ROBUST ALGORITHMS ON LOW-RANK APPROXIMATION AND THEIR APPLICATIONS

By

Ningyu Sha

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computational Mathematics, Science and Engineering – Doctor of Philosophy
Statistics – Dual Major

2021

ABSTRACT

ROBUST ALGORITHMS ON LOW-RANK APPROXIMATION AND THEIR APPLICATIONS

By

Ningyu Sha

Low-rank approximation models have been widely developed in computer vision, image analysis, signal processing, web data analysis, bioinformatics, etc. Generally, we assume that the intrinsic data lies in a low-dimensional subspace, and we need to extract the low-rank representation given observations. There are many well-known works such as Principal Component Analysis (PCA), factor analysis, least squares, etc. However, their performance may be affected when dealing with outliers. Robust PCA (RPCA) plays an important role in such cases, but RPCA based methods suffer from expensive computation costs. In this thesis, we discussed how to improve the performance of RPCA in terms of both speed and accuracy. The comparison between convex and non-convex models is also discussed. Notably, we propose a theory about matrix decomposition with unknown rank. A nonlinear RPCA approach is also proposed, given the assumption that data lie on a manifold. Then, we take examples from seismic event detection and 2D image denoising. The numerical experiments show the robustness of our techniques and present speedup and higher recovery accuracy compared with existing approaches.

It is usually common in practice that observed data has missing values. So, we need to make a low-rank approximation based on incomplete data. Also, it may take a long time for offline matrix completion since we need to collect all data first. The online version can offer up-to-date results based on a continuous data stream. Online matrix completion has applications in computer vision and web data analysis, especially in video image transmission and recommendation systems. To be better applied on color images with three channels, we introduced online quaternion matrix completion. We can get an updated result for every new observed entry using stochastic gradient descent on the quaternion matrix.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor and committee chair, Dr. Ming Yan. He is very patient and knowledgeable. He describes a big picture about optimization for me. Whenever I have any difficulty or questions, he is more than willing to help. I also want to thank my co-advisor, Dr. Yuying Xie. He is humorous and warm-hearted. He gives me precious opinions from statistics direction.

I would also like to thank my other committee members, Dr. Matthew Hirn, Dr. Yuehua Cui, and Dr. Haolei Weng, for their knowledgeable feedback and kind suggestions for my research life.

I want to thank our research partners, Dr. Youzuo Lin, Dr. Lei Shi, Dr. Rongrong Wang, He Lye, and Shuyang Qin. Thank you for the brilliant work on our projects.

Also, I would like to thank my friends Yuning Hao, Hongnan Wang, Yun Song, Yuejiao Sun, Lijiang Xu, Jieqian He, Binbin Huang, Hao Wang, Jialin Qu, Peide Li, Mi Hu, Ken Lee, Runze Su, Yao Xuan and Yao Li for giving me huge support and companion during my Ph.D.

I also want to thank the CMSE community, Lisa Roy, Heather Williams, Dr. Andrew Christlieb, etc. They are always there to offer help. I also want to thank Erica Schmittdiel, who stands with me through the darkest time.

Most importantly, I would like to thank my parents, Chunhua Sha and Hui Peng. Their selfless love is continued support to make me a better person, always.

I also need to thank a lot of people, even if I have forgotten their names.

TABLE OF CONTENTS

LIST OF TABLES vii

LIST OF FIGURES viii

LIST OF ALGORITHMS x

CHAPTER 1 BACKGROUND 1
1.1 Applications of robust low-rank optimization 1
1.2 Existing work on RPCA 5
1.3 Overview of this thesis 9

CHAPTER 2 ROBUST PRINCIPAL COMPONENT ANALYSIS FOR LOW RANK MATRIX APPROXIMATION 11
2.1 Introduction 11
2.1.1 Notation 14
2.1.2 Organization 15
2.2 Proposed algorithms 15
2.2.1 Forward-backward 20
2.2.1.1 Convergence analysis 21
2.2.2 An accelerated algorithm 27
2.3 Numerical experiments 27
2.3.1 Synthetic data 28
2.3.1.1 Low-rank matrix recovery 29
2.3.1.2 Robustness of the model 30
2.3.1.3 Low-rank matrix recovery with missing entries 31
2.3.2 Real image experiment 32
2.4 Concluding remarks 35
2.4.1 Nonconvex penalties on the singular values 35
2.4.2 Other regularization on the sparse component 36
2.4.3 Constrained problems 36

CHAPTER 3 ROBUST PRINCIPAL COMPONENT ANALYSIS FOR SEISMIC EVENT DETECTION 38
3.1 Introduction 38
3.2 Theory 39
3.2.1 New algorithms with infimal convolution 41
3.2.1.1 Comparison between PGM and IC-PGM 42
3.3 Results 43
3.3.1 Synthetic seismic data 43
3.3.2 Field seismic data 45
3.4 Conclusion 47

CHAPTER 4 MANIFOLD DENOISING BY NONLINEAR ROBUST PRINCIPAL COMPONENT ANALYSIS 49
4.1 Introduction 49
4.2 Methodology 50
4.3 Geometric explanation 52
4.4 Optimization algorithm 53
4.5 Numerical experiments 54
4.6 Conclusion 56

CHAPTER 5 ONLINE MATRIX COMPLETION WITH QUATERNION MATRIX 58
5.1 Introduction 58
5.2 Introduction on Quaternion Matrices 60
5.2.1 Quaternion Numbers 60
5.2.2 Basic Properties 61
5.2.3 Singular Value Decomposition 62
5.2.4 Incoherence Condition 63
5.2.5 Sampling Scheme 63
5.3 Online Matrix Completion Algorithms and its Theoretical Analysis 63
5.4 Algorithms 82
5.4.1 The Hermitian Case 82
5.4.2 The General Case 82
5.5 Numerical Experiments 83
5.6 Conclusion 85

BIBLIOGRAPHY 87

LIST OF TABLES

Table 2.1: Comparison of three RPCA algorithms. We compare the relative error of their solutions to the true low-rank matrix and the number of iterations. Both Alg. 2.1 and Alg. 2.2 have better performance than (Shen et al., 2019) in terms of the relative error and the number of iterations. Alg. 2.2 has the fewest iterations but the relative error could be large. It is because the true low-rank matrix is not the optimal solution to the optimization problem, and the trajectory of the iterations moves close to $\mathbf{L}^\star$ before it approaches the optimal solution. 30

Table 2.2: Performance of Alg. 2.2 on low-rank matrix recovery with missing entries. We change the level of sparsity in the sparse noise, standard deviation of the Gaussian noise, and the ratio of missing entries. 32

Table 3.1: Comparison of six algorithms. IC-ADMM is the fastest, which is the same as synthetic data. The function value for MCP is smaller because of a different model. 46

LIST OF FIGURES

Figure 1.1: Image from https://link.medium.com/bAUJGpEl5hb. Height and weight are correlated. If visualized, these two vectors have an acute angle (left figure). After using PCA, height and weight are combined as a new feature 'size' (right figure). There is also 'other' information left that is orthogonal to 'size'. The short length means that it is less important. . . 2

Figure 1.2: Image from https://link.medium.com/BbeEPVIl5hb. PCA works well for clean linear data. However, it works poorly for data with outliers. . . 3

Figure 1.3: An application of RPCA. Image from (Zhou et al., 2014). Top row: four frames from a video. Middle row: video background, which is viewed as a low-rank approximation. Bottom row: main objects which correspond to sparse components. 4

Figure 1.4: Image from https://link.medium.com/ERplrHLl5hb. Rating system for Netflix films. Each row represents each user while each column represents each film. Values from 1 to 5 are scores. 5

Figure 2.1: The contour map of the relative error to $\mathbf{L}^\star$ for different parameters. In this experiment, we set $r = 25$ and $s = 20$. The upper bound of the rank is set to be $p = 30$. 30

Figure 2.2: The relative error to the true low-rank matrix vs the rank p for Shen et al.'s and Alg. 2.2. Alg. 2.2 is robust to p, as long as p is not smaller than the true rank 25. 31

Figure 2.3: The numerical experiment on the 'cameraman' image. (A-C) show that the proposed model performs better than Shen et al.'s both visually and in terms of RE and PSNR. (D) compares the objective values vs time for general SVD, Alg. 2.1, and Alg. 2.2. Here $f^\star$ is the value obtained by Alg. 2.2 with more iterations. It shows the fast speed with the Gauss-Newton approach and acceleration. With the Gauss-Newton approach, the computation time for Alg. 2.1 is reduced to about 1/7 of the one with standard SVD (from 65.11s to 8.43s). The accelerated Alg. 2.2 requires 5.2s, though the number of iterations is reduced from 3194 to 360. 33

Figure 2.4: The numerical experiment on the 'Barbara' image. (A-C) show that the proposed model performs better than Shen et al.'s both visually and in terms of RE and PSNR. (D) compares the objective values vs time for general SVD, Alg. 2.1, and Alg. 2.2. Here $f^\star$ is the value obtained by Alg. 2.2 with more iterations. It shows the fast speed with the Gauss-Newton approach and acceleration. With the Gauss-Newton approach, the computation time for Alg. 2.1 is reduced to less than 1/3 of the one with standard SVD (from 148.6s to 43.7s). The accelerated Alg. 2.2 requires 23.3s, though the number of iterations is reduced from 3210 to 300. 34

Figure 3.1: Comparison of recovered results on synthetic seismic data with 500 receivers and 1000 measurements at each receiver. (a) simulated clean data. (b) noisy data (-26.2 dB). (c) recovered data by L_1 (13.4 dB). (d) recovered data by MCP (13.9 dB). (e) recovered sparse noise by L_1. (f) recovered sparse noise by MCP. (g) the difference between the clean data and the recovered one by L_1. (h) the difference between the clean data and the recovered one by L_1. (e-h) zoom-in over receivers 150-350 and measurements 1-400. 44

Figure 3.2: Comparison of five algorithms (PGM, FISTA, IC-PGM, IC-FISTA, IC-ADMM) for the convex RPCA on synthetic data. IC-ADMM has the fastest convergence rate and smallest computational time. IC technique improves the performance of PGM and FISTA significantly. 45

Figure 3.3: Noisy data generated in Oklahoma. 46

Figure 3.4: Recovered results of the real data with two models. 47

Figure 4.1: NRPCA applied to the noisy Swiss roll data set. $\tilde{X} - \hat{S}$ is the result after subtracting the estimated sparse noise via NRPCA with $T = 1$; "$\tilde{X} - \hat{S}$ with one neighbor update" is that with $T = 2$, i.e., patches are reassigned once; $\hat{X}$ is the denoised data obtained via NRPCA with $T = 2$; "Patch-wise Robust PCA" refers to the ad-hoc application of the vanilla RPCA to each local patch independently, whose performance is clearly worse than the proposed joint-recovery formulation. 56

Figure 4.2: Laplacian eigenmaps and Isomap results for the original and the NR-PCA denoised digits 4 and 9 from the MNIST dataset. 57

Figure 5.1: A movie rating system. For a given $d_1 \times d_2$ low-rank matrix with missing entries, it can be factorized by a $d_1 \times k$ user matrix and a $k \times d_2$ item matrix where k is the rank of the original matrix. 59

Figure 5.2: Loss function value versus number of iterations for the small Hermitian case. The stepsize is $3e^{-5}$. Within 40000 iterations, the value decreases from nearly 100 to 10^{-6}. The loss function value tends to keep decreasing after these 40000 iterations. 84

Figure 5.3: Loss function value versus number of iterations for the small general quaternion matrix case. The stepsize is tuned to be $1e^{-4}$. Within 5000 iterations, the value decreases from around 100 to 10^{-4}. The loss function value tends to keep decreasing after these 5000 iterations. 85

Figure 5.4: Online image recovery result after 10000 iterations. We randomly sampled 1000o observations from (a). We can see that the result for recovered image (b) is not good. We expect that the difference between (a) and (b) can be as small as possible. 86

Figure 5.5: Loss function value versus number of iterations for the real color image. The initial stepsize is tuned to be $1e^{-5}$. For every 300 iterations, we multiply the stepsize by 0.95. The loss function value decreases from around 170 to almost 45. At the beginning, the loss function value decreases the most, and it tends to keep decreasing after these 10000 iterations. 86

LIST OF ALGORITHMS

2.1 RPCA for low rank matrix approximation 21
2.2 Accelerated RPCA with nonmonotone APG 28
4.1 Nonlinear RPCA 55
5.1 Online learning algorithm for the Hermitian matrix $\mathbf{M}$ 82
5.2 Online learning algorithm for general $\mathbf{M}$ (theoretical version) 83
5.3 Online learning algorithm for general $\mathbf{M}$ (practical version) 83

CHAPTER 1

BACKGROUND

1.1 Applications of robust low-rank optimization

Many high-dimensional data points can be represented as points in a low-dimensional subspace of a high-dimensional space, and principal component analysis (PCA) is a popular tool to find the low-dimensional subspace. Here, we use a simple example to explain the intuition behind PCA. Height and weight are two measurements (features) for football players, and we say that the original dimension of these measurements is two. That is, the data for each player lies in a two-dimensional space. Obviously, there is a positive correlation between these two measurements. If one person is taller than another one, he/she is usually heavier too. More specifically, these two measurements are linearly correlated, as shown in Fig. 1.1. In this example, we can use one new variable named size to approximately describe the combination of height and weight at a high accuracy. In fact, it is also applied to the size of clothes.

