ & + \sum_i [0.5 \mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{s}|(\boldsymbol{M}^T)_i}^{(t)})^T \boldsymbol{\gamma}^{-1} + \mathrm{diag}(\boldsymbol{\Sigma}_{(\boldsymbol{E}^T)_i|(\boldsymbol{M}^T)_i}^{(t)})^T \mathrm{diag}(\boldsymbol{\Lambda}_{i\cdot}^{-1})] + Const, \end{aligned} \tag{4.24}$$

where $\boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}^{(t)} = \boldsymbol{\Gamma}^{(t)} \boldsymbol{A}^T (\boldsymbol{A} \boldsymbol{\Gamma}^{(t)} \boldsymbol{A}^T + \boldsymbol{\Lambda}^{(t)} + \beta \boldsymbol{I})^{-1} \boldsymbol{m}$, $\boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}^{(t)} = \boldsymbol{\Lambda}^{(t)} (\boldsymbol{A} \boldsymbol{\Gamma}^{(t)} \boldsymbol{A}^T + \boldsymbol{\Lambda}^{(t)} + \beta \boldsymbol{I})^{-1} \boldsymbol{m}$

$$\begin{aligned} & \boldsymbol{\Sigma}_{\boldsymbol{M}_j}^{(t)} \triangleq \boldsymbol{A}_{\cdot j} \boldsymbol{\Gamma}^{(t)} \boldsymbol{A}_{\cdot j}^T + \boldsymbol{\Lambda}_{\cdot j}^{(t)} + \beta \boldsymbol{I}, && \boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}^{(t)} \triangleq \boldsymbol{A}_{i\cdot} \boldsymbol{\Gamma}^{(t)} \boldsymbol{A}_{i\cdot}^T + \boldsymbol{\Lambda}_{i\cdot}^{(t)} + \beta \boldsymbol{I} \\ & \boldsymbol{\Sigma}_{\boldsymbol{E}_j|\boldsymbol{M}_j}^{(t)} \triangleq \boldsymbol{\Lambda}_{\cdot j}^{(t)} - \boldsymbol{\Lambda}_{\cdot j}^{(t)} (\boldsymbol{\Sigma}_{\boldsymbol{M}_j}^{(t)})^{-1} \boldsymbol{\Lambda}_{\cdot j}^{(t)}, && \boldsymbol{\Sigma}_{(\boldsymbol{E}^T)_i|(\boldsymbol{M}^T)_i}^{(t)} \triangleq \boldsymbol{\Lambda}_{i\cdot}^{(t)} - \boldsymbol{\Lambda}_{i\cdot}^{(t)} (\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}^{(t)})^{-1} \boldsymbol{\Lambda}_{i\cdot}^{(t)} \\ & \boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{M}_j}^{(t)} \triangleq \boldsymbol{\Gamma}^{(t)} - \boldsymbol{\Gamma}^{(t)} \boldsymbol{A}_{\cdot j}^T (\boldsymbol{\Sigma}_{\boldsymbol{M}_j}^{(t)})^{-1} \boldsymbol{A}_{\cdot j} \boldsymbol{\Gamma}^{(t)}, && \boldsymbol{\Sigma}_{\boldsymbol{s}|(\boldsymbol{M}^T)_i}^{(t)} \triangleq \boldsymbol{\Gamma}^{(t)} - \boldsymbol{\Gamma}^{(t)} \boldsymbol{A}_{i\cdot}^T (\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}^{(t)})^{-1} \boldsymbol{A}_{i\cdot} \boldsymbol{\Gamma}^{(t)} \end{aligned}$$

Taking derivatives, we obtain the modified updating rules for $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$:

Update $\boldsymbol{\alpha}$: Define $ii = i - n_1(\lceil \frac{i}{n_1} \rceil - 1)$ and $jj = \lceil \frac{i}{n_1} \rceil$,

$$\boldsymbol{\alpha}_i^{(t+1)} = (\boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}^{(t)}(i))^2 + \boldsymbol{\Sigma}_{\boldsymbol{E}_{jj}|\boldsymbol{M}_{jj}}^{(t)}(ii, ii) + \boldsymbol{\Sigma}_{(\boldsymbol{E}^T)_{ii}|(\boldsymbol{M}^T)_{ii}}^{(t)}(jj, jj).$$

Note that $\boldsymbol{\alpha}_i$ corresponds to the element in $iith$ row and $jjth$ column of the matrix $\mathrm{Mat}(\boldsymbol{\alpha})$.

Update γ:

$$\gamma^{(t+1)} = \frac{(\boldsymbol{\mu}^{(t)}_{\boldsymbol{s}|\boldsymbol{m}})^2 + 0.5\sum_j \text{diag}(\boldsymbol{\Sigma}^{(t)}_{\boldsymbol{s}|M_j}) + 0.5\sum_i \text{diag}(\boldsymbol{\Sigma}^{(t)}_{\boldsymbol{s}|(M^T)_i})}{2a + 2 + 0.5n_1 + 0.5n_2}$$

Comparing with the original SBL updating rules of $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, the original big posterior covariance matrix, e.g., $\boldsymbol{\Sigma}_{e|\boldsymbol{m}}$, is replaced by sum of small posterior covariance matrices corresponding to the row and the column. This replacement is intuitive since the purpose of our modified SBL is to enforce the outliers to be sparse in each row and each column.

In summary, the proposed modified SBL approach merely replaces the updating rules of $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ in Algorithm 1 by the new updating rules above. At the high level, this approach first minimizes the modified objective function, which is an upper bound of the original SBL objective function, to update both $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$; and then minimizes the original SBL objective to estimate $\boldsymbol{A}$ via MAP-EM, alternatively. As we will see in the numerical experiments, this utilization of the sparsity pattern information further improves the recovery performance.

4.4.2 Relation to Previous Work

In [62], it is assumed that the covariance matrix of vec($\boldsymbol{L}$) has a Kronecker sum structure, i.e., $\boldsymbol{\Psi}_r \oplus \boldsymbol{\Psi}_c$. The resulting objective function is $\boldsymbol{m}^T\boldsymbol{\Sigma}_{\boldsymbol{m}}^{-1}\boldsymbol{m} + \log|\boldsymbol{\Sigma}_{\boldsymbol{m}}|$, where $\boldsymbol{\Sigma}_{\boldsymbol{m}} = \boldsymbol{\Psi}_r \oplus \boldsymbol{\Psi}_c + \boldsymbol{\Lambda} + \beta\boldsymbol{I}$. However, the high computational complexity leads to breaking the term $\log|\boldsymbol{\Sigma}_{\boldsymbol{m}}|$ into smaller pieces: $\sum_j \log|\boldsymbol{\Psi}_c + \frac{1}{2}\boldsymbol{\Lambda}_{\cdot j} + \frac{\beta}{2}\boldsymbol{I}| + \sum_i \log|\boldsymbol{\Psi}_r + \frac{1}{2}\boldsymbol{\Lambda}_{i\cdot} + \frac{\beta}{2}\boldsymbol{I}|$, which is a *lower* bound of $\log|\boldsymbol{\Sigma}_{\boldsymbol{m}}|$. Then they use the variational bounding technique to minimize the associated new objective function. Though this modification of the objective function is purely for reducing complexity and non-intuitive, it achieves much better performance than other competing methods. By investigating their objective function, we find that the efficacy lies in the term $\sum_j \log|\boldsymbol{\Psi}_c + \frac{1}{2}\boldsymbol{\Lambda}_{\cdot j} + \frac{\beta}{2}\boldsymbol{I}| + \sum_i \log|\boldsymbol{\Psi}_r + \frac{1}{2}\boldsymbol{\Lambda}_{i\cdot} + \frac{\beta}{2}\boldsymbol{I}|$, which implicitly encourages the outliers to be sparse in each row and each column.

Motivated by the above observation, we proposed the modified SBL in this section, not due to complexity issue. Our modified objective function is an *upper* bound of the original objective function, and the terms $\boldsymbol{\Sigma}_{\boldsymbol{M}_j}$ and $\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}$ here have physical meanings.

It is also worth mentioning that, the sparsity pattern together with the sparsity level information of the outliers have been explicitly utilized in a recent Robust PCA work [103], where a thresholding operator is used to keep each row and each column of the estimated $\boldsymbol{E}$ having at most p fraction of outliers. However, the parameter p needs to be specified. While in our modified SBL, the sparsity level of the outliers in each column and each row is automatically learned from the data, and does not require to be specified.