Generally, correlations within features happen when there are more than two features. Mathematically, we say that these features are not linearly independent. PCA finds a linear transform such that the new features after the transformation are linearly independent. In addition, PCA finds the most important features that represent the most information (variance) in the dataset. For example, the first principal component explains the most variance in the dataset, and the second principal component is the vector that is orthogonal to the first principal component and explains the second most variance in the dataset. PCA can be calculated from the singular value decomposition (SVD) of the data matrix, and the singular values represent the variance in the data corresponding to the principal components, which are corresponding right eigenvectors. Then, we keep the principal components corresponding to the largest singular values to represent most information using a

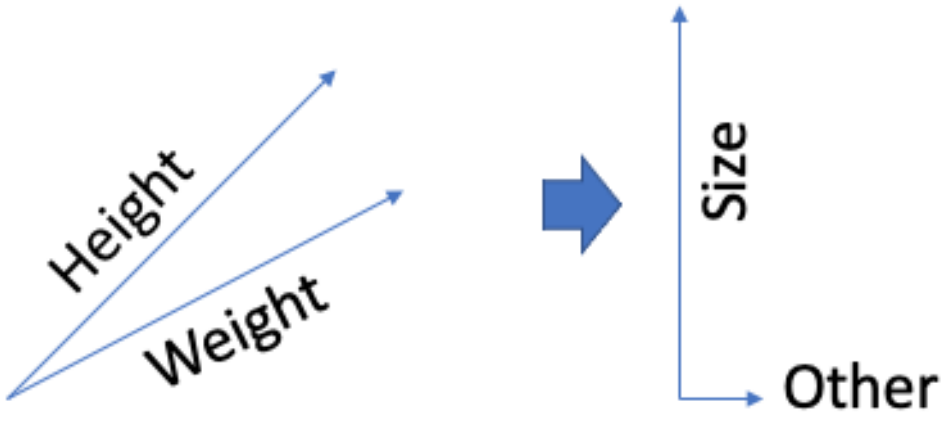
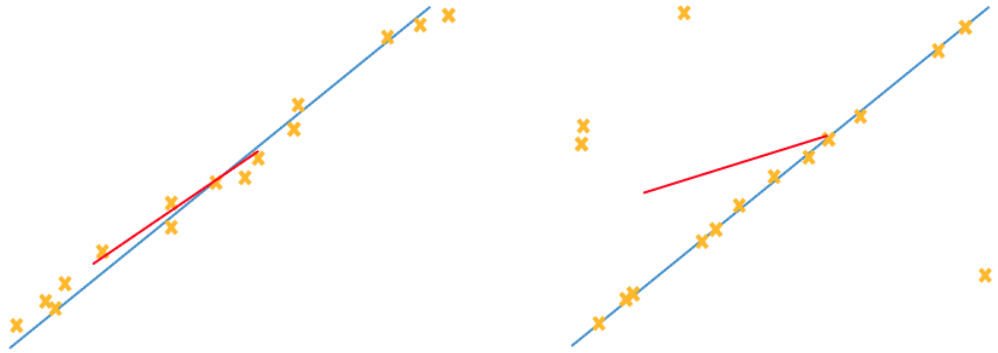


Figure 1.1: Image from https://link.medium.com/bAUJGpEl5hb. Height and weight are correlated. If visualized, these two vectors have an acute angle (left figure). After using PCA, height and weight are combined as a new feature 'size' (right figure). There is also 'other' information left that is orthogonal to 'size'. The short length means that it is less important.

small number of features. PCA has been applied to many different areas, such as computer vision, bioinformatics, finance, psychology, etc.

PCA removes the principal components corresponding to smallest singular values because the data points with small random distortions are not exactly on the low-dimensional space. The small random distortions are equivalent to adding small random values to all singular values. Therefore, by setting small singular values as zero, PCA can reduce the effect of these distortions. In the left figure of Figure 1.2, with the small distortions, PCA is able to get the first principal component, shown in red, and it is very close to the true direction shown in blue. However, in many real cases, there are outliers accompanied with the data set. In statistics, an outlier is a data point that differs significantly from other observations. It can be caused by equipment error or the population that has a heavy-tailed distribution. Outliers usually have large distance from normal data points, but the total number of outliers is much smaller than the total number of data points. In the case with outliers, PCA may not perform well. For example, in the right figure of Figure 1.2, there are four outliers, and PCA will give the red direction, which is very different from the true blue one.



(a) Yellow dots are sampled from one dimensional subspace (blue line) corrupted by Gaussian noise. The red line is the output of PCA. It can reserve most variance.

(b) Yellow dots are sampled from one dimensional subspace (blue line) corrupted by large sparse outliers. PCA is affected by outliers and can not capture the data well.

Figure 1.2: Image from https://link.medium.com/BbeEPVIl5hb. PCA works well for clean linear data. However, it works poorly for data with outliers.

Standard PCA can not deal with outliers because it is based on the assumption that the additional noise follows a Gaussian distribution. However, outliers follow heavy-tail distributions and a different model is required. Robust PCA (RPCA) is one approach to remove the outliers while preserving the low-dimensional structure. This approach has been successfully applied in a wide range of areas, including computer vision (De la Torre and Black, 2001), image processing (Liu et al., 2012; Elhamifar and Vidal, 2013), dimensionality reduction (Cunningham and Ghahramani, 2015), bioinformatics data analysis (Da Costa et al., 2009), and web data and services (Koren, 2009). More specifically, RPCA has achieved great success in video surveillance and face recognition (Candès et al., 2011; Bouwmans and Zahzah, 2014). Also, it has been proved by Candès et al. (2011) that the low-rank part can be a good approximation of the original complete matrix if the data satisfies some assumptions.

RPCA assumes that we observe all entries of the full noisy matrix. Compared with standard PCA, RPCA utilizes the difference between the low-dimensional data structure and outliers. Also, different from two-stage methods, which first detect the outliers and remove them, RPCA describes the outliers as a sparse matrix and decompose the noisy

matrix into the sum of two or three matrices, with or without the Gaussian noise. In some applications, the outliers contain useful information. For example, in video surveillance, the low-rank part preserves the stationary background, whereas the sparse part can capture a moving object or person in the foreground. See Figure 1.3 for an example with four frames in a video.

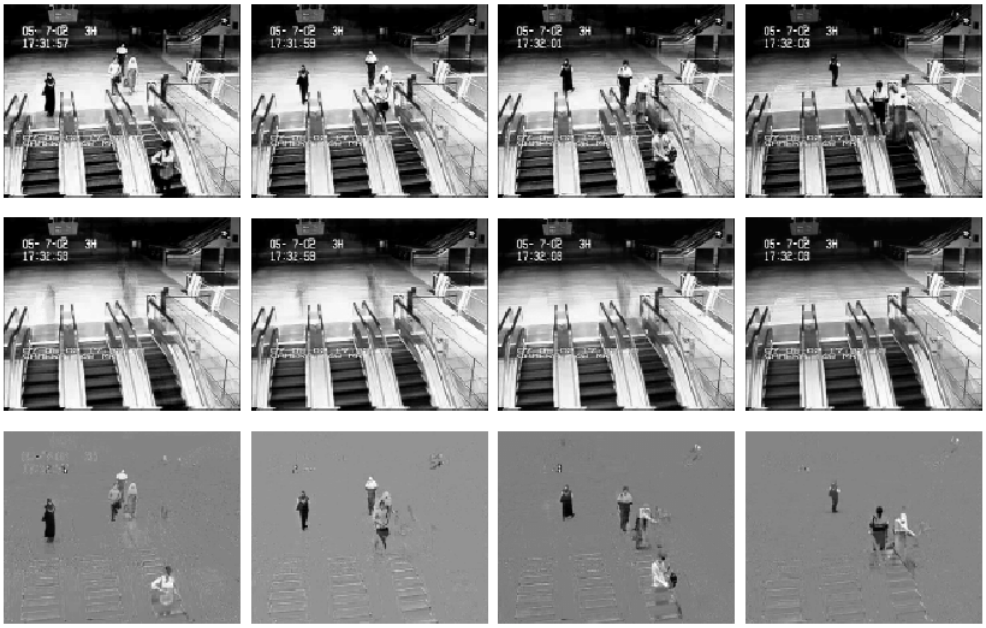


Figure 1.3: An application of RPCA. Image from (Zhou et al., 2014). Top row: four frames from a video. Middle row: video background, which is viewed as a low-rank approximation. Bottom row: main objects which correspond to sparse components.

RPCA separates a data matrix into the sum of a low-rank matrix, a sparse matrix (outliers), and a small noise matrix (Gaussian noise). If there is no Gaussian distortions, the small noise matrix is just a zero matrix. However, in practice, we are often confronted with such a situation where the collected data is incomplete. The good thing is that a low-rank matrix can be recovered from only a few entries of the matrix. For example, an $m \times n$ rank-one matrix can be recovered by one row and one column. When the given noisy matrix has missing entries, it is a low-rank matrix completion problem, that is, we will remove the outliers and recover the whole low-rank matrix. For example, in a film rating system, as shown in Figure 1.4, each user rates the films they watched. However, this data matrix tends

to be incomplete because people cannot watch all film showed in the rating website. Matrix completion techniques can predict the rates of each user for unwatched films and promote to the user the films they may like but did not watch.

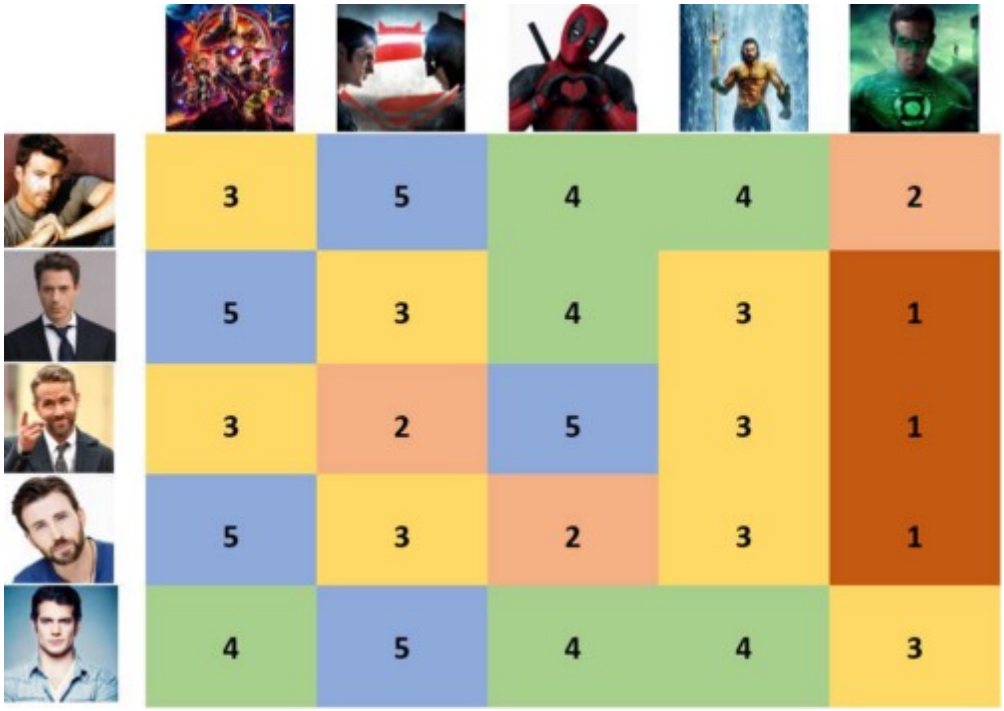


Figure 1.4: Image from https://link.medium.com/ERplrHLl5hb. Rating system for Netflix films. Each row represents each user while each column represents each film. Values from 1 to 5 are scores.

1.2 Existing work on RPCA

Assuming that the true data points lie on a low-dimensional subspace, we use a matrix $\mathbf{D} \in \mathbb{R}^{n \times p}$ to represent the observed noisy data, where n is the number of samples and p is the dimension of each sample. We also assume that the intrinsic dimension of the data is $k < p$. Therefore, in the linear case, k is also the rank of the matrix $\mathbf{L}_0$ consisting of the true samples. In this case, we have the following model,

$$\mathbf{D} = \mathbf{L}_0 + \mathbf{N}_0, \tag{1.1}$$

where $\mathbf{L}_0$ is a low-rank matrix with rank k and $\mathbf{N}_0$ is a small perturbation matrix (Gaussian noise). Then the classical PCA solves the following problem

$$\begin{aligned} &\underset{\mathbf{L}}{\text{minimize}} \ \|\mathbf{D}-\mathbf{L}\|_F^2 \\ &\text{subject to } \operatorname{rank}(\mathbf{L}) = k. \end{aligned}$$

The classical PCA assumes that the noise follows a Gaussian distribution. When only some components of the matrix are affected by large noise and the distribution of the noise is unknown, we consider RPCA, which assumes that

$$\mathbf{D} = \mathbf{L}_0 + \mathbf{S}_0,$$

with $\mathbf{S}_0$ being a sparse matrix. Because we do not have any knowledge about the distribution of the noise, the components can be considered as damaged and should be removed. If we know the locations, then we can remove them and fill-in new values for those locations. However, the locations are not given, and we have to find the sparse matrix $\mathbf{S}_0$ and the low-rank matrix $\mathbf{L}_0$ together using RPCA algorithms.

RPCA is an inverse problem to recover $\mathbf{L}$ and $\mathbf{S}$ from the matrix $\mathbf{D}$, which can be realized via solving the idealized nonconvex problem

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \operatorname{rank}(\mathbf{L}) + \lambda\|\mathbf{S}\|_0, \ \text{subject to } \mathbf{L}+\mathbf{S}=\mathbf{D}, \tag{1.2}$$

where λ is a parameter to balance the two objectives and $\|\mathbf{S}\|_0$ counts the number of non-zero entries in $\mathbf{S}$. However, this problem is NP-hard in general (Amaldi and Kann, 1998). Therefore, much attention is focused on the following convex relaxation:

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1, \ \text{subject to } \mathbf{L}+\mathbf{S}=\mathbf{D}. \tag{1.3}$$

Here $\|\cdot\|_*$ and $\|\cdot\|_1$ denote the nuclear norm and ℓ_1-norm of a matrix, respectively. This is called principal component pursuit (PCP) (Zhou et al., 2010a).