4.4.3 Complexity Analysis

The only difference between Modified SBL and the original SBL is the updating rules of $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$ in Step 1. In the original SBL, calculating the posterior mean and variance of $\boldsymbol{e}$ and $\boldsymbol{s}$ is $O(d^2 n_1 n_2)$. In the modified SBL, the complexity of calculating $\boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}}$ and $\boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}}$ is $O(d^2 n_1 n_2)$. To calculate $\text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{E}_{jj}|\boldsymbol{M}_{jj}})$ and $\text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{s}|\boldsymbol{M}_j})$, the complexity is $O(d^2 n_1)$, thanks to the matrix inversion lemma. There are n_2 columns in total. It is useful to note that this allows for a high level of parallelism. Similarly, to calculate $\text{diag}(\boldsymbol{\Sigma}_{(\boldsymbol{E}^T)_{ii}|(\boldsymbol{M}^T)_{ii}})$ and $\text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{s}|(\boldsymbol{M}^T)_i})$, the complexity is $O(d^2 n_2)$ and there are n_1 rows, which can also be parallelized.

Note that in [62], there is a need to calculate the inverse of the $n_1 \times n_1$ matrix $(\boldsymbol{\Psi}_c + 0.5\boldsymbol{\Lambda}_{\cdot j} + 0.5\beta\boldsymbol{I})$ for n_2 columns, and the inverse of the $n_2 \times n_2$ matrix $(\boldsymbol{\Psi}_r + 0.5\boldsymbol{\Lambda}_{i\cdot} + 0.5\beta\boldsymbol{I})$ for n_1 rows. The complexity of a single matrix inversion there is $O(\max(n_1, n_2)^3)$, which is *prohibitive* for large scale problems. Note that the matrix inversion lemma unfortunately can not be applied to reduce their complexity. As a result, the complexity of the method in [62] is much higher than the proposed methods.

4.5 Empirical Studies

In this section, we compare the proposed SBL methods with the following state-of-the-art methods: PCP [39]–[41], SPCP [44], [98], Iterative Reweighted PCP (IR-PCP) [51], [104], AltProj [96], SRPCP [105], Babacan's VB-RPCA [56], BRMF [57], and PB-RPCA [62]. We additionally compare with the nuclear norm based noisy matrix completion (MC) [100], [106], where the locations of the outliers are known, and only outlier-free entries are observed. This problem is much easier than Robust PCA and serves as an oracle solution [62], [90]. We use the source codes from the authors and the corresponding parameters are carefully tuned. For the proposed methods, we initialize $\boldsymbol{\gamma}^{(0)}$ and $\boldsymbol{\alpha}^{(0)}$ to be vector of ones. $\boldsymbol{U}^{(0)}$ and $\boldsymbol{V}^{(0)}$ are initialized from the singular vectors of the low-rank matrix estimated by SRPCP. More specifically, denote $\tilde{\boldsymbol{L}}$ as the solution of SRPCP, $\boldsymbol{U}^{(0)}$ and $\boldsymbol{V}^{(0)}$ are initialized as its first $2 \times \text{rank}(\tilde{\boldsymbol{L}})$ singular vector pairs. The noise standard deviation is provided to SPCP, SRPCP, PB-RPCA, VB-RPCA, and the proposed SBL methods. BRMF, AltProj, and VB-RPCA need the knowledge of the maximal possible rank of $\boldsymbol{L}$, and it is specified as 2 times the true rank.

4.5.1 Comparison on Simulated Data

The experimental setup is the same as [105] and similar to [97], [98], which is as follows:

1) Given the rank r, the low-rank component $\boldsymbol{L}_0$ is built as $\boldsymbol{L}_0 = \boldsymbol{A}\boldsymbol{B}^T$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are randomly generated $n \times r$ standard Gaussian matrices;

2) Given the fraction ρ (corruption rate) of non-zero entries in $\boldsymbol{E}_0$, the support of $\boldsymbol{E}_0$ is chosen uniformly at random with size ρn^2, and the value of each non-zero entry is independently drawn from a uniform distribution over the interval $[0, 100]$;

3) Each entry of the noise $\boldsymbol{N}$ is independently drawn from a Gaussian distribution with

mean 0 and variance σ^2.

4) Finally, generate $\boldsymbol{M} = \boldsymbol{L}_0 + \boldsymbol{E}_0 + \boldsymbol{N}$. Estimate $\boldsymbol{L}_0$ and $\boldsymbol{E}_0$ from $\boldsymbol{M}$ using different methods.

We set $n = 100$, $\sigma = 0.1$. For each $r \in \{1 : 40\}$, and each $\rho \in \{0.01\ 0.05 : 0.05 : 0.60\}$, we repeat the above procedure 10 times. For evaluation, the estimated $\hat{\boldsymbol{L}}$ is compared with the ground truth via the Relative Error $\frac{\|\hat{\boldsymbol{L}}-\boldsymbol{L}_0\|_F}{\|\boldsymbol{L}_0\|_F}$. We report the average Relative Error over all trials in the log scale as in [105], i.e., $2\log(\text{Average}(\frac{\|\hat{\boldsymbol{L}}-\boldsymbol{L}_0\|_F}{\|\boldsymbol{L}_0\|_F}))$. We also compute the distance between the estimated outlier support and the true support. Denoting the two supports as $\hat{\Omega}$ and Ω, the Support Distance is defined as in [107]: $\text{dist}(\hat{\Omega}, \Omega) = (\max\{|\hat{\Omega}|, |\Omega|\} - |\hat{\Omega} \cap \Omega|)/\max\{|\hat{\Omega}|, |\Omega|\}$. The outlier support is determined by thresholding $\hat{\boldsymbol{E}}$ or $(\boldsymbol{M} - \hat{\boldsymbol{L}})$.

Fig. 4.1 shows the average Relative Error of different methods in the log scale. Note that in the color scale bar, 0 means $2\log(\text{Average}(\frac{\|\hat{\boldsymbol{L}}-\boldsymbol{L}_0\|_F}{\|\boldsymbol{L}_0\|_F})) = 0$, i.e., the average Relative Error is 1. So the red color indicates very poor recovery. Similarly, -2 means the average Relative Error is 10^{-1}, indicated by the green color. Fig. 4.2 shows the average Support Distance of each method. Note that in the gray scale bar, 0 means exact support recovery, while 1 means very poor support recovery. It is clear that the proposed SBL methods have significantly better support recovery than the other methods. In terms of the recovery of the low-rank matrix, we can see that the proposed SBL method demonstrates an improvement over its initialization SRPCP. And the modified SBL method has a further improvement and nearly matches the performance of the oracle Matrix Completion method. In the experiments, we notice that there are many cases where SRPCP fails, which means the initialization of $\boldsymbol{U}^{(0)}$ and $\boldsymbol{V}^{(0)}$ are poor, while the proposed SBL methods finally return good estimates.

As discussed in [105], the superior performance of the Matrix Completion agrees with the intuition that correcting erasures with known locations is easier than correcting errors with unknown locations [90]. In Robust PCA, there is no knowledge of the locations and values of the outliers/errors. While in matrix completion, the locations of the erasures are known.

The proposed modified SBL method utilizes some location information (i.e., sparsity pattern) of the outliers and therefore pushes its performance towards the oracle Matrix Completion solution.

The conclusion for the double-sided outlier corruptions (i.e., $\sim U[-100, 100]$) is similar. We have also tested the case where the entries of $\boldsymbol{A}$ and $\boldsymbol{B}$ are randomly drawn from uniform distribution $U[-1, 1]$. The relative performance of the compared methods remains unchanged.