When Gaussian noise is also involved, we consider

$$\mathbf{D} = \mathbf{L} + \mathbf{S} + \mathbf{N}, \tag{1.4}$$

where $\mathbf{N}$ is the Gaussian noise. We can set the noise level to be ϵ and use the Frobenius norm to measure the Gaussian noise. A relaxed version of PCP is defined as

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1, \ \text{subject to} \ \|\mathbf{L}+\mathbf{S}-\mathbf{D}\|_F^2 \leq \epsilon. \tag{1.5}$$

This constrained optimization problem is also equivalent to the following unconstrained one

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \frac{1}{2\mu}\|\mathbf{L}+\mathbf{S}-\mathbf{D}\|_F^2 + \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1. \tag{1.6}$$

Classical optimization algorithms such as proximal gradient method (PGM), accelerated PGM, and alternating direction method of multipliers (ADMM) have been used to solve the unconstraned problem (1.6). Let's briefly go through these algorithms. For PGM, we can view the loss function as two parts corresponding with $\mathbf{L}$ and $\mathbf{S}$ respectively. Then the PGM takes two steps. The first step is the gradient descent, and the second step is to calculate the proximal functions. One iteration is described as

$$\begin{cases} \hat{\mathbf{L}}^k = \mathbf{L}^k - \dfrac{t}{\mu}(\mathbf{L}^k + \mathbf{S}^k - \mathbf{D}), \\ \hat{\mathbf{S}}^k = \mathbf{S}^k - \dfrac{t}{\mu}(\mathbf{L}^k + \mathbf{S}^k - \mathbf{D}), \\ \mathbf{L}^{k+1} = \arg\min\limits_{\mathbf{L}} \ t\|\mathbf{L}\|_* + \dfrac{1}{2}\|\mathbf{L} - \hat{\mathbf{L}}^k\|_F^2, \\ \mathbf{S}^{k+1} = \arg\min\limits_{\mathbf{S}} \ t\lambda\|\mathbf{S}\|_1 + \dfrac{1}{2}\|\mathbf{S} - \hat{\mathbf{S}}^k\|_F^2. \end{cases} \tag{1.7}$$

To solve the second proximal function, we can directly use soft-thresholding on each entry of the matrix. To solve the first proximal function, we need to use SVD to calculate the singular values and do the soft-thresholding on the singular values. We can set the initial condition $\mathbf{L} = \mathbf{0}$ and $\mathbf{S} = \mathbf{0}$.

As for accelerated proximal gradient method, we use FISTA (Beck and Teboulle, 2009b) as one example. The main change compared with the original PGM is that the shrinkage operator is not used on the previous point $\hat{\mathbf{L}}^k$ but on the linear combination of the previous two points $\hat{\mathbf{L}}^k$, $\hat{\mathbf{L}}^{k-1}$ as follows:

$$\bar{\mathbf{L}}^k = \mathbf{L}^k + \frac{\theta_{k-1}-1}{\theta_k}(\mathbf{L}^k - \mathbf{L}^{k-1}) \tag{1.8}$$

where $\theta_{k+1} = \frac{1+\sqrt{1+4\theta_k^2}}{2}$. Generally, it updates variables starting with $\theta_{-1} = \theta_0 = 1$. It has been proved that FISTA has improved the convergence rate from $O(1/k)$ to $O(1/k^2)$ for general convex problems.

When it goes to ADMM, we need to convert the original problem to a constrained form

$$\underset{\mathbf{Z},\mathbf{L},\mathbf{S}}{\text{minimize}} \ \frac{1}{2\mu}\|\mathbf{Z}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \|\mathbf{L}\|_*, \ \text{subject to } \mathbf{Z}+\mathbf{L}+\mathbf{S} = \mathbf{D}. \tag{1.9}$$

Unfortunately, its convergence for general three blocks has not been proved to converge. But in practice, it usually works very well.

In order to solve (1.9), we need to establish the augmented Lagrangian function

$$L_\beta(\mathbf{Z},\mathbf{S},\mathbf{L},\mathbf{Q}) = \frac{1}{2\mu}\|\mathbf{Z}\|_F^2+\lambda\|\mathbf{S}\|_1+\|\mathbf{L}\|_*-\beta\langle\mathbf{Q},\mathbf{Z}+\mathbf{S}+\mathbf{L}-\mathbf{D}\rangle+\frac{\beta}{2}\|\mathbf{Z}+\mathbf{S}+\mathbf{L}-\mathbf{D}\|_F^2. \tag{1.10}$$

Alternatively, we consider

$$L_\beta^{'}(\mathbf{Z},\mathbf{S},\mathbf{L},\mathbf{Q}) = \frac{1}{2\mu}\|\mathbf{Z}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \|\mathbf{L}\|_* + \frac{\beta}{2}\|\mathbf{Z}+\mathbf{S}+\mathbf{L}-\mathbf{D}-\mathbf{Q}\|_F^2, \tag{1.11}$$

with $L_\beta^{'}(\mathbf{Z},\mathbf{S},\mathbf{L},\mathbf{Q}) = L_\beta(\mathbf{Z},\mathbf{S},\mathbf{L},\mathbf{Q}) + \frac{\beta}{2}\|\mathbf{Q}\|_F^2$ if we want to get the optimal variables $\mathbf{Z}$, $\mathbf{S}$ and $\mathbf{L}$, respectively. Then we minimize $\mathbf{L}_\beta^{'}$ alternatingly.

$$\begin{cases} \mathbf{Z}^{k+1} := \arg\min_{\mathbf{Z}} \mathbf{L}_\beta^{'}(\mathbf{Z},\mathbf{S}^k,\mathbf{L}^k,\mathbf{Q}^k); \\ \mathbf{S}^{k+1} := \arg\min_{\mathbf{S}} \mathbf{L}_\beta^{'}(\mathbf{Z}^{k+1},\mathbf{S},\mathbf{L}^k,\mathbf{Q}^k); \\ \mathbf{L}^{k+1} := \arg\min_{\mathbf{L}} \mathbf{L}_\beta^{'}(\mathbf{Z}^{k+1},\mathbf{S}^{k+1},\mathbf{L}^k,\mathbf{Q}^k); \\ \mathbf{Q}^{k+1} := \mathbf{Q}^k - (\mathbf{Z}^{k+1}+\mathbf{S}^{k+1}+\mathbf{L}^{k+1}-\mathbf{D}). \end{cases} \tag{1.12}$$

More specifically,

$$\begin{cases} \mathbf{Z}^{k+1} := \arg\min_{\mathbf{Z}} \frac{1}{2\mu}\|\mathbf{Z}\|_F^2 + \frac{\beta}{2}\|\mathbf{Z}+\mathbf{S}^k+\mathbf{L}^k-\mathbf{D}-\mathbf{Q}^k\|_F^2; \\ \mathbf{S}^{k+1} := \arg\min_{\mathbf{S}} \lambda\|\mathbf{S}\|_1 + \frac{\beta}{2}\|\mathbf{Z}^{k+1}+\mathbf{S}+\mathbf{L}^k-\mathbf{D}-\mathbf{Q}^k\|_F^2; \\ \mathbf{L}^{k+1} := \arg\min_{\mathbf{S}} \|\mathbf{L}\|_* + \frac{\beta}{2}\|\mathbf{Z}^{k+1}+\mathbf{S}^{k+1}+\mathbf{L}-\mathbf{D}-\mathbf{Q}^k\|_F^2; \\ \mathbf{Q}^{k+1} := \mathbf{Q}^k - (\mathbf{Z}^{k+1}+\mathbf{S}^{k+1}+\mathbf{L}^{k+1}-\mathbf{D}). \end{cases} \tag{1.13}$$

All these approaches need to find the proximal of the nuclear norm, which requires SVD. When the matrix size is large, the SVD computation is very expensive and dominates other computation (Trefethen and Bau III, 1997).

1.3 Overview of this thesis

We briefly introduce the following chapters in this thesis. In Chapter 2, we study the theoretical parts of low-rank approximation. There are mainly two types of algorithms for RPCA. The first type of algorithm applies regularization terms on the singular values of a matrix to obtain the low-rank matrix. However, calculating singular values can be very expensive for large matrices. The second type of algorithm replaces the low-rank matrix as the multiplication of two smaller matrices. They are faster than the first type because no SVD is required. However, the rank of the low-rank matrix is required, and an accurate rank estimation is required to obtain a reasonable solution. In this chapter, we propose algorithms that combine both types. Our proposed algorithms require an upper bound of the rank and SVD on small matrices. First, they are faster than the first type because the cost of SVD on small matrices is negligible. Second, they are more robust than the second type because an upper bound of the rank instead of the rank is required. Numerical experiments show the good performance of our proposed algorithms.

In Chapter 3, we take examples from seismic data. Seismic events are usually buried in noise. Our goal is to discover the underlying signal from noise. RPCA based seismic denoising approaches yield promising results in separating useful seismic events from noise. However, current RPCA-based methods suffer from expensive computational costs, which hinders their wide applications in seismic data denoising and preprocessing. In this work, we develop a cost-effective denoising technique based on RPCA. Instead of solving for the clean data and noise simultaneously, we alternatively update them. This approach admits a large stepsize and increases the speed. In addition, we improve the model by incorporating a nonconvex term. To verify the effectiveness of our technique, we applied our denoising

technique to both synthetic and field reflection seismic data. From the numerical results, we observe that our denoising methods not only produce comparable or better denoising results but also yield efficient computational cost. Through comparison to other RPCA-based denoising methods, our method is at least 4-10x faster.

In Chapter 4, we extend RPCA to nonlinear manifolds. Suppose that the data matrix contains a sparse component and a component drawn from some low-dimensional manifold. Is it possible to separate both components by using the low dimensionality assumption of the manifold? Is there a benefit to treat the manifold as a whole as opposed to treating each local region individually? We answer these two questions affirmatively by proposing an optimization framework that separates these two components from noisy data. The efficacy of the proposed method is demonstrated on both synthetic and real dataset.

The three chapters consider offline algorithms, in which we have access to all data before the algorithm is applied. In the last chapter, we applied low-rank approximation on online matrix completion. Online optimization is more and more useful nowadays in data science and machine learning areas. As we know, for offline setting, we need to get all observations first and use all information to train the model. On the other side, online algorithms aim to make decision sequentially based on sequentially sampled data. For example, in many applications, like film recommendation systems, the user will give one rating for a film each time after watching. In this case, we can only get one observation each time. There are many work that aims to develop algorithms which can give updated result after every new observed entry. In Chapter 5, we consider the completion of a quaternion matrix, whose entries are quaternion numbers. Each quaternion number has one real number and three imaginary number. Therefore, it can be applied to color image with or without depth. We develop and analyze an online matrix completion algorithm for quaternion matrices. We want to find a decomposition of the matrix such that the matrix is a product of two small matrices, In this algorithm, after we receive an entry from the matrix, we update one row and one one column of the two small matrices, respectively.

CHAPTER 2

ROBUST PRINCIPAL COMPONENT ANALYSIS FOR LOW RANK MATRIX APPROXIMATION

2.1 Introduction

Robust principal component analysis (RPCA) decomposes a data matrix into a low-rank part and a sparse part. It has applications in a wide range of areas, including computer vision (De la Torre and Black, 2001), image processing (Liu et al., 2012; Elhamifar and Vidal, 2013), dimensionality reduction (Cunningham and Ghahramani, 2015), and bioinformatics data analysis (Da Costa et al., 2009). More specifically, the RPCA model has achieved great success in video surveillance and face recognition (Candès et al., 2011; Bouwmans and Zahzah, 2014). For example, in video surveillance, the low-rank part preserves the stationary background, whereas the sparse part can capture a moving object or person in the foreground.

We first assume that the data matrix $\mathbf{D}$ is obtained by the sum of a low-rank matrix and a spare matrix. That is

$$\mathbf{D} = \mathbf{L} + \mathbf{S},$$

where $\mathbf{L}$ is a low-rank matrix and $\mathbf{S}$ is a sparse matrix, which has only a few nonzero components. RPCA is an inverse problem to recover $\mathbf{L}$ and $\mathbf{S}$ from the matrix $\mathbf{D}$, which can be realized via solving the idealized nonconvex problem

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}}\ \text{rank}(\mathbf{L}) + \lambda\|\mathbf{S}\|_0,\ \text{subject to}\ \mathbf{L}+\mathbf{S}=\mathbf{D}, \tag{2.1}$$

where λ is a parameter to balance the two objectives and $\|\mathbf{S}\|_0$ counts the number of non-zero entries in $\mathbf{S}$. However, this problem is NP-hard in general (Amaldi and Kann, 1998). Therefore, much attention is focused on the following convex relaxation

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}}\ \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1,\ \text{subject to}\ \mathbf{L}+\mathbf{S}=\mathbf{D}. \tag{2.2}$$

Here $\|\cdot\|_*$ and $\|\cdot\|_1$ denote the nuclear norm and ℓ_1-norm of a matrix, respectively. It is shown that under mild conditions, the convex model (2.2) can exactly recover the low-rank and sparse parts with high probabilities (Candès et al., 2011). When additional Gaussian noise is considered, we can set the noise level to be ϵ and use the Frobenius norm $\|\cdot\|_F$ to measure the reconstruction error. Then, the problem becomes

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1, \text{ subject to } \|\mathbf{L}+\mathbf{S}-\mathbf{D}\|_F^2 \leq \epsilon. \tag{2.3}$$

Then, this constrained optimization problem is equivalent to the unconstrained problem

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \frac{1}{2\mu}\|\mathbf{L}+\mathbf{S}-\mathbf{D}\|_F^2 + \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \tag{2.4}$$

with a trade-off parameter μ. There is a correspondence between the two parameters ϵ and μ in (2.3) and (2.4), but the explicit expression does not exist. In this chapter, we will focus on the unconstrained problem (2.4), and the technique introduced in this chapter can be applied to the convex models (2.2) and (2.3). Please see Section 2.4 for more details.