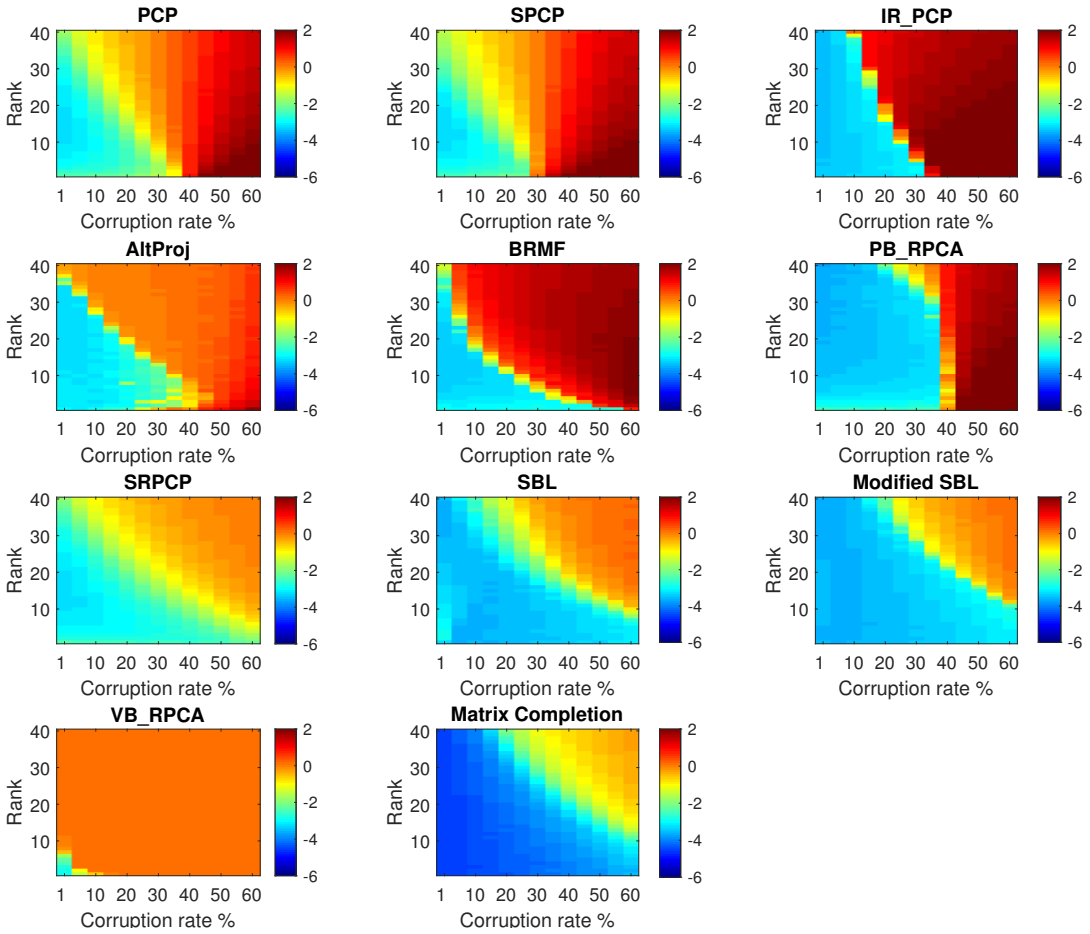

Figure 4.1. Average Relative Error of each method in log scale w.r.t. different rank and corruption rate.

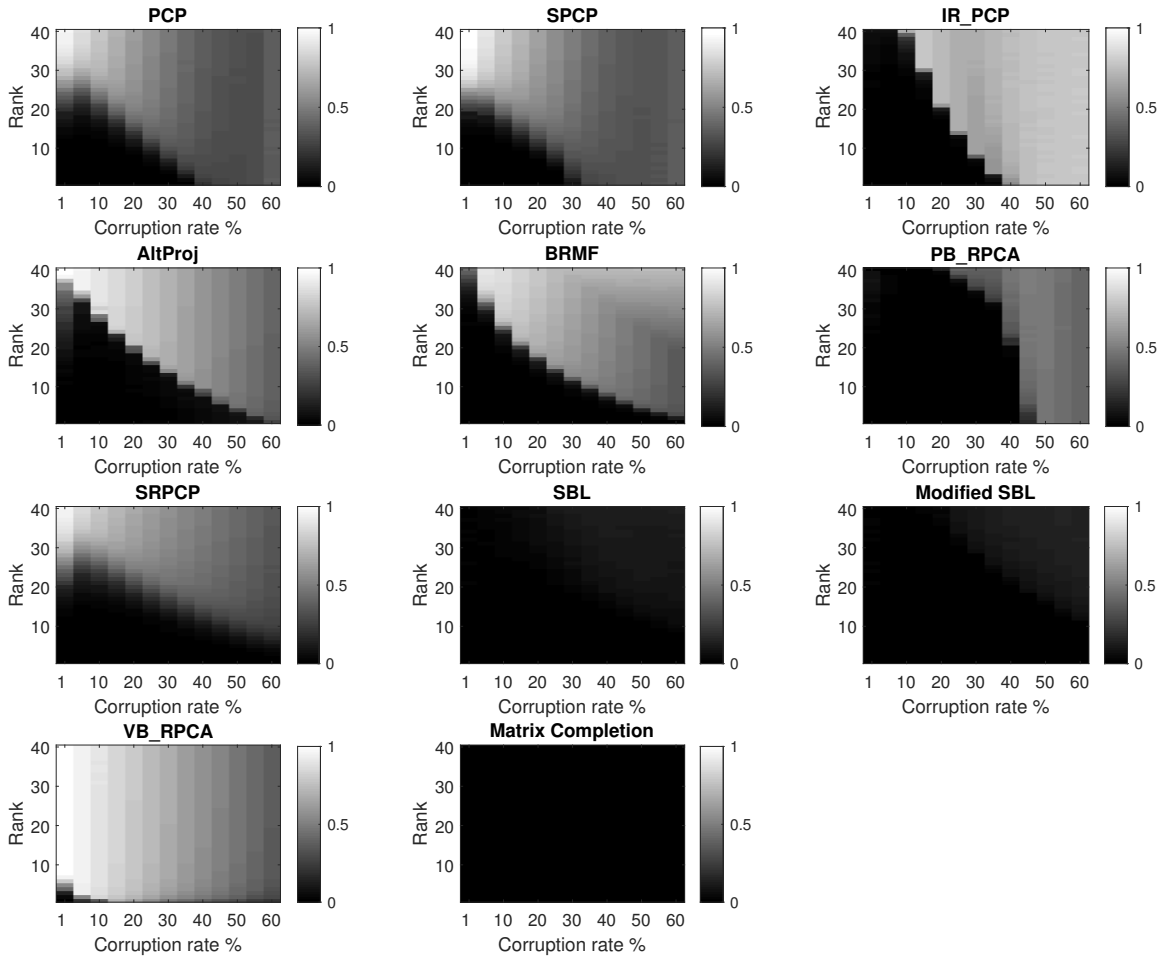

Figure 4.2. Average Support Distance of each method w.r.t. different rank and corruption rate.

4.5.2 Comparison on Text Removal

In this subsection, we follow [57] to conduct a text removal image processing simulation, where the results are directly visible. The experiment settings are the same as in [105]. The ground truth low-rank clean image is a 256×256 matrix with rank equal to 10, whose values are between -1 and 1. We embed black text in the image, where the values of the text are randomly drawn from $\mathrm{U}[-1, 0]$. The text can be viewed as sparse outliers. For evaluation, we compare the recovered low-rank matrix with the ground truth via ℓ_2 error, i.e., $\|\hat{\boldsymbol{L}} - \boldsymbol{L}_0\|_F$. As the support (mask) of the text is also of interest, the mask of the text is usually obtained by

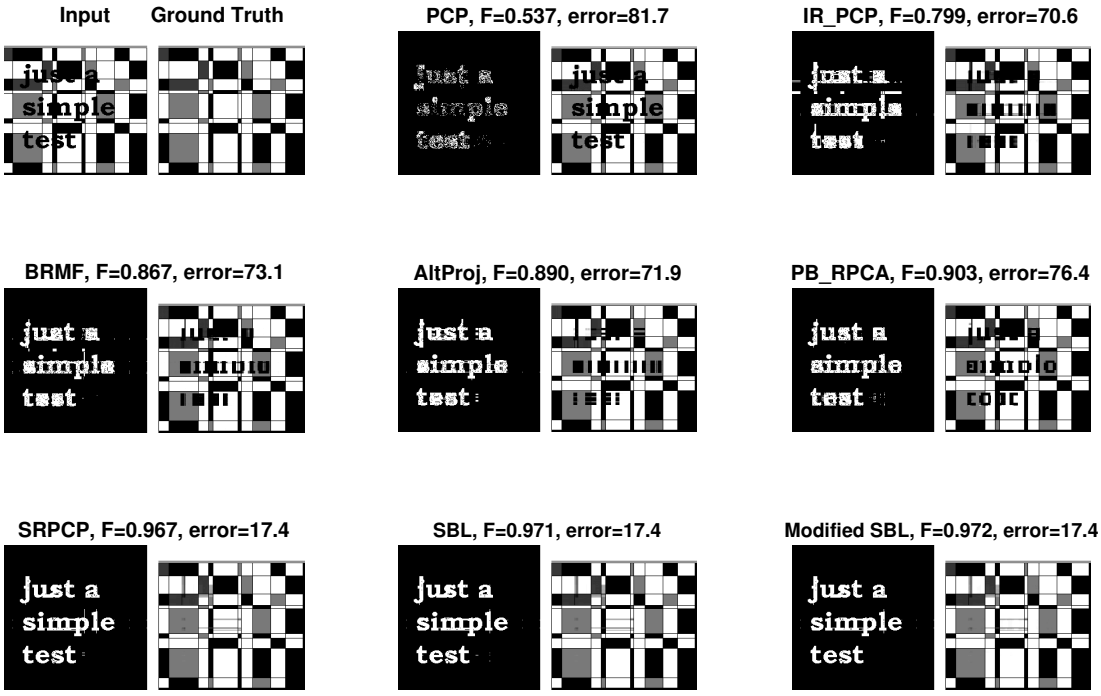

Figure 4.3. Recovered text mask (left, measured by F-measure) and low-rank matrix (right, measured by ℓ_2 error) by each method.

thresholding the estimated $\hat{\boldsymbol{E}}$. We vary the threshold as in [57], [105] to find the maximum F-measure for each method, where the F-measure is commonly used in pattern recognition and is defined as: 2(precision $\cdot$ recall)/(precision+recall). Fig. 4.3 shows the results of each method. It can be seen that most methods failed to return a clean low-rank image. SRPCP and the proposed SBL methods are able to recover a relatively clean low-rank image. The modified SBL method performs best in terms of F-measure and ℓ_2 error.

Finally, we remind the reader that, better performance can be expected via training based text recognition or incorporating the continuity prior [57] of the text. The main purpose here is to use an example to visually illustrate the effectiveness of the proposed methods.

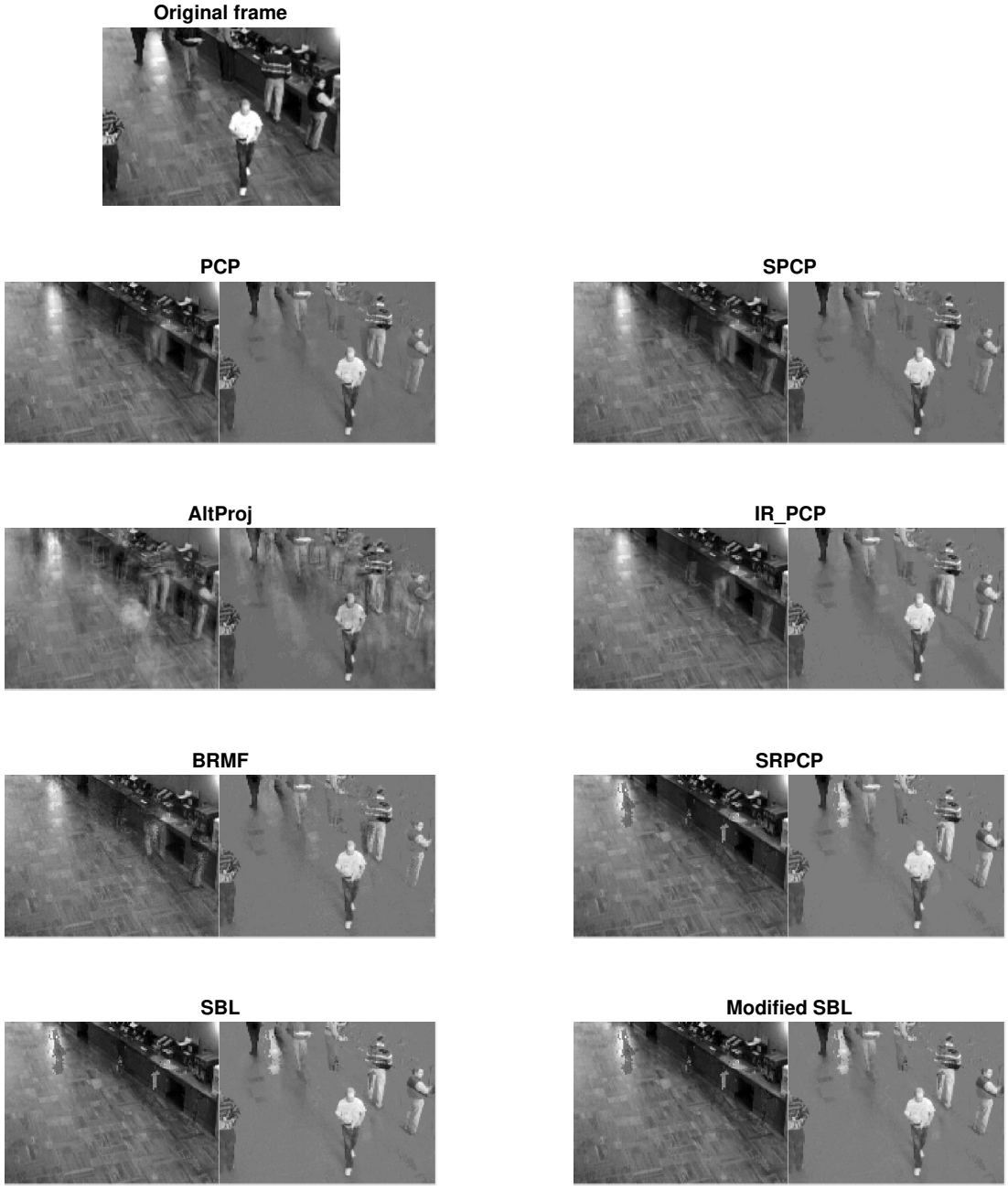

Figure 4.4. Recovered background (left) and foreground (right) by each method.

4.5.3 Comparison on Real Data

Lastly, we compare the performance of the methods on first 200 frames of a surveillance video[4], where each frame is converted to a column vector, and the integer pixel values

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

are scaled to the range [-1,1]. The background over the frames is the low-rank component and the moving objects over the frames can be considered as the sparse component. We exclude the PB-RPCA method for comparison due to its very high computational complexity. The parameter settings of other compared methods are the same as that in [105]. Fig. 4.4 shows the recovered background (left) and foreground (right) in the first frame. We can see that there are some ghosting effects in the recovered background of the competing BRMF Bayesian method, while the proposed SBL methods separate the foreground with the background very well. Note that the lighting at the top of the video changes over the frames, which can be considered as the foreground.

4.6 Conclusion

A concise SBL model is developed in light of a new objective, which is proven to be equivalent to the fundamental Robust PCA objective. This new model allows simple and effective Empirical Bayesian inference via MAP-EM. To further utilize the sparsity pattern information of the outliers, a modified SBL approach is further proposed. Empirical studies demonstrate the superiority and efficacy of the proposed methods.

Chapter 4, in part, is a reprint of the material as it appears in the papers: J. Liu and B. D. Rao, "Sparse Bayesian Learning for Robust PCA: Algorithms and Analyses," Submitted, and J. Liu, Y. Ding and B. Rao, "Sparse Bayesian Learning for Robust PCA," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May. 2019, pp. 4883-4887. The dissertation author was the primary investigator and author of these papers.

4.7 Appendices

4.7.1 Construct the Upper Bound for (4.23)

Following the variational bounding method used in [62], we construct the upper bounds for each term of (4.23). We first have

$$\boldsymbol{m}^T\boldsymbol{\Sigma}_{\boldsymbol{m}}^{-1}\boldsymbol{m} \leq \frac{1}{\beta}\|\boldsymbol{m}-\boldsymbol{A}\boldsymbol{s}-\boldsymbol{e}\|_2^2 + \boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s} + \boldsymbol{e}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{e}$$

for any $\boldsymbol{s}$ and $\boldsymbol{e}$, with equality achieved when $\boldsymbol{e} = \boldsymbol{\mu}_{\boldsymbol{e}|\boldsymbol{m}} = \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{\boldsymbol{m}}^{-1}\boldsymbol{m}$, and $\boldsymbol{s} = \boldsymbol{\mu}_{\boldsymbol{s}|\boldsymbol{m}} = \boldsymbol{\Gamma}\boldsymbol{A}^T\boldsymbol{\Sigma}_{\boldsymbol{m}}^{-1}\boldsymbol{m}$.