There are many existing approaches for solving (2.4) including the augmented Lagrange method (Lin et al., 2010; Bouwmans and Zahzah, 2014; Wright et al., 2009). Some examples are proximal gradient method for $(\mathbf{L},\mathbf{S})$, alternating minimization for $\mathbf{L}$ and $\mathbf{S}$ (Shen et al., 2019), proximal gradient method for $\mathbf{L}$ after $\mathbf{S}$ is eliminated (Sha et al., 2019), alternating direction method of multipliers (ADMM) (Yuan and Yang, 2009; Tao and Yuan, 2011). All these approaches need to find the proximal of the nuclear norm, which require the singular value decomposition (SVD). When the matrix size is large, the SVD computation is very expensive and dominates all the computation (Trefethen and Bau III, 1997).

Alternative approaches for RPCA use matrix decomposition (Wen et al., 2012) and do not require SVD. Assuming that the rank of $\mathbf{L}$ is known as p, we can decompose it as

$$\mathbf{L} = \mathbf{X}\mathbf{Y}^\top,$$

with $\mathbf{X} \in \mathbb{R}^{m\times p}$ and $\mathbf{Y} \in \mathbb{R}^{n\times p}$. Then the following nonconvex optimization problem

$$\underset{\mathbf{X},\mathbf{Y},\mathbf{S}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{X}\mathbf{Y}^\top+\mathbf{S}-\mathbf{D}\|_F^2 + \lambda\|\mathbf{S}\|_1, \tag{2.5}$$

is considered. There are infinite many optimal solutions for this problem, since for any invertable matrix $\mathbf{A} \in \mathbb{R}^{p\times p}$, $(\mathbf{X}, \mathbf{Y}, \mathbf{S})$ and $(\mathbf{X}\mathbf{A}^{-1}, \mathbf{Y}\mathbf{A}^{\top}, \mathbf{S})$ have the same function value. In fact, for any matrix $\mathbf{L}$ with rank no greater than p, we can find $\mathbf{L} = \mathbf{X}\mathbf{Y}^{\top}$ and $\mathbf{Y}^{\top}\mathbf{Y} = \mathbf{I}_{p\times p}$. Therefore, we can have an additional constraint $\mathbf{Y}^{\top}\mathbf{Y} = \mathbf{I}_{p\times p}$. The resulting problem still has infinite many optimal solutions, since for any orthogonal matrix $\mathbf{A} \in \mathbb{R}^{p\times p}$, $(\mathbf{X}, \mathbf{Y}, \mathbf{S})$ and $(\mathbf{X}\mathbf{A}, \mathbf{Y}\mathbf{A}, \mathbf{S})$ have the same function value. Though $(\mathbf{X}, \mathbf{Y})$ are not unique, the low-rank matrix $\mathbf{L}$ that we need is unique at probability one. This resulting problem was discussed in (Shen et al., 2019), and an efficient algorithm by alternatively minimizing $\mathbf{X}\mathbf{Y}^{\top}$ and $\mathbf{S}$ is provided. In this algorithm, a Gauss-Newton algorithm is applied to update $\mathbf{X}\mathbf{Y}^{\top}$ and increase the speed.

Though the matrix decomposition problem is fast to solve, it is nonconvex and requires an accurate estimation of the rank of $\mathbf{L}$. Fig. 2.2 in Section 2.3.1.2 demonstrates that a good estimation of the rank is critical. However, in most scenarios, we do not have the exact rank of $\mathbf{L}$, but we can have an upper bound of the true rank. Therefore, we can combine the matrix decomposition and the nuclear norm minimization to have the benefits of both problems. The problem we consider in this chapter is

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{L}+\mathbf{S}-\mathbf{D}\|_F^2 + \mu\|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1, \ \text{subject to } \text{rank}(\mathbf{L}) \leq p. \tag{2.6}$$

When $\mu = 0$, the problem (2.6) is equivalent to (2.5). In addition, we consider the following more general problem

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \frac{1}{2}\|\mathcal{A}(\mathbf{L})+\mathbf{S}-\mathbf{D}\|_F^2 + \mu\|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1, \ \text{subject to } \text{rank}(\mathbf{L}) \leq p. \tag{2.7}$$

where $\mathbf{D}$ is the measurement of $\mathcal{A}(\mathbf{L})$ contaminated with both Gaussian noise and a sparse component. Here $\mathcal{A}$ is a bounded linear operator that describes how the measurements are calculated. For example, in robust matrix completion, we let $\mathcal{A}$ be the restriction operator on the given components of the matrix $\mathbf{L}$.

Note that the alternating minimization algorithm in (Shen et al., 2019) can not be applied to this general problem because the subproblem for $\mathbf{L}$ can no longer be solved efficiently by

the Gauss-Newton method. We will show the equivalency of the alternating minimization algorithm in (Shen et al., 2019) as a proximal gradient method applied to a problem with $\mathbf{L}$ only. Then the subproblem of $\mathbf{L}$ in our general problem (2.7) can still be solved efficiently with the Gauss-Newton method. Please see more details in Section 2.2.

For simplicity, we use the nuclear norm and ℓ_1-norm for the low-rank and sparse matrices, respectively. The purpose of this chapter is to introduce a fast algorithm to solve a type of RPCA algorithm, while the comparison of different penalties is out of the scopes of this chapter. The contributions of this chapter are:

- We propose a new model for RPCA, which combines the nuclear norm minimization and the matrix decomposition. The matrix decomposition brings efficient algorithms, and the nuclear norm miminization on a smaller matrix removes the requirement of the rank of the low-rank matrix. Note that the nuclear norm minimization can be replaced by other nonconvex penalties, and the results in this chapter are still valid.
- We develop efficient algorithms to solve this problem and show its convergence.

2.1.1 Notation

Throughout this chapter, matrices are denoted by bold capital letters (e.g., $\mathbf{A}$), and operators are denoted by calligraphic letters (e.g., $\mathcal{A}$). In particular, $\mathbf{I}$ denotes the identity matrix, $\mathbf{0}$ denotes the zero matrix (all entries equal to zero), and $\mathcal{I}$ denotes the identity operator. If there is potential for confusion, we indicate the dimension of the matrix with subscripts. For a matrix $\mathbf{A}$, $\mathbf{A}^\top$ represents its transpose and $\mathbf{A}(:, j:k)$ denotes the matrix composed by the columns of $\mathbf{A}$ indexing from j to k. Let $\mathbf{A}_{i,j}$ be the (i,j) entry of $\mathbf{A}$. The ℓ_1-norm of $\mathbf{A}$ is given by $\|\mathbf{A}\|_1 = \sum_{i,j} |\mathbf{A}_{i,j}|$. Denote the ith singular value of $\mathbf{A}$ by $\sigma_i(\mathbf{A})$. The nuclear norm of $\mathbf{A}$ is given by $\|\mathbf{A}\|_* = \sum_i \sigma_i(\mathbf{A})$. We will use $\partial\|\cdot\|_1$ and $\partial\|\cdot\|_*$ to denote the subgradients of the ℓ_1-norm and nuclear norm, respectively. The linear space of all $m \times n$ real matrices is denoted by $\mathbb{R}^{m\times n}$. For $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$, the inner product of $\mathbf{A}, \mathbf{B}$ is defined by

$\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{Tr}(\mathbf{A}^\top \mathbf{B})$, which induces the Frobenius norm $\|\mathbf{A}\|_F = \sqrt{\mathrm{Tr}(\mathbf{A}^\top \mathbf{A})} = \sqrt{\sum_i \sigma_i^2(\mathbf{A})}$. Let $\mathcal{A}$ be a linear bounded operator on $\mathbb{R}^{m\times n}$. The operator norm of $\mathcal{A}$ is given by $\|\mathcal{A}\| = \sup\{\|\mathcal{A}(\mathbf{A})\|_F : \mathbf{A} \in \mathbb{R}^{m\times n}, \|\mathbf{A}\|_F = 1\}$. The adjoint operator of $\mathcal{A}$ denoted by $\mathcal{A}^*$ is also linear and bounded on $\mathbb{R}^{m\times n}$ such that $\langle \mathcal{A}(\mathbf{A}), \mathbf{B} \rangle = \langle \mathbf{A}, \mathcal{A}^*(\mathbf{B}) \rangle$. The notation $\odot$ is used to denote the component-wise multiplication. Additionally, for a function $f : \mathbb{R} \to \mathbb{R}$, without further reference, f acting on a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ specifies that f is evaluated on each entry of $\mathbf{A}$, i.e., $f(\mathbf{A}) \in \mathbb{R}^{m\times n}$ with $f(\mathbf{A})_{i,j} = f(\mathbf{A}_{i,j})$. For example, if $f(x) = |x| - \lambda$, we can denote $f(\mathbf{A}) \in \mathbb{R}^{m\times n}$ by $|\mathbf{A}| - \lambda$ with $(|\mathbf{A}| - \lambda)_{i,j} = |\mathbf{A}_{i,j}| - \lambda$.

2.1.2 Organization

The rest of the chapter is organized as follows. We introduce our proposed algorithms and show their convergence in Section 2.2. Then we conduct numerical experiments to compare the performance of our proposed algorithms with existing approaches in Section 2.3. In Section 2.4, we introduce some potential extension from this chapter. We end this chapter with a short conclusion.

2.2 Proposed algorithms

The problem (2.6) is nonconvex because of the constraint $\mathrm{rank}(\mathbf{L}) \leq p$. It has several equivalent formulations. E.g., it is equivalent to the following nonconvex weighted nuclear norm minimization problem:

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{L}+\mathbf{S}-\mathbf{D}\|_F^2 + \mu \sum_{i=1}^{p} \sigma_i(\mathbf{L}) + C \sum_{i=p+1}^{\min(m,n)} \sigma_i(\mathbf{L}) + \lambda \|\mathbf{S}\|_1,$$

where C is a sufficiently large number such that the optimal $\mathbf{L}$ has at most p nonzero singular values. However, this formulation also requires the singular value decomposition of an $m \times n$ matrix in each iteration, which is expensive when m and n are large. We consider another equivalent problem with matrix decomposition in the following theorem.

Theorem 2.2.1. *Problem* (2.6) *is equivalent to*

$$\underset{\mathbf{X},\mathbf{Y},\mathbf{S}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{X}\mathbf{Y}^\top + \mathbf{S} - \mathbf{D}\|_F^2 + \mu\|\mathbf{X}\|_* + \lambda\|\mathbf{S}\|_1, \ \text{subject to } \mathbf{Y}^\top\mathbf{Y} = \mathbf{I}_{p\times p}. \tag{2.8}$$

More specifically, if $(\mathbf{X}, \mathbf{Y}, \mathbf{S})$ *is an optimal solution to* (2.8)*, then* $(\mathbf{X}\mathbf{Y}^\top, \mathbf{S})$ *is an optimal solution to* (2.6)*. If* $(\mathbf{L}, \mathbf{S})$ *is an optimal solution to* (2.6) *and we have the decomposition* $\mathbf{L} = \mathbf{X}\mathbf{Y}^\top$ *with* $\mathbf{Y}^\top\mathbf{Y} = \mathbf{I}_{p\times p}$*, then* $(\mathbf{X}, \mathbf{Y}, \mathbf{S})$ *is an optimal solution to* (2.8)*.*

Proof. For any matrix $\mathbf{L} \in \mathbb{R}^{m\times n}$ with rank no greater than p, we can have the decomposition

$$\mathbf{L} = \mathbf{X}\mathbf{Y}^\top,$$

with $\mathbf{Y}^\top\mathbf{Y} = \mathbf{I}_{p\times p}$. This decomposition is not unique, and one decomposition can be easily obtained from the compact SVD of $\mathbf{L}$. Let $\mathbf{L} = \mathbf{U}_p\Sigma_p\mathbf{V}_p^\top$ be the SVD of $\mathbf{L}$ with a square $p \times p$ matrix Σ_p, we have $\mathbf{V}_p^\top\mathbf{V}_p = \mathbf{I}_{p\times p}$. Thus, problem (2.6) is equivalent to

$$\underset{\mathbf{X},\mathbf{Y},\mathbf{S}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{X}\mathbf{Y}^\top + \mathbf{S} - \mathbf{D}\|_F^2 + \mu\|\mathbf{X}\mathbf{Y}^\top\|_* + \lambda\|\mathbf{S}\|_1, \ \text{subject to } \mathbf{Y}^\top\mathbf{Y} = \mathbf{I}_{p\times p}.$$

For any $\mathbf{X} \in \mathbb{R}^{m\times p}$, let $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$ be its SVD with $\mathbf{U} \in \mathbb{R}^{m\times p}$ and $\mathbf{V} \in \mathbb{R}^{p\times p}$. We have

$$\mathbf{X}\mathbf{Y}^\top = \mathbf{U}\Sigma\mathbf{V}^\top\mathbf{Y}^\top = \mathbf{U}\Sigma(\mathbf{Y}\mathbf{V})^\top.$$

Since $(\mathbf{Y}\mathbf{V})^\top(\mathbf{Y}\mathbf{V}) = \mathbf{V}^\top\mathbf{Y}^\top\mathbf{Y}\mathbf{V} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}_{p\times p}$. The SVD of $\mathbf{X}\mathbf{Y}^\top$ is $\mathbf{U}\Sigma(\mathbf{Y}\mathbf{V})^\top$, and $\|\mathbf{X}\mathbf{Y}^\top\|_* = \sum_{i=1}^p \Sigma_{ii} = \|\mathbf{X}\|_*$. Thus, problem (2.6) is equivalent to (2.8). □

Next, we consider problem (2.8) with $\mathbf{S}$ fixed. When $\mathbf{S}$ is fixed, it becomes a problem of $\mathbf{L} = \mathbf{X}\mathbf{Y}^\top$, and solving this problem is to find the proximal operator of the corresponding nonconvex weighted nuclear norm, which is denoted as

$$\underset{\mathbf{L}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{L} - \mathbf{M}\|_F^2 + \mu\|\mathbf{L}\|_*, \ \text{subject to } \text{rank}(\mathbf{L}) \le p, \tag{2.9}$$

or equivalently

$$\underset{\mathbf{X},\mathbf{Y}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{X}\mathbf{Y}^\top - \mathbf{M}\|_F^2 + \mu\|\mathbf{X}\|_*, \ \text{subject to } \mathbf{Y}^\top\mathbf{Y} = \mathbf{I}_{p\times p}, \tag{2.10}$$

where $\mathbf{M} = \mathbf{D} - \mathbf{S}$.

Theorem 2.2.2. *Let $q = \min(m, n)$. Problem (2.9) can be solved in two steps:*

1. *Find the compact SVD of $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\top$, with $\Sigma = \mathrm{diag}(\sigma_1(\mathbf{M}), \cdots, \sigma_q(\mathbf{M}))$ satisfying $\sigma_1(\mathbf{M}) \geq \sigma_2(\mathbf{M}) \geq \cdots \geq \sigma_q(\mathbf{M})$;*

2. *Construct a diagonal matrix $\hat{\Sigma}_\mu \in \mathbb{R}^{p\times p}$ with $(\hat{\Sigma}_\mu)_{ii} = \max(\Sigma_{ii} - \mu, 0)$, then one solution of (2.9) is $\mathbf{U}(:, 1:p)\hat{\Sigma}_\mu\mathbf{V}(:, 1:p)^\top$.*

In addition, for any orthogonal matrix $\mathbf{A} \in \mathbb{R}^{p\times p}$, $(\mathbf{U}(:, 1:p)\hat{\Sigma}_\mu\mathbf{A}, \mathbf{V}(:, 1:p)\mathbf{A})$ is an optimal solution of (2.10).

Proof. Given any $\mathbf{L} \in \mathbb{R}^{m\times n}$ with $\mathrm{rank}(\mathbf{L}) \leq p$, let $\sigma_1, \sigma_2, \cdots, \sigma_q$ be its singular values in the decreasing order such that $\sigma_{p+1} = \cdots = \sigma_q = 0$. Note that the main diagonal entries of Σ are the singular values of $\mathbf{M}$. According to the von-Neumann trace inequality (Horn and Johnson, 2012, Theorem 7.4.1.1), one can bound the matrix inner product by the singular values, i.e., $\langle \mathbf{L}, \mathbf{M}\rangle \leq \sum_{i=1}^q \sigma_i\Sigma_{ii}$. Then we have

$$\begin{aligned}
\frac{1}{2}\|\mathbf{L} - \mathbf{M}\|_F^2 + \mu\|\mathbf{L}\|_* =& \frac{1}{2}\|\mathbf{L}\|_F^2 + \frac{1}{2}\|\mathbf{M}\|_F^2 - \langle \mathbf{L}, \mathbf{M}\rangle + \mu\|\mathbf{L}\|_* \\
\geq& \frac{1}{2}\sum_{i=1}^q \sigma_i^2 + \frac{1}{2}\sum_{i=1}^q \Sigma_{ii}^2 - \sum_{i=1}^q \sigma_i\Sigma_{ii} + \mu\sum_{i=1}^q \sigma_i \\
=& \frac{1}{2}\sum_{i=1}^p \sigma_i^2 + \frac{1}{2}\sum_{i=1}^q \Sigma_{ii}^2 - \sum_{i=1}^p \sigma_i\Sigma_{ii} + \mu\sum_{i=1}^p \sigma_i,
\end{aligned} \tag{2.11}$$

where the equality is satisfied when $\mathbf{L}$ has a simultaneous SVD with $\mathbf{M}$ through $\mathbf{U}$ and $\mathbf{V}$. Therefore, the optimal $\mathbf{L}$ minimizing $\frac{1}{2}\|\mathbf{L} - \mathbf{M}\|_F^2 + \mu\|\mathbf{L}\|_*$ can be selected from the matrices that have a simultaneous SVD with $\mathbf{M}$ through $\mathbf{U}$ and $\mathbf{V}$. Then we can assume that the optimal $\mathbf{L}$ satisfies

$$\mathbf{L} = \mathbf{U}\mathrm{diag}(\sigma_1, \cdots, \sigma_p, \sigma_{p+1}, \cdots, \sigma_q)\mathbf{V}^\top = \mathbf{U}(:, 1:p)\mathrm{diag}(\sigma_1, \cdots, \sigma_p)\mathbf{V}(:, 1:p)^\top,$$

where the last equality holds because of the fact that $\sigma_{p+1} = \cdots = \sigma_q = 0$. Next, one can construct an optimal $\mathbf{L}$ of the above form by letting $\sigma_i = \max(\Sigma_{ii} - \mu, 0)$ for $i = 1, 2, \cdots, p$, which minimizes the last equation in (2.11). Thus $\mathbf{U}(:, 1:p)\hat{\Sigma}_\mu\mathbf{V}(:, 1:p)^\top$ minimizes the objective function of (2.9) over all $\mathbf{L} \in \mathbb{R}^{m\times n}$ with rank no greater than p.

By the same argument in the proof of Theorem 2.2.1, we see that problem (2.10) is equivalent to problem (2.9). Since for any orthogonal matrix $\mathbf{A} \in \mathbb{R}^{p\times p}$, there hold

$$\mathbf{L} = (\mathbf{U}(:, 1:p)\hat{\Sigma}_\mu \mathbf{A})(\mathbf{V}(:, 1:p)\mathbf{A})^\top$$

and

$$(\mathbf{V}(:, 1:p)\mathbf{A})^\top(\mathbf{V}(:, 1:p)\mathbf{A}) = \mathbf{A}^\top\mathbf{A} = \mathbf{I}_{p\times p}.$$

Therefore, $(\mathbf{U}(:, 1:p)\hat{\Sigma}_\mu\mathbf{A}, \mathbf{V}(:, 1:p)\mathbf{A})$ is an optimal solution of problem (2.10). □

The first step to solve problem (2.10) in the previous theorem requires the truncated SVD of an $m \times n$ matrix $\mathbf{M}$. Since we only need the first p ($p < q = \min(m, n)$) singular values, we use the Gauss-Newton algorithm to find $(\mathbf{X}, \mathbf{Y})$ alternatively. In this approach, we require the SVD of a $m \times p$ matrix, which is much faster than the truncated SVD of a $m \times n$ matrix when p is small. In addition, we use the previous $\mathbf{X}$ as the initial guess in the next iteration to reduce the number of inner iterations for the Gauss-Newton algorithm.

Lemma 2.2.3. *If the rank of $\mathbf{M} \in \mathbb{R}^{m\times n}$ is larger than p, problem* (2.10) *can be solved in the following three steps:*

1. *Find $\hat{\mathbf{X}} \in \mathbb{R}^{m\times p}$ ($p < m$) by solving the following optimization problem*

$$\underset{\mathbf{X}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{X}\mathbf{X}^\top - \mathbf{M}\mathbf{M}^\top\|_F^2;$$

2. $\mathbf{Y} = \mathbf{M}^\top\hat{\mathbf{X}}(\hat{\mathbf{X}}^\top\hat{\mathbf{X}})^{-1}$;

3. *Let $\hat{\mathbf{X}} = \mathbf{U}_p\hat{\Sigma}\mathbf{A}$ be its thin SVD with $\hat{\Sigma} \in \mathbb{R}^{p\times p}$ and choose $\mathbf{X}$ as $\mathbf{X} = \mathbf{U}_p\hat{\Sigma}_\mu\mathbf{A}$ with $(\hat{\Sigma}_\mu)_{ii} = \max(0, \hat{\Sigma}_{ii} - \mu)$ for $i = 1, \ldots, p$. Then $(\mathbf{X}, \mathbf{Y})$ is a solution of problem* (2.10).

Proof. Given any $\mathbf{X} \in \mathbb{R}^{m\times p}$, let $\lambda_1, \lambda_2, \cdots, \lambda_m$ be the non-negative eigenvalues of the matrix $\mathbf{X}\mathbf{X}^\top$. Since $\text{rank}(\mathbf{X}) \le p < m$, we have $\lambda_{p+1} = \cdots = \lambda_m = 0$. Recall that the compact SVD of $\mathbf{M}$ given in Theorem 2.2.2 is $\mathbf{U}\Sigma\mathbf{V}^\top$ with $\Sigma \in \mathbb{R}^{q\times q}$ (here $q = \min(m, n)$). Then

$\Sigma_{11}^2 \geq \Sigma_{11}^2 \geq \cdots \geq \Sigma_{qq}^2$ are the largest q eigenvalues of the matrix $\mathbf{M}\mathbf{M}^\top$, and if $q < m$, the remaining eigenvalues of $\mathbf{M}\mathbf{M}^\top$ are all zeros. Then we have

$$\begin{aligned}\|\mathbf{X}\mathbf{X}^\top - \mathbf{M}\mathbf{M}^\top\|_F^2 &\geq \sum_{i=1}^{p} \lambda_i^2 + \sum_{i=1}^{q} \Sigma_{ii}^4 - 2\sum_{i=1}^{p} \lambda_i \Sigma_{ii}^2 \\ &= \sum_{i=1}^{p} (\lambda_i - \Sigma_{ii}^2)^2 + \sum_{i=p+1}^{q} \Sigma_{ii}^4 \geq \sum_{i=p+1}^{q} \Sigma_{ii}^4,\end{aligned}$$

where the equality is satisfied when we choose $\mathbf{X} = \mathbf{U}(:, 1:p)\mathrm{diag}(\Sigma_{11}, \cdots, \Sigma_{pp})$. Let $\hat{\Sigma} = \mathrm{diag}(\Sigma_{11}, \cdots, \Sigma_{pp})$. The matrix $\hat{\Sigma}$ is invertible as the rank of $\mathbf{M}$ is larger than p. Then for any orthogonal matrix $\mathbf{A} \in \mathbb{R}^{p\times p}$, $\hat{\mathbf{X}} = \mathbf{U}(:, 1:p)\hat{\Sigma}\mathbf{A}$ minimizes the objective function $\frac{1}{2}\|\mathbf{X}\mathbf{X}^\top - \mathbf{M}\mathbf{M}^\top\|_F^2$.

After we find $\hat{\mathbf{X}} = \mathbf{U}(:, 1:p)\hat{\Sigma}\mathbf{A}$ for a certain orthogonal matrix $\mathbf{A}$, we have

$$\begin{aligned}\mathbf{Y} = \mathbf{M}^\top\hat{\mathbf{X}}(\hat{\mathbf{X}}^\top\hat{\mathbf{X}})^{-1} =& \mathbf{V}\Sigma\mathbf{U}^\top\mathbf{U}(:, 1:p)\hat{\Sigma}\mathbf{A}((\mathbf{U}(:, 1:p)\hat{\Sigma}\mathbf{A})^\top\mathbf{U}(:, 1:p)\hat{\Sigma}\mathbf{A})^{-1} \\ =& \mathbf{V}\Sigma\mathbf{U}^\top\mathbf{U}(:, 1:p)\hat{\Sigma}^{-1}\mathbf{A} \\ =& \mathbf{V}(:, 1:p)\hat{\Sigma}\hat{\Sigma}^{-1}\mathbf{A} = \mathbf{V}(:, 1:p)\mathbf{A},\end{aligned}$$

where the third equality is due to the fact that

$$\Sigma\mathbf{U}^\top\mathbf{U}(:, 1:p) = \begin{bmatrix} \hat{\Sigma}_{p\times p} \\ \mathbf{0}_{(q-p)\times p} \end{bmatrix}.$$

According to Theorem 2.2.2, $(\hat{\mathbf{X}}, \mathbf{Y})$ is an optimal solution of problem (2.10) if $\mu = 0$. Note that $\hat{\mathbf{X}} = \mathbf{U}(:, 1:p)\hat{\Sigma}\mathbf{A}$ is the thin SVD with $\hat{\Sigma} \in \mathbb{R}^{p\times p}$. Then, the third step gives $\mathbf{X} = \mathbf{U}(:, 1:p)\hat{\Sigma}_\mu\mathbf{A}$. Theorem 2.2.2 shows that $(\mathbf{X}, \mathbf{Y})$ is an optimal solution of problem (2.10). □

Remark: To find $\hat{\mathbf{X}}$ in the first step, we apply the Gauss-Newton algorithm from (Liu et al., 2015), which is previously used for RPCA in (Shen et al., 2019). The iteration is $\mathbf{X} \leftarrow \mathbf{M}\mathbf{M}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} - \mathbf{X}((\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{M}\mathbf{M}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} - \mathbf{I})/2$. When p is small, computing the inverse of $\mathbf{X}^\top\mathbf{X}$ is fast. Though an iterative algorithm is required to solve this subproblem at each outer iteration, we can use the output from the previous outer iteration

as the initial and the number of inner iterations is reduced significantly. Therefore, the computational time can be reduced significantly, as shown in Section 2.3. In the numerical experiments, the first Gauss-Newton algorithm requires several hundred iterations, while the number for following Gauss-Newton algorithms reduces to less than ten.

From Theorem 2.2.2, we say that we solve the proximal operator of the nonconvex function $\|\mathbf{L}\|_* + \iota_{\mathrm{rank}(\mathbf{L})\leq p}(\mathbf{L})$ exactly. Here the indicator function is defined as

$$\iota_{\mathrm{rank}(\mathbf{L})\leq p}(\mathbf{L}) = \begin{cases} 0, & \text{if } \mathrm{rank}(\mathbf{L}) \leq p; \\ +\infty, & \text{otherwise.} \end{cases}$$

With these theorems, we are ready to develop optimization algorithms for the general problem (2.7).