For the second term, following the approaches in [18], [62], we have

$$\begin{aligned}
&0.5\log|\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T + \boldsymbol{\Lambda}_{\cdot j} + \beta\boldsymbol{I}| \\
=&0.5\log|\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T| + 0.5\log|\boldsymbol{\Lambda}_{\cdot j}| + 0.5\log|W(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T, \boldsymbol{\Lambda}_{\cdot j})| + C_0 \\
\leq&0.5\log|\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T| + 0.5\log|\boldsymbol{\Lambda}_{\cdot j}| + 0.5\,\mathrm{Trace}[\nabla^T_{(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T)^{-1}}(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T)^{-1}] + 0.5\nabla^T_{\boldsymbol{\Lambda}_{\cdot j}^{-1}}\mathrm{diag}(\boldsymbol{\Lambda}_{\cdot j}^{-1}) + C_1 \\
\leq&0.5\log|\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T| + 0.5\log|\boldsymbol{\Lambda}_{\cdot j}| + 0.5\,\mathrm{Trace}[\nabla^T_{(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T)^{-1}}(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T)^{-1}] + \nabla^T_{\boldsymbol{\Lambda}_{\cdot j}^{-1}}\mathrm{diag}(\boldsymbol{\Lambda}_{\cdot j}^{-1}) + C_1 \\
=&0.5\log|\boldsymbol{\Gamma}| + 0.5\log|\boldsymbol{\Lambda}_{\cdot j}| + 0.5\,\mathrm{Trace}[\nabla^T_{(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T)^{-1}}(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T)^{-1}] + \nabla^T_{\boldsymbol{\Lambda}_{\cdot j}^{-1}}\mathrm{diag}(\boldsymbol{\Lambda}_{\cdot j}^{-1}) + C_2 \\
=&0.5\log|\boldsymbol{\Gamma}| + 0.5\log|\boldsymbol{\Lambda}_{\cdot j}| + 0.5\,\mathrm{Trace}[(\boldsymbol{\Gamma}^{(t)} - \boldsymbol{\Gamma}^{(t)}\boldsymbol{A}_{\cdot j}^T(\boldsymbol{\Sigma}_{\boldsymbol{M}_j}^{(t)})^{-1}\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}^{(t)})\boldsymbol{\Gamma}^{-1}] + \nabla^T_{\boldsymbol{\Lambda}_{\cdot j}^{-1}}\mathrm{diag}(\boldsymbol{\Lambda}_{\cdot j}^{-1}) + C_2 \\
=&0.5\log|\boldsymbol{\Gamma}| + 0.5\log|\boldsymbol{\Lambda}_{\cdot j}| + 0.5\mathrm{diag}(\boldsymbol{\Gamma}^{(t)} - \boldsymbol{\Gamma}^{(t)}\boldsymbol{A}_{\cdot j}^T(\boldsymbol{\Sigma}_{\boldsymbol{M}_j}^{(t)})^{-1}\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}^{(t)})^T\boldsymbol{\gamma}^{-1} + \nabla^T_{\boldsymbol{\Lambda}_{\cdot j}^{-1}}\mathrm{diag}(\boldsymbol{\Lambda}_{\cdot j}^{-1}) + C_2.
\end{aligned}$$

Here we use the superscript $^{(t)}$ to indicate the previous estimate.

$$W(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T, \boldsymbol{\Lambda}_{\cdot j}) \triangleq \beta^{-1}\begin{bmatrix}\boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{I}\end{bmatrix} + \begin{bmatrix}(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T)^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda}_{\cdot j}^{-1}\end{bmatrix}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{M}_j}^{(t)} \triangleq \boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}^{(t)}\boldsymbol{A}_{\cdot j}^T + \boldsymbol{\Lambda}_{\cdot j}^{(t)} + \beta\boldsymbol{I}$$

$$\nabla_{(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T)^{-1}} \triangleq \frac{\partial \log|W(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T,\boldsymbol{\Lambda}_{\cdot j})|}{\partial(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T)^{-1}}\Big|_{(\boldsymbol{\Gamma},\boldsymbol{\Lambda}_{\cdot j})=(\boldsymbol{\Gamma}^{(t)},\boldsymbol{\Lambda}_{\cdot j}^{(t)})} = \boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}^{(t)}\boldsymbol{A}_{\cdot j}^T - (\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}^{(t)}\boldsymbol{A}_{\cdot j}^T)(\boldsymbol{\Sigma}_{\boldsymbol{M}_j}^{(t)})^{-1}(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}^{(t)}\boldsymbol{A}_{\cdot j}^T)$$

$$\nabla_{\boldsymbol{\Lambda}_{\cdot j}^{-1}} \triangleq \operatorname{diag}\left[\frac{\partial \log|W(\boldsymbol{A}_{\cdot j}\boldsymbol{\Gamma}\boldsymbol{A}_{\cdot j}^T,\boldsymbol{\Lambda}_{\cdot j})|}{\partial \boldsymbol{\Lambda}_{\cdot j}^{-1}}\Big|_{(\boldsymbol{\Gamma},\boldsymbol{\Lambda}_{\cdot j})=(\boldsymbol{\Gamma}^{(t)},\boldsymbol{\Lambda}_{\cdot j}^{(t)})}\right] = \operatorname{diag}[\boldsymbol{\Lambda}_{\cdot j}^{(t)} - \boldsymbol{\Lambda}_{\cdot j}^{(t)}(\boldsymbol{\Sigma}_{\boldsymbol{M}_j}^{(t)})^{-1}\boldsymbol{\Lambda}_{\cdot j}^{(t)}] \triangleq \operatorname{diag}(\boldsymbol{\Sigma}_{\boldsymbol{E}_j|\boldsymbol{M}_j}^{(t)})$$

In the second inequality, we relax the upper bound, which empirically leads to better performance. And it holds because the term $\nabla_{\boldsymbol{\Lambda}_{\cdot j}^{-1}}^T \operatorname{diag}(\boldsymbol{\Lambda}_{\cdot j}^{-1})$ is non-negative.

Similarly, for the third term, we have

$$\begin{aligned}
&0.5\log|\boldsymbol{A}_{i\cdot}\boldsymbol{\Gamma}\boldsymbol{A}_{i\cdot}^T + \boldsymbol{\Lambda}_{i\cdot} + \beta\boldsymbol{I}| \\
\leq& 0.5\log|\boldsymbol{\Gamma}| + 0.5\log|\boldsymbol{\Lambda}_{i\cdot}| + 0.5\operatorname{diag}(\boldsymbol{\Gamma}^{(t)} - \boldsymbol{\Gamma}^{(t)}\boldsymbol{A}_{i\cdot}^T(\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}^{(t)})^{-1}\boldsymbol{A}_{i\cdot}\boldsymbol{\Gamma}^{(t)})^T\boldsymbol{\gamma}^{-1} + \nabla_{\boldsymbol{\Lambda}_{i\cdot}^{-1}}^T\operatorname{diag}(\boldsymbol{\Lambda}_{i\cdot}^{-1}) + C_3,
\end{aligned}$$

where $\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}^{(t)} \triangleq \boldsymbol{A}_{i\cdot}\boldsymbol{\Gamma}^{(t)}\boldsymbol{A}_{i\cdot}^T + \boldsymbol{\Lambda}_{i\cdot}^{(t)} + \beta\boldsymbol{I}$, $\nabla_{\boldsymbol{\Lambda}_{i\cdot}^{-1}} \triangleq \operatorname{diag}\left[\frac{\partial \log|W(\boldsymbol{A}_{i\cdot}\boldsymbol{\Gamma}\boldsymbol{A}_{i\cdot}^T,\boldsymbol{\Lambda}_{i\cdot})|}{\partial \boldsymbol{\Lambda}_{i\cdot}^{-1}}\Big|_{(\boldsymbol{\Gamma},\boldsymbol{\Lambda}_{i\cdot})=(\boldsymbol{\Gamma}^{(t)},\boldsymbol{\Lambda}_{i\cdot}^{(t)})}\right] = \operatorname{diag}[\boldsymbol{\Lambda}_{i\cdot}^{(t)} - \boldsymbol{\Lambda}_{i\cdot}^{(t)}(\boldsymbol{\Sigma}_{(\boldsymbol{M}^T)_i}^{(t)})^{-1}\boldsymbol{\Lambda}_{i\cdot}^{(t)}] \triangleq \operatorname{diag}(\boldsymbol{\Sigma}_{(\boldsymbol{E}^T)_i|(\boldsymbol{M}^T)_i}^{(t)})$.