2.2.1 Forward-backward

First, we eliminate $\mathbf{S}$, and it becomes the following problem with $\mathbf{L}$ only:

$$\begin{aligned} &\underset{\mathbf{L}:\mathrm{rank}(\mathbf{L})\leq p}{\text{minimize}} \min_{\mathbf{S}} \frac{1}{2}\|\mathcal{A}(\mathbf{L}) + \mathbf{S} - \mathbf{D}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \mu\|\mathbf{L}\|_* \\ =& \underset{\mathbf{L}:\mathrm{rank}(\mathbf{L})\leq p}{\text{minimize}} \min_{\mathbf{S}} \left\{\frac{1}{2}\|\mathcal{A}(\mathbf{L}) + \mathbf{S} - \mathbf{D}\|_F^2 + \lambda\|\mathbf{S}\|_1\right\} + \mu\|\mathbf{L}\|_* \\ =& \underset{\mathbf{L}:\mathrm{rank}(\mathbf{L})\leq p}{\text{minimize}} f_\lambda(\mathbf{D} - \mathcal{A}(\mathbf{L})) + \mu\|\mathbf{L}\|_*. \end{aligned} \tag{2.12}$$

Here f_λ is the Moreau envelope of $\lambda|\cdot|$ defined by $f_\lambda(x) = \min_{y\in\mathbb{R}}\{\lambda|y| + \frac{1}{2}(y-x)^2\}$. So it is differential and has a 1-Lipschitz continuous gradient. Then we can apply the proximal-gradient method (or forward-backward operator splitting). We take the gradient of f_λ, which is given by

$$f_\lambda'(x) = x - \mathrm{sign}(x)\max(0, |x| - \lambda) = \mathrm{sign}(x)\min(\lambda, |x|). \tag{2.13}$$

The forward-backward iteration for $\mathbf{L}$ with stepsize t is

$$\mathbf{L}^{k+1} = \mathbf{prox}_{t\mu}\left(\mathbf{L}^k - t\mathcal{A}^* f_\lambda'(\mathcal{A}(\mathbf{L}^k) - \mathbf{D})\right), \tag{2.14}$$

where the proximal operator is defined by

$$\mathbf{prox}_\mu(\mathbf{A}) = \underset{\mathbf{L}:\mathrm{rank}(\mathbf{L})\leq p}{\arg\min} \frac{1}{2}\|\mathbf{L} - \mathbf{A}\|_F^2 + \mu\|\mathbf{L}\|_*. \tag{2.15}$$

The algorithm is summarized in Alg. 2.1.

Algorithm 2.1: RPCA for low rank matrix approximation

Input: $\mathbf{D}$, μ, λ, p, $\mathcal{A}$, stepsize t, stopping criteria ϵ, maximum number of iterations Max_Iter, initialization $\mathbf{L}^0 = \mathbf{0}$

Output: $\mathbf{L}$, $\mathbf{S}$

for *k = 0, 1, 2, 3, . . . , Max_ Iter* **do**
  $\mathbf{S} = \text{sign}(\mathbf{D} - \mathcal{A}(\mathbf{L}^k)) \odot \max(0, |\mathbf{D} - \mathcal{A}(\mathbf{L}^k)| - \lambda)$;
  $\mathbf{L}^{k+1} = \mathbf{prox}_{t\mu}(\mathbf{L}^k - t\mathcal{A}^*(\mathcal{A}(\mathbf{L}^k) - \mathbf{D} + \mathbf{S})$ using Gauss-Newton;
  if $\|\mathbf{L}^{k+1} - \mathbf{L}^k\|_F / \|\mathbf{L}^k\|_F < \epsilon$ **then**
    break
  end
end

Connection to (Shen et al., 2019). Consider the special case with $\mathcal{A} = \mathcal{I}$ and $\mu = 0$. We let $t = 1$ in (2.14) and obtain the following iteration

$$\mathbf{L}^{k+1} = \mathbf{prox}_0(\mathbf{L}^k - f'_\lambda(\mathbf{L}^k - \mathbf{D})) = \underset{\mathbf{L}:\text{rank}(\mathbf{L})\le p}{\arg\min} \frac{1}{2}\|\mathbf{L} + \mathbf{S}^{k+1} - \mathbf{D}\|^2,$$

where $\mathbf{S}^{k+1} = \text{sign}(\mathbf{D} - \mathbf{L}^k) \odot \max(0, |\mathbf{D} - \mathbf{L}^k| - \lambda)$. This is exactly the algorithm in (Shen et al., 2019) for solving (2.5). It alternates between finding the best $\mathbf{S}$ with $\mathbf{L}$ fixed and the best $\mathbf{L}$ (or $(\mathbf{X}, \mathbf{Y})$) with $\mathbf{S}$ fixed.

Recently, the work (Cai et al., 2019) proposed a novel RPCA algorithm with linear convergence. It projects matrices to special manifolds of low-rank matrices, and their truncated SVD can be computed efficiently. Our matrix does not have this property in our algorithm, and a good initial guess from the previous iteration is necessary to reduce the computation in the Gauss-Newton method.

2.2.1.1 Convergence analysis

From the discussion above, problem (2.7) can be solved by an iteration process of forward-backward splitting. In each iteration, we reduce the value of the objective function

$$E(\mathbf{L}, \mathbf{S}) = \frac{1}{2}\|\mathcal{A}(\mathbf{L}) + \mathbf{S} - \mathbf{D}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \mu\|\mathbf{L}\|_* \tag{2.16}$$

by applying proximal operators to $\mathbf{L}$ and $\mathbf{S}$ alternatively. The resulting iteration sequence $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$ with some initial $(\mathbf{L}^0, \mathbf{S}^0)$ is explicitly given by

$$\begin{aligned} \mathbf{S}^k &= \operatorname{sign}(\mathbf{D} - \mathcal{A}(\mathbf{L}^{k-1})) \odot \max(0, |\mathbf{D} - \mathcal{A}(\mathbf{L}^{k-1})| - \lambda), \\ \mathbf{L}^k &= \mathbf{prox}_{t\mu}\left(\mathbf{L}^{k-1} - t\mathcal{A}^*(\mathcal{A}(\mathbf{L}^{k-1}) + \mathbf{S}^k - \mathbf{D})\right), \end{aligned} \tag{2.17}$$

where the proximal operator $\mathbf{prox}_{t\mu}(\cdot)$ for updating $\mathbf{L}$ is defined by (2.15). Here we use (2.13) to derive

$$\begin{aligned} &f'_\lambda(\mathcal{A}(\mathbf{L}^{k-1}) - \mathbf{D}) \\ &= \mathcal{A}(\mathbf{L}^{k-1}) - \mathbf{D} + \operatorname{sign}(\mathbf{D} - \mathcal{A}(\mathbf{L}^{k-1})) \odot \max(0, |\mathbf{D} - \mathcal{A}(\mathbf{L}^{k-1})| - \lambda) \\ &= \mathcal{A}(\mathbf{L}^{k-1}) + \mathbf{S}^k - \mathbf{D}. \end{aligned}$$

In this subsection, we establish the convergence results for $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$. We will show that every limit point of $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$, denoted by $(\mathbf{L}^\star, \mathbf{S}^\star)$, is a fixed point of the proximal operator, i.e.,

$$\begin{aligned} \mathbf{S}^\star &= \operatorname{sign}(\mathbf{D} - \mathcal{A}(\mathbf{L}^\star)) \odot \max(0, |\mathbf{D} - \mathcal{A}(\mathbf{L}^\star)| - \lambda), \\ \mathbf{L}^\star &= \mathbf{prox}_{t\mu}\left(\mathbf{L}^\star - t\mathcal{A}^*(\mathcal{A}(\mathbf{L}^\star) + \mathbf{S}^\star - \mathbf{D})\right). \end{aligned} \tag{2.18}$$

In practical execution, one can efficiently solve the proximal operator for $\mathbf{L}$ by solving $(\mathbf{X}^k, \mathbf{Y}^k)$ through

$$\begin{aligned} &\underset{\mathbf{X},\mathbf{Y}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{X}\mathbf{Y}^\top - \mathbf{L}^{k-1} + t\mathcal{A}^*(\mathcal{A}(\mathbf{L}^{k-1}) + \mathbf{S}^k - \mathbf{D})\|_F^2 + \mu\|\mathbf{X}\|_*, \\ &\text{subject to } \mathbf{Y}^\top\mathbf{Y} = \mathbf{I}_{p\times p}, \end{aligned} \tag{2.19}$$

and letting $\mathbf{L}^k = \mathbf{X}^k(\mathbf{Y}^k)^\top$. We will also prove that if $(\mathbf{X}^\star, \mathbf{Y}^\star, \mathbf{S}^\star)$ is a limit point of $\{(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{S}^k)\}_{k\geq 1}$, then $(\mathbf{X}^\star(\mathbf{Y}^\star)^\top, \mathbf{S}^\star)$ is a limit point of $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$, and the limit point $(\mathbf{X}^\star, \mathbf{Y}^\star, \mathbf{S}^\star)$ is a stationary point of

$$E(\mathbf{X}\mathbf{Y}^\top, \mathbf{S}) = \frac{1}{2}\|\mathcal{A}(\mathbf{X}\mathbf{Y}^\top) + \mathbf{S} - \mathbf{D}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \mu\|\mathbf{X}\mathbf{Y}^\top\|_*,$$

i.e., $(\mathbf{X}^\star, \mathbf{Y}^\star, \mathbf{S}^\star)$ satisfies the first-order optimality condition

$$\begin{aligned}
&\mathbf{0} \in [\mathcal{A}^*(\mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top) + \mathbf{S}^\star - \mathbf{D}) + \mu\partial\|\mathbf{X}^\star(\mathbf{Y}^\star)^\top\|_*]\mathbf{Y}^\star,\\
&\mathbf{0} \in (\mathbf{X}^\star)^\top[\mathcal{A}^*(\mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top) + \mathbf{S}^\star - \mathbf{D}) + \mu\partial\|\mathbf{X}^\star(\mathbf{Y}^\star)^\top\|_*],\\
&\mathbf{0} \in \mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top) + \mathbf{S}^\star - \mathbf{D} + \lambda\partial\|\mathbf{S}^\star\|_1.
\end{aligned} \tag{2.20}$$

We summarize these results in the following theorem.

Theorem 2.2.4. *Define the objective function $E(\mathbf{L}, \mathbf{S})$ as (2.16). Let $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$ be a sequence generated by (2.17) with initial $(\mathbf{L}^0, \mathbf{S}^0)$ and stepsize $t < \frac{1}{\|\mathcal{A}\|^2}$, where $\mathbf{L}^k = \mathbf{X}^k(\mathbf{Y}^k)^\top$ with $(\mathbf{X}^k, \mathbf{Y}^k)$ being solved from (2.19). We have the following statements:*

1. *The objective values $\{E(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$ are non-increasing along $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$.*
2. *The sequence $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$ is bounded and thus has limit points.*
3. *Every limit point $(\mathbf{L}^\star, \mathbf{S}^\star)$ of $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$ satisfies (2.18).*
4. *The sequence $\{(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{S}^k)\}_{k\geq 1}$ is also bounded. In addition, for any limit point $(\mathbf{X}^\star, \mathbf{Y}^\star, \mathbf{S}^\star)$ of $\{(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{S}^k)\}_{k\geq 1}$, $(\mathbf{X}^\star(\mathbf{Y}^\star)^\top, \mathbf{S}^\star)$ is a limit point of $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$.*
5. *Every limit point $(\mathbf{X}^\star, \mathbf{Y}^\star, \mathbf{S}^\star)$ of $\{(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{S}^k)\}_{k\geq 1}$ is a stationary point of $E(\mathbf{X}\mathbf{Y}^\top, \mathbf{S})$, which satisfies the first-order optimality condition in (2.20).*

In addition, if $\mathcal{A} = \mathcal{I}$, we can take the stepsize $t = 1$, and all the statements above still hold.

Proof. We start by verifying the first two statements. For $k \geq 0$ and $t < \frac{1}{\|\mathcal{A}\|^2}$, we have

$$\begin{aligned}
&E(\mathbf{L}^{k+1}, \mathbf{S}^{k+1}) \\
&= \frac{1}{2}\|\mathcal{A}(\mathbf{L}^{k+1}) - \mathcal{A}(\mathbf{L}^k)\|_F^2 + \langle \mathcal{A}(\mathbf{L}^{k+1}) - \mathcal{A}(\mathbf{L}^k), \mathcal{A}(\mathbf{L}^k) + \mathbf{S}^{k+1} - \mathbf{D}\rangle \\
&\quad + \frac{1}{2}\|\mathcal{A}(\mathbf{L}^k) + \mathbf{S}^{k+1} - \mathbf{D}\|_F^2 + \lambda\|\mathbf{S}^{k+1}\|_1 + \mu\|\mathbf{L}^{k+1}\|_* \\
&\leq \frac{1}{2t}\|\mathbf{L}^{k+1} - \mathbf{L}^k\|_F^2 + \langle \mathbf{L}^{k+1} - \mathbf{L}^k, \mathcal{A}^* f_\lambda'(\mathcal{A}(\mathbf{L}^k) - \mathbf{D})\rangle + \mu\|\mathbf{L}^{k+1}\|_* \\
&\quad + \frac{1}{2}\|\mathcal{A}(\mathbf{L}^k) + \mathbf{S}^{k+1} - \mathbf{D}\|_F^2 + \lambda\|\mathbf{S}^{k+1}\|_1 + \left(\frac{\|\mathcal{A}\|^2}{2} - \frac{1}{2t}\right)\|\mathbf{L}^{k+1} - \mathbf{L}^k\|_F^2 \\
&= \frac{1}{t}\left\{\frac{1}{2}\|\mathbf{L}^{k+1} - \mathbf{L}^k + t\mathcal{A}^* f_\lambda'(\mathcal{A}(\mathbf{L}^k) - \mathbf{D})\|_F^2 + t\mu\|\mathbf{L}^{k+1}\|_*\right\} \\
&\quad - \frac{t}{2}\|\mathcal{A}^* f_\lambda'(\mathcal{A}(\mathbf{L}^k) - \mathbf{D})\|_F^2 + \left(\frac{\|\mathcal{A}\|^2}{2} - \frac{1}{2t}\right)\|\mathbf{L}^{k+1} - \mathbf{L}^k\|_F^2 \\
&\quad + \frac{1}{2}\|\mathcal{A}(\mathbf{L}^k) + \mathbf{S}^{k+1} - \mathbf{D}\|_F^2 + \lambda\|\mathbf{S}^{k+1}\|_1,
\end{aligned} \tag{2.21}$$

where the inequality is due to the facts that

$$\|\mathcal{A}(\mathbf{L}^{k+1}) - \mathcal{A}(\mathbf{L}^k)\|_F^2 \leq \|\mathcal{A}\|^2\|\mathbf{L}^{k+1} - \mathbf{L}^k\|_F^2$$

and

$$\mathcal{A}(\mathbf{L}^k) + \mathbf{S}^{k+1} - \mathbf{D} = f_\lambda'(\mathcal{A}(\mathbf{L}^k) - \mathbf{D}).$$

Note that $\mathbf{L}^{k+1} = \mathbf{prox}_{t\mu}\left(\mathbf{L}^k - t\mathcal{A}^* f_\lambda'(\mathcal{A}(\mathbf{L}^k) - \mathbf{D})\right)$, which solves

$$\underset{\mathbf{L}:\mathrm{rank}(\mathbf{L})\leq p}{\text{minimize}} \ \frac{1}{2}\|\mathbf{L} - \mathbf{L}^k + t\mathcal{A}^* f_\lambda'(\mathcal{A}(\mathbf{L}^k) - \mathbf{D})\|_F^2 + t\mu\|\mathbf{L}\|_*.$$

Since $\mathrm{rank}(\mathbf{L}^k) \leq p$, we have

$$\begin{aligned}
&\frac{1}{2}\|\mathbf{L}^{k+1} - \mathbf{L}^k + t\mathcal{A}^* f_\lambda'(\mathcal{A}(\mathbf{L}^k) - \mathbf{D})\|_F^2 + t\mu\|\mathbf{L}^{k+1}\|_* \\
&\leq \frac{1}{2}\|\mathbf{L}^k - \mathbf{L}^k + t\mathcal{A}^* f_\lambda'(\mathcal{A}(\mathbf{L}^k) - \mathbf{D})\|_F^2 + t\mu\|\mathbf{L}^k\|_* \\
&= \frac{t^2}{2}\|\mathcal{A}^* f_\lambda'(\mathcal{A}(\mathbf{L}^k) - \mathbf{D})\|_F^2 + t\mu\|\mathbf{L}^k\|_*.
\end{aligned}$$

Substituting the above estimate to (2.21) yields

$$\begin{aligned}
E(\mathbf{L}^{k+1}, \mathbf{S}^{k+1}) &\leq \left(\frac{\|\mathcal{A}\|^2}{2} - \frac{1}{2t}\right)\|\mathbf{L}^{k+1} - \mathbf{L}^k\|_F^2 \\
&\quad + \frac{1}{2}\|\mathcal{A}(\mathbf{L}^k) + \mathbf{S}^{k+1} - \mathbf{D}\|_F^2 + \mu\|\mathbf{L}^k\|_* + \lambda\|\mathbf{S}^{k+1}\|_1.
\end{aligned} \tag{2.22}$$

Moreover, we see that

$$\mathbf{S}^{k+1} = \arg\min_{\mathbf{S}} \ \frac{1}{2}\|\mathbf{S} - (\mathbf{D} - \mathcal{A}(\mathbf{L}^k))\|_F^2 + \lambda\|\mathbf{S}\|_1.$$

Then from (Lou and Yan, 2018, Lemma 2), there holds

$$\begin{aligned} &\frac{1}{2}\|\mathbf{S}^{k+1} - (\mathbf{D} - \mathcal{A}(\mathbf{L}^k))\|_F^2 + \lambda\|\mathbf{S}^{k+1}\|_1 \\ &\leq \frac{1}{2}\|\mathbf{S}^{k} - (\mathbf{D} - \mathcal{A}(\mathbf{L}^k))\|_F^2 + \lambda\|\mathbf{S}^{k}\|_1 - \frac{1}{2}\|\mathbf{S}^{k+1} - \mathbf{S}^k\|_F^2. \end{aligned} \tag{2.23}$$

Combining estimates (2.22) and (2.23), we find that

$$E(\mathbf{L}^{k+1}, \mathbf{S}^{k+1}) \leq E(\mathbf{L}^k, \mathbf{S}^k) + \left(\frac{\|\mathcal{A}\|^2}{2} - \frac{1}{2t}\right)\|\mathbf{L}^{k+1} - \mathbf{L}^k\|_F^2 - \frac{1}{2}\|\mathbf{S}^{k+1} - \mathbf{S}^k\|_F^2. \tag{2.24}$$

Since $\frac{\|\mathcal{A}\|^2}{2} - \frac{1}{2t} < 0$, the estimate above implies $E(\mathbf{L}^{k+1}, \mathbf{S}^{k+1}) \leq E(\mathbf{L}^k, \mathbf{S}^k)$ for any $k \geq 0$, which verifies the first statement.

Note that the target function $E(\mathbf{L}, \mathbf{S})$ is coercive, i.e., $E(\mathbf{L}, \mathbf{S}) \to +\infty$ when $\|\mathbf{L}\|_F + \|\mathbf{S}\|_F \to +\infty$. Since $E(\mathbf{L}^k, \mathbf{S}^k) \leq E(\mathbf{L}^0, \mathbf{S}^0) < +\infty, \forall k \geq 1$, this property guarantees that both $\{\mathbf{L}^k\}_{k\geq 1}$ and $\{\mathbf{S}^k\}_{k\geq 1}$ are bounded sequences, and thus the second statement holds.

For any limit point $(\mathbf{L}^\star, \mathbf{S}^\star)$ of $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$, there exists a convergent subsequence $\{(\mathbf{L}^{k_i}, \mathbf{S}^{k_i})\}_{i\geq 1}$ such that $\mathbf{L}^{k_i} \to \mathbf{L}^\star$ and $\mathbf{S}^{k_i} \to \mathbf{S}^\star$. On the other hand, we see that

$$\begin{aligned} \mathbf{S}^{k_i+1} &= \operatorname{sign}(\mathbf{D} - \mathcal{A}(\mathbf{L}^{k_i})) \odot \max(0, |\mathbf{D} - \mathcal{A}(\mathbf{L}^{k_i})| - \lambda), \\ \mathbf{L}^{k_i+1} &= \mathbf{prox}_{t\mu}\left(\mathbf{L}^{k_i} - t\mathcal{A}^*(\mathcal{A}(\mathbf{L}^{k_i}) + \mathbf{S}^{k_i+1} - \mathbf{D})\right). \end{aligned} \tag{2.25}$$

Summing both sides of (2.24) from $k = 0$ to ∞, we obtain

$$\left(\frac{1}{t} - \|\mathcal{A}\|^2\right)\sum_{k=0}^{\infty}\|\mathbf{L}^{k+1} - \mathbf{L}^k\|_F^2 + \sum_{k=0}^{\infty}\|\mathbf{S}^{k+1} - \mathbf{S}^k\|_F^2 \leq 2E(\mathbf{L}^0, \mathbf{S}^0) < \infty.$$

This inequality guarantees that $\{\mathbf{S}^{k_i+1}\}_{i\geq 1}$ has the same limit point $\mathbf{S}^\star$ as that of $\{\mathbf{S}^{k_i}\}_{i\geq 1}$, and $\{\mathbf{L}^{k_i+1}\}_{i\geq 1}$ has the same limit point $\mathbf{L}^\star$ as that of $\{\mathbf{L}^{k_i}\}_{i\geq 1}$. Then by taking limits in both sides of the two equations in (2.25), we obtain the third statement.

Next we will prove the last two statements. As $\|\mathbf{X}^k\|_F^2 = \|\mathbf{L}^k\|_F^2$ and $\|\mathbf{Y}^k\|_F^2 = p$, we know that the sequence $\{(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{S}^k)\}_{k\geq 1}$ is also bounded. Let $(\mathbf{X}^\star, \mathbf{Y}^\star, \mathbf{S}^\star)$ be a limit point

of $\{(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{S}^k)\}_{k\geq 1}$, which is the limitation of a subsequence $\{(\mathbf{X}^{k_i}, \mathbf{Y}^{k_i}, \mathbf{S}^{k_i})\}_{i\geq 1}$. Then we have

$$\mathbf{L}^{k_i} = \mathbf{X}^{k_i}(\mathbf{Y}^{k_i})^\top \to \mathbf{X}^\star(\mathbf{Y}^\star)^\top \text{ and } \mathbf{S}^{k_i} \to \mathbf{S}^\star,$$

i.e., $(\mathbf{X}^\star(\mathbf{Y}^\star)^\top, \mathbf{S}^\star)$ is the limit point of the subsequence $\{(\mathbf{L}^{k_i}, \mathbf{S}^{k_i})\}_{i\geq 1}$. Thus the fourth statement is verified.

Now we are in the position to prove the fifth statement. Due to the third and fourth statements, if $(\mathbf{X}^\star, \mathbf{Y}^\star, \mathbf{S}^\star)$ is a limit point of $\{(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{S}^k)\}_{k\geq 1}$, i.e., $(\mathbf{X}^\star(\mathbf{Y}^\star)^\top, \mathbf{S}^\star)$ should satisfy (2.18)

$$\begin{aligned} \mathbf{S}^\star &= \operatorname{sign}(\mathbf{D} - \mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top)) \odot \max(0, |\mathbf{D} - \mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top)| - \lambda), \\ \mathbf{X}^\star(\mathbf{Y}^\star)^\top &= \mathbf{prox}_{t\mu}\left(\mathbf{X}^\star(\mathbf{Y}^\star)^\top - t\mathcal{A}^*(\mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top) + \mathbf{S}^\star - \mathbf{D})\right). \end{aligned} \tag{2.26}$$

The first condition in (2.26) implies that the limit point $\mathbf{S}^\star$ minimizes

$$\frac{1}{2}\|\mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top) + \mathbf{S} - \mathbf{D}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \mu\|\mathbf{X}^\star(\mathbf{Y}^\star)^\top\|_*$$

over all $\mathbf{S} \in \mathbb{R}^{m\times n}$. Thus, $\mathbf{S}^\star$ should satisfy the third condition in (2.20).

Moreover, since $\operatorname{rank}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top) \leq p$, the second condition in (2.26) actually implies that $(\mathbf{X}^\star, \mathbf{Y}^\star)$ is an optimal solution of the problem

$$\underset{\mathbf{X},\mathbf{Y}}{\text{minimize}}\ \frac{1}{2}\|\mathbf{X}\mathbf{Y}^\top - \mathbf{X}^\star(\mathbf{Y}^\star)^\top + t\mathcal{A}^*(\mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top) + \mathbf{S}^\star - \mathbf{D})\|_F^2 + t\mu\|\mathbf{X}\mathbf{Y}^\top\|_*.$$

Therefore, $(\mathbf{X}^\star, \mathbf{Y}^\star)$ should satisfy the first-order optimality condition for $\mathbf{X}$, which gives

$$\begin{aligned} &[\mathbf{X}^\star(\mathbf{Y}^\star)^\top - \mathbf{X}^\star(\mathbf{Y}^\star)^\top + t\mathcal{A}^*(\mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top) + \mathbf{S}^\star - \mathbf{D})]\mathbf{Y}^\star \\ &+ t\mu\partial\|\mathbf{X}^\star(\mathbf{X}^\star(\mathbf{Y}^\star)^\top)^\top\|_*\mathbf{Y}^\star \\ =&t[\mathcal{A}^*(\mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top) + \mathbf{S}^\star - \mathbf{D}) + \mu\partial\|\mathbf{X}^\star(\mathbf{Y}^\star)^\top\|_*]\mathbf{Y}^\star \ni \mathbf{0}. \end{aligned}$$

Similarly, from the first-order opitmality condition for $\mathbf{Y}$, one can verify that

$$\mathbf{0} \in (\mathbf{X}^\star)^\top[\mathcal{A}^*(\mathcal{A}(\mathbf{X}^\star(\mathbf{Y}^\star)^\top) + \mathbf{S}^\star - \mathbf{D}) + \mu\partial\|\mathbf{X}^\star(\mathbf{Y}^\star)^\top\|_*].$$

We thus derive the first two conditions in (2.20).

We will complete our proof by verifying the convergence results for the special case of $\mathcal{A} = \mathcal{I}$ and $t = 1$. In this case, by the same method, one can derive a similar inequality as (2.24), which is

$$E(\mathbf{L}^{k+1}, \mathbf{S}^{k+1}) \leq E(\mathbf{L}^k, \mathbf{S}^k) - \frac{1}{2}\|\mathbf{S}^{k+1} - \mathbf{S}^k\|_F^2.$$

Then $\{E(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$ are non-increasing along $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$, and $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$ is bounded due to the coerciveness of $E(\mathbf{L}, \mathbf{S})$. Let $(\mathbf{L}^\star, \mathbf{S}^\star)$ be the limit point of $\{(\mathbf{L}^k, \mathbf{S}^k)\}_{k\geq 1}$ achieved by the subsequence $\{(\mathbf{L}^{k_i}, \mathbf{S}^{k_i})\}_{i\geq 1}$. Recall the iterations for updating $\mathbf{S}^{k_i+1}$ and $\mathbf{L}^{k_i}$ given by

$$\begin{aligned} \mathbf{S}^{k_i+1} &= \operatorname{sign}(\mathbf{D} - \mathbf{L}^{k_i}) \odot \max(0, |\mathbf{D} - \mathbf{L}^{k_i}| - \lambda), \\ \mathbf{L}^{k_i} &= \mathbf{prox}_{\mu}\left(\mathbf{D} - \mathbf{S}^{k_i}\right). \end{aligned} \tag{2.27}$$

Since $\sum_{k=0}^{\infty} \|\mathbf{S}^{k+1} - \mathbf{S}^k\|_F^2 \leq 2E(\mathbf{L}^0, \mathbf{S}^0) < +\infty$, $\{\mathbf{S}^{k_i+1}\}_{i\geq 1}$ has the same limit point $\mathbf{S}^\star$ as that of $\{\mathbf{S}^{k_i}\}_{i\geq 1}$. Taking limits in both sides of equations (2.27) yields the condition (2.18) for $\mathcal{A} = \mathcal{I}$ and $t = 1$. The last two statements can be verified by exactly the same arguments for the general case. We thus complete the proof. □

2.2.2 An accelerated algorithm

We show in the previous subsection that Alg. 2.1 is a forward-backward splitting or proximal gradient algorithm for a nonconvex problem. Recently, accelerated proximal gradient (APG) algorithms are proposed for nonconvex problems to reduce the computational time without sacrificing convergence (Li and Pong, 2015; Li and Lin, 2015). In this chapter, we adopt the nonmonotone APG (Li and Lin, 2015, Alg. 2) because of its better performance shown in (Li and Lin, 2015). The algorithm is described in Alg. 2.2. We let $\delta = 1$ and $\eta = 0.6$ in the numerical experiments.