4.7.2 Proof of Proposition 4.2

Proof. Since $d \in [\operatorname{rank}(\boldsymbol{L}_{opt}),\ \min(n_1, n_2)]$, it's not hard to verify that the optimization problems in (4.1) and (4.2) have the same minimal objective value and global optimal solution(s) in terms of $\boldsymbol{L}$ and $\boldsymbol{E}$. So we focus on building the connection between (4.1), (4.2) and (4.8). Consider the following equivalent optimization problem to that of (4.8):

$$\begin{aligned}
\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{s},\boldsymbol{E}} \|\boldsymbol{s}\|_0 + \lambda\|\boldsymbol{E}\|_0 \quad s.t.\ \|\boldsymbol{M} - \boldsymbol{U}\operatorname{diag}(\boldsymbol{s})\boldsymbol{V}^T - \boldsymbol{E}\|_F \leq \delta, \\
\|\boldsymbol{U}_i\|_2 = \|\boldsymbol{V}_i\|_2 = 1, \forall i, \boldsymbol{U} \in \mathbb{R}^{n_1\times d}, \boldsymbol{V} \in \mathbb{R}^{n_2\times d}.
\end{aligned} \tag{4.25}$$

Denote f_5 as the minimal objective value of (4.2), f_7 as the minimal objective value of (4.25). We will first prove that $f_5 = f_7$.

Since (4.2) has additional constraints than (4.25), we must have $f_5 \geq f_7$.

Let $(\boldsymbol{U}^*, \boldsymbol{V}^*, \boldsymbol{s}^*, \boldsymbol{E}^*)$ be any global minima of (4.25), so $\|\boldsymbol{s}^*\|_0 + \lambda\|\boldsymbol{E}^*\|_0 = f_7$. By performing **rank-d** SVD on $(\boldsymbol{U}^*\operatorname{diag}(\boldsymbol{s}^*)\boldsymbol{V}^{*T})$ to get $\tilde{\boldsymbol{U}}$, $\tilde{\boldsymbol{s}}$, and $\tilde{\boldsymbol{V}}$, we must have $\tilde{\boldsymbol{s}} \geq 0, \|\tilde{\boldsymbol{s}}\|_0 \leq \|\boldsymbol{s}^*\|_0$, and $\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{V}}$ orthonormal. So $(\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{V}}, \tilde{\boldsymbol{s}}, \boldsymbol{E}^*)$ is a feasible solution of (4.2) and thus

$f_5 \leq \|\tilde{\boldsymbol{s}}\|_0 + \lambda\|\boldsymbol{E}^*\|_0 \leq \|\boldsymbol{s}^*\|_0 + \lambda\|\boldsymbol{E}^*\|_0 = f_7$.

In sum, we must have $f_5 = f_7$. So (4.1), (4.2) and (4.25) share the same minimal objective value, denoted as f_*.

Then it remains to show (4.2) and (4.25) have the same global optimal solution(s) in terms of $\boldsymbol{L}(= \boldsymbol{U}\text{diag}(\boldsymbol{s})\boldsymbol{V}^T)$ and $\boldsymbol{E}$.

Let the pair $(\boldsymbol{L}^{\#} = \boldsymbol{U}^{\#}\text{diag}(\boldsymbol{s}^{\#})\boldsymbol{V}^{\#^T}, \boldsymbol{E}^{\#})$ be *any* global optimal solution of (4.2), we must have $\|\boldsymbol{s}^{\#}\|_0 + \lambda\|\boldsymbol{E}^{\#}\|_0 = f_*$. Again, since (4.2) has additional constraints than (4.25), $(\boldsymbol{L}^{\#} = \boldsymbol{U}^{\#}\text{diag}(\boldsymbol{s}^{\#})\boldsymbol{V}^{\#^T}, \boldsymbol{E}^{\#})$ must be a feasible solution of (4.25). Further, we can claim it is also the global optimal solution of (4.25), since $\|\boldsymbol{s}^{\#}\|_0 + \lambda\|\boldsymbol{E}^{\#}\|_0$ achieves the minimal objective value f_*. So we have proved that *any* global optimal solution of (4.2) must also be the global optimal solution of (4.25).

Now let us prove the opposite direction. Let the pair $(\boldsymbol{L}^* = \boldsymbol{U}^*\text{diag}(\boldsymbol{s}^*)\boldsymbol{V}^{*^T}, \boldsymbol{E}^*)$ be *any* global optimal solution of (4.25), we must have $\|\boldsymbol{s}^*\|_0 + \lambda\|\boldsymbol{E}^*\|_0 = f_*$. Note that $(\boldsymbol{L}^*, \boldsymbol{E}^*)$ is a feasible solution of (4.1) and $\text{rank}(\boldsymbol{L}^*) \leq \|\boldsymbol{s}^*\|_0$, so $\text{rank}(\boldsymbol{L}^*) + \lambda\|\boldsymbol{E}^*\|_0 \leq \|\boldsymbol{s}^*\|_0 + \lambda\|\boldsymbol{E}^*\|_0 = f_*$. Since f_* is the minimal objective value of (4.1), then $(\boldsymbol{L}^*, \boldsymbol{E}^*)$ is actually the global optimal solution of (4.1). So we proved that *any* global optimal solution of (4.25) must also be the global optimal solution of (4.1). As (4.1) and (4.2) have the same global optimal solution(s) in terms of $\boldsymbol{L}(= \boldsymbol{U}\text{diag}(\boldsymbol{s})\boldsymbol{V}^T)$ and $\boldsymbol{E}$, we further have *any* global optimal solution $(\boldsymbol{L}^*, \boldsymbol{E}^*)$ of (4.25) must also be the global optimal solution of (4.2).

In sum, (4.2) and (4.25) have the same global optimal solution(s) in terms of the low rank matrix $\boldsymbol{L}$ and the sparse matrix $\boldsymbol{E}$, where $\boldsymbol{L} = \boldsymbol{U}\text{diag}(\boldsymbol{s})\boldsymbol{V}^T$ in (4.2) and (4.25).

4.7.3 Proof for the Optimal Solution of (4.19)

The optimization problem is as follows:

$$\arg \min_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \|\boldsymbol{B} - \boldsymbol{u}\boldsymbol{v}^T\|_F^2. \tag{4.26}$$

The *optimal* solution is given by $\hat{\boldsymbol{u}} = \boldsymbol{U}_1$, $\hat{\boldsymbol{v}} = \boldsymbol{V}_1$, where $\boldsymbol{U}_1$ and $\boldsymbol{V}_1$ are the first singular vector pair of $\boldsymbol{B}$ that correspond to its largest singular value.

Proof.

$$\begin{aligned}
&\arg \min_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \|\boldsymbol{B} - \boldsymbol{u}\boldsymbol{v}^T\|_F^2 \\
=&\arg \min_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \text{Trace}\{(\boldsymbol{B} - \boldsymbol{u}\boldsymbol{v}^T)^T(\boldsymbol{B} - \boldsymbol{u}\boldsymbol{v}^T)\} \\
=&\arg \min_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \text{Trace}\{\boldsymbol{B}^T\boldsymbol{B} - 2\boldsymbol{v}\boldsymbol{u}^T\boldsymbol{B} + \boldsymbol{v}\boldsymbol{u}^T\boldsymbol{u}\boldsymbol{v}^T\} \\
=&\arg \min_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \text{Trace}\{-2\boldsymbol{v}\boldsymbol{u}^T\boldsymbol{B} + \boldsymbol{v}\boldsymbol{v}^T\} \\
=&\arg \min_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \text{Trace}\{-2\boldsymbol{v}\boldsymbol{u}^T\boldsymbol{B}\} + \text{Trace}\{\boldsymbol{v}\boldsymbol{v}^T\} \\
=&\arg \min_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \text{Trace}\{-2\boldsymbol{v}\boldsymbol{u}^T\boldsymbol{B}\} + \text{Trace}\{\boldsymbol{v}^T\boldsymbol{v}\} \\
=&\arg \min_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \text{Trace}\{-2\boldsymbol{v}\boldsymbol{u}^T\boldsymbol{B}\} + 1 \\
=&\arg \max_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \text{Trace}\{\boldsymbol{v}\boldsymbol{u}^T\boldsymbol{B}\} \\
=&\arg \max_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \text{Trace}\{\boldsymbol{u}^T\boldsymbol{B}\boldsymbol{v}\}
\end{aligned}$$

$$= \arg \max_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \boldsymbol{u}^T \boldsymbol{B} \boldsymbol{v} \tag{4.27}$$

Since

$$\max_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \boldsymbol{u}^T \boldsymbol{B} \boldsymbol{v} \leq \max_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \|\boldsymbol{u}\|_2 \|\boldsymbol{B}\boldsymbol{v}\|_2 = \max_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \|\boldsymbol{B}\boldsymbol{v}\|_2 \leq \max_{\substack{\boldsymbol{u},\boldsymbol{v} \\ \|\boldsymbol{u}\|_2=1 \\ \|\boldsymbol{v}\|_2=1}} \|\boldsymbol{B}\| \|\boldsymbol{v}\|_2 = \|\boldsymbol{B}\| \tag{4.28}$$

Apparently, the maximum value $\|\boldsymbol{B}\| (= \sigma_1)$ can be achieved by setting $\boldsymbol{u} = \boldsymbol{U}_1, \boldsymbol{v} = \boldsymbol{V}_1$ in (4.27).

Chapter 5

Future Work

The propose ℓ_0 regularization framework has numerous applications in many data analysis and machine learning problems, both linear and non-linear. In this chapter, we briefly mention a few important problems where the proposed framework can be directly applied to.

5.1 Robust Tensor Decomposition

A tensor is an N-way array. Decompositions of higher-order tensors (i.e., N-way arrays with $N \geq 3$) have applications in psycho-metrics, chemometrics, signal processing, numerical linear algebra, computer vision, numerical analysis, data mining, neuroscience, graph analysis, and elsewhere [108]. The classic Tucker decomposition and CANDECOMP/PARAFAC (CP) decomposition methods are based on the Least Squares criteria and thus suffer from the outliers in the data. Recently, researchers extended the idea of Robust PCA to the tensor version (e.g., [109]), where the ℓ_1-norm is again used to encourage the sparseness of the outliers. The ℓ_0 regularization framework proposed in this thesis can be straightforwardly applied to this problem, where the genuine ℓ_0-'norm' enforces the sparseness of the outliers and the ℓ_1-norm addresses the inlier noise. And it is very promising to achieve better performance and with

better recovery guarantees than the existing approach.

5.2 Robust Matrix Sensing

Recovering a structured matrix from a small number of linear measurements also has numerous applications [110]. In many modern applications, the measurements are often corrupted by the outliers, e.g., due to the interference or the failure of some sensors. Mathematically, we observe $y = \mathcal{A}(X) + e + n$, where e is the outlier corruption vector, n models the inlier noise, X is some structured matrix (e.g., low-rank) to be recovered, and $\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ is a known linear operator. The proposed ℓ_0 regularization framework can also be straightforwardly applied to address this problem, e.g., via the objective $f(X) + \alpha\|e\|_0 + \|y - \mathcal{A}(X) - e\|_1$. Superior performance can be expected over the existing approach [111] in robust low-rank matrix sensing, which purely relies on the ℓ_1-norm to model the outliers. On the theoretical side, recall that in Robust Linear Regression, we discovered an important slowly decreasing property of $m(A)$, which measures the resilience to the maximum number of the outliers. It's quite promising to find a similar property of $m(\mathcal{A})$ here, which would then allow us to answer some fundamental questions and establish better recovery guarantees than the existing methods.

5.3 Robust Deep Autoencoders

Deep Autoencoders play a fundamental role in deep learning. It can be viewed as a non-linear version of PCA and suffers from the outliers in the training data. Recently, the idea of Robust PCA is borrowed to address this challenge [112]. Mathematically, they propose to

solve the following problem:

$$\min_{\boldsymbol{L},\boldsymbol{E},\theta} \|\boldsymbol{M} - \boldsymbol{E} - D_\theta(E_\theta(\boldsymbol{L}))\|_F + \lambda\|\boldsymbol{E}\|_1 \tag{5.1}$$

where $E_\theta(\cdot)$ denotes an encoder, and $D_\theta(\cdot)$ denotes a decoder, $\boldsymbol{M}$ is the outlier corrupted training data, $\boldsymbol{E}$ models the outlier corruptions, and $\boldsymbol{L}$ is the underlying clean data. The ℓ_1-norm is used to enforce the sparseness of the outliers.

The proposed ℓ_0 regularization framework can also be straightforwardly applied to address this problem, e.g., via the following objective:

$$\min_{\boldsymbol{L},\boldsymbol{E},\theta} \|\boldsymbol{M} - \boldsymbol{E} - D_\theta(E_\theta(\boldsymbol{L}))\|_1 + \lambda\|\boldsymbol{E}\|_0 \tag{5.2}$$

where we directly use the genuine ℓ_0-'norm' to enforce the sparseness of the outliers, and use ℓ_1-norm to address the inlier noise. And the optimization procedure can be similar to that of [112]. It's very promising to achieve much better performance as in the Robust PCA case, and with convergence guarantee.

Bibliography

[1] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery," *IEEE Signal Processing Magazine*, vol. 35, no. 4, pp. 32–55, Jul. 2018.

[2] A. Tajer, V. V. Veeravalli, and H. V. Poor, "Outlying sequence detection in large data sets: A data-driven approach," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 44–56, Sep. 2014.

[3] G. Papageorgiou, P. Bouboulis, and S. Theodoridis, "Robust linear regression analysis—a greedy approach," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 3872–3887, 2015.

[4] P. J. Huber, "The 1972 wald lecture robust statistics: A review," *The Annals of Mathematical Statistics*, vol. 43, no. 4, pp. 1041–1067, 1972.

[5] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005, vol. 589.

[6] R. A. Maronna, R. D. Martin, V. J. Yohai, and M. Salibián-Barrera, *Robust statistics: theory and methods (with R)*. Wiley, 2018.

[7] P. J. Huber, *Robust statistics*. Springer, 2011.

[8] P. J. Rousseeuw, "Least median of squares regression," *Journal of the American statistical association*, vol. 79, no. 388, pp. 871–880, 1984.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] E. J. Candes and P. A. Randall, "Highly robust error correction byconvex programming," *IEEE Transactions on Information Theory*, vol. 54, no. 7, pp. 2829–2840, 2008.

[11] Y. She and A. B. Owen, "Outlier detection using nonconvex penalized regression," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 626–639, 2011.

[12] K. Bhatia, P. Jain, and P. Kar, "Robust regression via hard thresholding," in *Advances in Neural Information Processing Systems*, 2015, pp. 721–729.

[13] E. Candes and T. Tao, "Decoding by linear programming," *arXiv preprint math/0502327*, 2005.

[14] J.-J. Fuchs, "An inverse problem approach to robust regression," in *1999 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1999, pp. 1809–1812.

[15] K. Mitra, A. Veeraraghavan, and R. Chellappa, "Robust regression using sparse learning for high dimensional parameter estimation problems," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2010, pp. 3846–3849.

[16] 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, Mar. 2013.

[17] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of machine learning research*, vol. 1, pp. 211–244, Jun. 2001.

[18] 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.

[19] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 8, pp. 1207–1223, 2006.

[20] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE transactions on information theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[21] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on information theory*, vol. 52, no. 1, pp. 6–18, 2006.

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

[23] V. Kekatos and G. B. Giannakis, "From sparse signals to sparse residuals for robust sensing," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3355–3368, 2011.

[24] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted 1 minimization," *Journal of Fourier analysis and applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

[25] M. Fazel, "Matrix rank minimization with applications [ph. d. thesis]," *Elec. Eng. Dept, Stanford University*, 2002.

[26] M. S. Lobo, M. Fazel, and S. Boyd, "Portfolio optimization with linear and fixed transaction costs," *Annals of Operations Research*, vol. 152, no. 1, pp. 341–365, 2007.

[27] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2008, pp. 3869–3872.

[28] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar conference on signals, systems and computers*, IEEE, 1993, pp. 40–44.

[29] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[30] N. A. Campbell, "Robust procedures in multivariate analysis i: Robust covariance estimation," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 29, no. 3, pp. 231–237, 1980.

[31] F. De la Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, no. 1, pp. 117–142, Aug. 2003.

[32] K. R. Gabriel and S. Zamir, "Lower rank approximation of matrices by least squares with any choice of weights," *Technometrics*, vol. 21, no. 4, pp. 489–498, 1979.

[33] C. Croux and P. Filzmoser, "Robust factorization of a data matrix," in *COMPSTAT*, R. Payne and P. Green, Eds., Heidelberg: Physica-Verlag HD, 1998, pp. 245–250.

[34] D. Skočaj, H. Bischof, and A. Leonardis, "A robust pca algorithm for building representations from panoramic images," in *European Conference on Computer Vision*, Springer, 2002, pp. 761–775.

[35] K. R. Gabriel and C. L. Odoroff, *Resistant lower rank approximation of matrices*. University of Rochester, 1984.

[36] R. He, T. Tan, and L. Wang, "Robust recovery of corrupted low-rankmatrix by implicit regularizers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 770–783, Apr. 2014.

[37] L. Xu and A. L. Yuille, "Robust principal component analysis by self-organizing rules based on statistical physics approach," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 131–143, 1995.

[38] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.

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

[40] 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. eprint: http://dx.doi.org/10.1137/090761793.

[41] 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.

[42] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Computer Science Review*, vol. 23, pp. 1–71, 2017.

[43] Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu, "Low-rank structure learning via nonconvex heuristic recovery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 383–396, Mar. 2013.

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

[45] D. Hsu, S. M. Kakade, and T. Zhang, "Robust matrix decomposition with sparse corruptions," *IEEE Trans. Inf. Theory*, vol. 57, no. 11, pp. 7221–7234, Nov. 2011.

[46] R. Chartrand, "Nonconvex splitting for regularized low-rank + sparse decomposition," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5810–5819, Nov. 2012.

[47] Q. Sun, S. Xiang, and J. Ye, "Robust principal component analysis via capped norms," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13, Chicago, Illinois, USA: ACM, 2013, pp. 311–319.

[48] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *International Conference on Machine Learning*, 2011, pp. 33–40.

[49] L. Xiong, X. Chen, and J. Schneider, "Direct robust matrix factorizatoin for anomaly detection," in *2011 IEEE 11th International Conference on Data Mining*, Dec. 2011, pp. 844–853.

[50] M. O. Ulfarsson, V. Solo, and G. Marjanovic, "Sparse and low rank decomposition using ℓ_0 penalty," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 3312–3316.

[51] 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.

[52] 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*, vol. 3, Jun. 2003, pp. 2156–2162.

[53] 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.

[54] R. Giri and B. Rao, "Type I and Type II bayesian methods for sparse signal recovery using scale mixtures," *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3418–3428, Jul. 2016.

[55] 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.

[56] 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.

[57] 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.

[58] C. Aicher, "A variational bayes approach to robust principal component analysis," Santa Fe Institute, University of Colorado Boulder, Tech. Rep., 2013.

[59] N. Han, Y. Song, and Z. Song, "Bayesian robust principal component analysis with structured sparse component," *Computational Statistics & Data Analysis*, vol. 109, pp. 144–158, 2017.

[60] 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, Catalina Island, CA, 2012, pp. 914–923.

[61] M. Sundin, S. Chatterjee, and M. Jansson, "Bayesian learning for robust principal component analysis," in *23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 2361–2365.

[62] 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.

[63] J. N. Laska, M. A. Davenport, and R. G. Baraniuk, "Exact signal recovery from sparsely corrupted measurements through the pursuit of justice," in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, IEEE, 2009, pp. 1556–1560.

[64] J. Wright and Y. Ma, "Dense error correction via ℓ_1-minimization," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3540–3560, 2010.

[65] N. H. Nguyen and T. D. Tran, "Exact recoverability from dense corrupted observations via ℓ_1-minimization," *IEEE transactions on information theory*, vol. 59, no. 4, pp. 2017–2035, 2013.

[66] N. Nguyen and T. D. Tran, "Robust lasso with missing and grossly corrupted observations," *IEEE transactions on information theory*, vol. 59, no. 4, pp. 2036–2058, 2013.

[67] C. Studer and R. G. Baraniuk, "Stable restoration and separation of approximately sparse signals," *Applied and Computational Harmonic Analysis*, vol. 37, no. 1, pp. 12–35, 2014.

[68] C. Studer, P. Kuppinger, G. Pope, and H. Bolcskei, "Recovery of sparsely corrupted signals," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3115–3130, 2012.

[69] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *arXiv preprint math/0409186*, 2004.

[70] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[71] A. Giloni and M. Padberg, "Alternative methods of linear regression," *Mathematical and Computer Modelling*, vol. 35, no. 3-4, pp. 361–374, 2002.

[72] S. Flores, "Sharp non-asymptotic performance bounds for ℓ_1 and huber robust regression estimators," *Test*, vol. 24, no. 4, pp. 796–812, 2015.

[73] E. Candes and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *The annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, 2007.

[74] S. A. Flores, "Robustness of ℓ_1-norm estimation: From folklore to fact," *IEEE Signal Processing Letters*, vol. 25, no. 11, pp. 1640–1644, Nov. 2018.

[75] Y. Sharon, J. Wright, and Y. Ma, "Minimum sum of distances estimator: Robustness and stability," in *2009 American Control Conference*, IEEE, 2009, pp. 524–530.

[76] P. J. Rousseeuw and B. C. Van Zomeren, "Unmasking multivariate outliers and leverage points," *Journal of the American Statistical association*, vol. 85, no. 411, pp. 633–639, 1990.

[77] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[78] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear algebra and its applications*, vol. 284, no. 1-3, pp. 193–228, 1998.

[79] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics-theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[80] W. Dumouchel and F. O'Brien, "Integrating a robust option into a multiple regression computing environment," in *Computing and graphics in statistics*, Springer-Verlag New York, Inc., 1992, pp. 41–48.

[81] A. E. Beaton and J. W. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," *Technometrics*, vol. 16, no. 2, pp. 147–185, 1974.

[82] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011, vol. 196.

[83] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[84] Y. Wang and W. Yin, "Sparse signal reconstruction via iterative support detection," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 462–491, 2010.

[85] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, no. 1, pp. 1–20, 2000.

[86] J. Liu, P. C. Cosman, and B. D. Rao, "Sparsity regularized principal component pursuit," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 4431–4435.

[87] J. Liu, P. C. Cosman, and B. D. Rao, "Robust linear regression via ℓ_0 regularization," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 698–713, Feb. 2018.

[88] A. Petukhov and I. Kozlov, "Greedy approach for low-rank matrix recovery," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2013, p. 1.

[89] F. Shang, Y. Liu, J. Cheng, and H. Cheng, "Robust principal component analysis with missing data," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, ser. CIKM '14, Shanghai, China: ACM, 2014, pp. 1149–1158.

[90] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, "Low-rank matrix recovery from errors and erasures," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4324–4337, Jul. 2013.

[91] X. Li, "Compressed sensing and matrix completion with constant proportion of corruptions," *Constructive Approximation*, vol. 37, no. 1, pp. 73–99, Feb. 2013.

[92] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Comput.*, vol. 15, no. 4, pp. 915–936, Apr. 2003.

[93] L. Han and Q. Zhang, "Multi-stage convex relaxation method for low-rank and sparse matrix separation problem," *Appl. Math. Comput.*, vol. 284, no. Supplement C, pp. 175–184, 2016.

[94] 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.

[95] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, Dec. 2008.

[96] 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.

[97] Q. Sun, S. Xiang, and J. Ye, "Robust principal component analysis via capped norms," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13, Chicago, Illinois, USA: ACM, 2013, pp. 311–319.

[98] 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.

[99] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, p. 717, Apr. 2009.

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

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

[102] 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.

[103] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust pca via gradient descent," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 4152–4160.

[104] 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.

[105] J. Liu and B. D. Rao, "Robust pca via $\ell 0$-$\ell 1$regularization," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 535–549, Jan. 2019.

[106] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *Journal of machine learning research*, vol. 11, pp. 2287–2322, Aug. 2010.

[107] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st. Springer, 2010.

[108] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[109] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5249–5257.

[110] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 608–622, Jun. 2016.

[111] X. Li, Z. Zhu, A. M.-C. So, and R. Vidal, "Nonconvex robust low-rank matrix recovery," *arXiv preprint arXiv:1809.09237*, 2018.

[112] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2017, pp. 665–674.