2.3 Numerical experiments

In this section, we use synthetic data and real images to demonstrate the performance of our proposed model and algorithms. The code to reproduce the results in this section can be found at https://github.com/mingyan08/RPCA_Rank_Bound.

Algorithm 2.2: Accelerated RPCA with nonmonotone APG

Input: $\mathbf{D}$, μ, λ, p, $\mathcal{A}$, stepsize t, $\eta \in [0,1)$, $\delta > 0$, stopping criteria ϵ, maximum number of iterations Max_Iter, initialization: $\mathbf{L}^0 = \mathbf{L}^1 = \mathbf{Z}^1 = \mathbf{0}$, $t^0 = 0$, $t^1 = q^1 = 1$, $c^1 = F(\mathbf{L}^1)$

Output: $\mathbf{L}$, $\mathbf{S}$

for *k = 1, 2, 3, .., Max_ Iter* **do**

$\mathbf{L} = \mathbf{L}^k + \frac{t^{k-1}}{t^k}(\mathbf{Z}^k - \mathbf{L}^k) + \frac{t^{k-1}-1}{t^k}(\mathbf{L}^k - \mathbf{L}^{k-1})$;

$\mathbf{S} = \text{sign}(\mathbf{D} - \mathcal{A}(\mathbf{L})) \odot \max(0, |\mathbf{D} - \mathcal{A}(\mathbf{L})| - \lambda)$;

$\mathbf{Z}^{k+1} = \mathbf{prox}_{t\mu}(\mathbf{L} - t\mathcal{A}^*(\mathcal{A}(\mathbf{L}) - \mathbf{D} + \mathbf{S}))$;

if $F(\mathbf{Z}^{k+1}) \leq c^k - \delta\|\mathbf{Z}^{k+1} - \mathbf{L}\|^2$ **then**

$\mathbf{L}^{k+1} = \mathbf{Z}^{k+1}$;

else

$\mathbf{S}^k = \text{sign}(\mathbf{D} - \mathcal{A}(\mathbf{L}^k)) \odot \max(0, |\mathbf{D} - \mathcal{A}(\mathbf{L}^k)| - \lambda)$;

$\mathbf{V}^{k+1} = \mathbf{prox}_{t\mu}(\mathbf{L}^k - t\mathcal{A}^*(\mathcal{A}(\mathbf{L}^k) - \mathbf{D} + \mathbf{S}^k))$;

$$\mathbf{L}^{k+1} = \begin{cases} \mathbf{Z}^{k+1} & \textbf{if } F(\mathbf{Z}^{k+1}) \leq F(\mathbf{V}^{k+1}); \\ \mathbf{V}^{k+1} & \textbf{otherwise}; \end{cases}$$

end

if $\|\mathbf{L}^k - \mathbf{L}^{k-1}\|_F / \|\mathbf{L}^{k-1}\|_F < \epsilon$ **then**

break

end

$t^{k+1} = \frac{\sqrt{4(t^k)^2+1}+1}{2}$;

$q^{k+1} = \eta q^k + 1$;

$c^{k+1} = \frac{\eta q^k c^k + F(\mathbf{L}^{k+1})}{q^{k+1}}$;

end

2.3.1 Synthetic data

We would like to recover the low-rank matrix from a noisy matrix that is contaminated by a sparse matrix and Gaussian noise. We create a true low-rank 500×500 matrix $\mathbf{L}^\star$ by multiplying a random $500 \times r$ matrix and a random $r \times 500$ matrix, where their components are generated from standard normal distribution independently. We calculate the mean of the absolute values of all the components in $\mathbf{L}^\star$ and denote it as c. Then we randomly select $s\%$ of the components and replace their values with uniformly distributed random values from $[-3c, 3c]$. After that, we add small Gaussian noise $\mathcal{N}(0, \sigma^2)$ to all components of the matrix. We let $t = 1.7$ in the experiments because of fast convergence, though the convergence results in Theorem 2.2.4 require $t < 1$.

2.3.1.1 Low-rank matrix recovery

We fix $\sigma = 0.05$ for the Gaussian noise and set the upper bound of the rank to be $p = r + 5$. We stop all algorithms when the relative error at the k-th iteration, which is defined as

$$RE(\mathbf{L}^{k+1}, \mathbf{L}^k) := \frac{\|\mathbf{L}^{k+1} - \mathbf{L}^k\|_F}{\|\mathbf{L}^k\|_F},$$

is less than 10^{-4}. We use the relative error to $\mathbf{L}^\star$, which is defined as

$$RE(\mathbf{L}, \mathbf{L}^*) := \frac{\|\mathbf{L} - \mathbf{L}^\star\|_F}{\|\mathbf{L}^\star\|_F},$$

to evaluate the performance of our proposed model and that in (Shen et al., 2019). First, we consider the case with $r = 25$ and $s = 20$. We plot a contour map of the relative error to $\mathbf{L}^\star$ for different parameters μ and λ in Fig. 2.1. From this contour map, we can see that the best parameter does not happen when $\mu = 0$, which corresponds to the model in (Shen et al., 2019). It verifies the better performance of our proposed model with appropriate parameters. In this subsection, we set $\lambda = 0.02$ for Shen et al.'s and ($\mu = 0.6,\ \lambda = 0.04$) for our proposed algorithms.

In addition, we consider another two settings for (r, s), and the comparison with different algorithms is shown in Table 2.1. In this table, we also compare the number of iterations for three algorithms: Shen et al.'s, Alg. 2.1, and Alg. 2.2. From this table, we can see that both Alg. 2.1 and Alg. 2.2 have better performance and fewer iterations than (Shen et al., 2019). The accelerated Alg. 2.2 has the fewest iterations, but its performance in terms of $RE(\mathbf{L}, \mathbf{L}^\star)$ is not as good as Alg. 2.1 for the last case. It is because we stop both algorithms when the stopping criteria is satisfied, and the algorithms are not converged yet. We checked the objective function values for both algorithms, and the value for Alg. 2.2 is smaller than that for Alg. 2.1 in this case. Therefore, if we want a solution close to the true low-rank matrix $\mathbf{L}^\star$, we may need to stop early before the convergence, which is the same as many models for inverse problems.

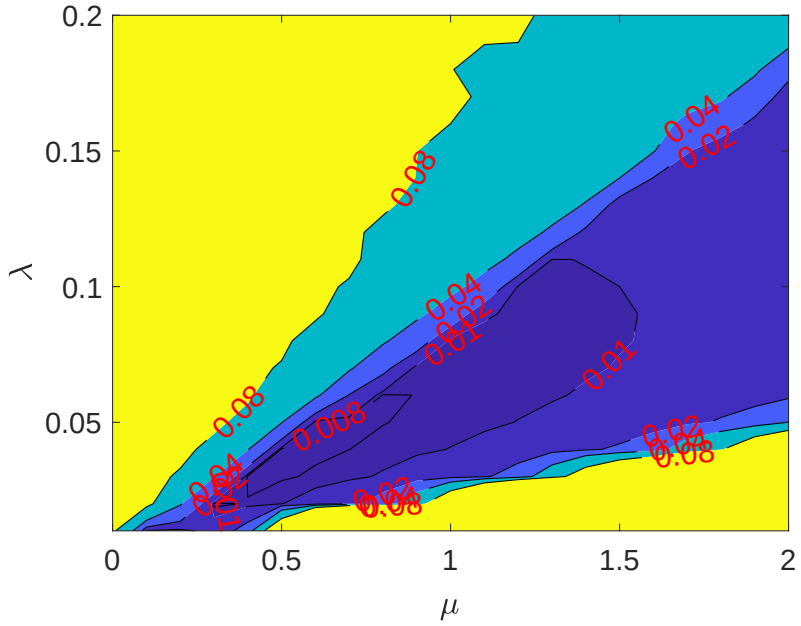


Figure 2.1: The contour map of the relative error to $\mathbf{L}^\star$ for different parameters. In this experiment, we set $r = 25$ and $s = 20$. The upper bound of the rank is set to be $p = 30$.

r	s	Shen et al.'s (Shen et al., 2019)		Alg. 1		Alg.2	
		$RE(\mathbf{L}, \mathbf{L}^\star)$	# iter	$RE(\mathbf{L}, \mathbf{L}^\star)$	# iter	$RE(\mathbf{L}, \mathbf{L}^\star)$	# iter
25	20	0.0745	1318	0.0075	296	0.0075	68
50	20	0.0496	1434	0.0101	473	0.0088	77
25	40	0.0990	2443	0.0635	796	0.0915	187

Table 2.1: Comparison of three RPCA algorithms. We compare the relative error of their solutions to the true low-rank matrix and the number of iterations. Both Alg. 2.1 and Alg. 2.2 have better performance than (Shen et al., 2019) in terms of the relative error and the number of iterations. Alg. 2.2 has the fewest iterations but the relative error could be large. It is because the true low-rank matrix is not the optimal solution to the optimization problem, and the trajectory of the iterations moves close to $\mathbf{L}^\star$ before it approaches the optimal solution.

2.3.1.2 Robustness of the model

In this experiment, we compare the robustness of our proposed model with that of (Shen et al., 2019). We let $r = 25$ and $s = 20$. Then we run both models for p from 15 to 35. The comparison of the relative error to $\mathbf{L}^\star$ is shown in Fig. 2.2. We let $\lambda = 0.02$ for Shen et al.'s and ($\mu = 0.6,\ \lambda = 0.04$) for Alg. 2.2. It shows that our proposed model is robust to the

parameter p, as long as it is not smaller than the true rank r.

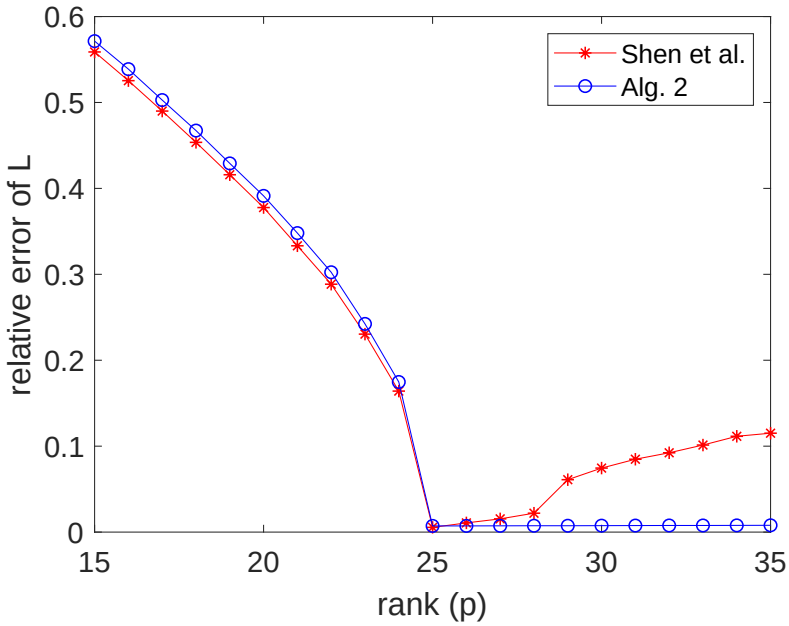


Figure 2.2: The relative error to the true low-rank matrix vs the rank p for Shen et al.'s and Alg. 2.2. Alg. 2.2 is robust to p, as long as p is not smaller than the true rank 25.

2.3.1.3 Low-rank matrix recovery with missing entries

In this experiment, we try to recover the low-rank matrix when there are missing entries in the matrix. Therefore, the operator $\mathcal{A}$ is not the identity $\mathcal{I}$. We randomly select the missing entries from all the entries. We let $r = 25$ and add both the sparse noise with parameter s and the Gaussian noise with parameter σ to the true matrix $\mathbf{L}^\star$. Then we apply Alg. 2.2 to recover the low-rank matrix, and the relative error to $\mathbf{L}^\star$ is used to evaluate the performance. The results for different settings are in Table 2.2. For the first three cases with $s = 20$, we choose $(\mu = 0.5,\ \lambda = 0.04)$, while we let $(\mu = 0.1,\ \lambda = 0.01)$ for the last case with $s = 5$. Note that, even with missing entries, Alg. 2.2 can reconstruct the low-rank matrix accurately.

s	σ	ratio of missing entries	$RE(\mathbf{L}, \mathbf{L}^\star)$ by Alg. 2.2
20	0.05	10%	0.0079
20	0.05	20%	0.0088
20	0.05	50%	0.0201
5	0.01	50%	0.0015

Table 2.2: Performance of Alg. 2.2 on low-rank matrix recovery with missing entries. We change the level of sparsity in the sparse noise, standard deviation of the Gaussian noise, and the ratio of missing entries.

2.3.2 Real image experiment

In this section, we consider the three algorithms applied to image processing problems. Since natural images are not low-rank essentially, we consider two cases on two different images ('cameraman' and 'Barbara'). For the 256×256 cameraman image (the pixel values are from 0 to 255), we create an image with rank 37 from a low-rank approximation of the original image. Then we add 20% salt and pepper impulse noise and Gaussian noise with standard variance 4. We set 42 as the upper bound of the rank of the low-rank image for all algorithms. We let $\lambda = 0.03$ for Shen et al. and ($\mu = 0.5,\ \lambda = 0.06$) for our model. To compare the performance of both models, we use the relative error defined in the last subsection and peak signal to noise ratio (PSNR) defined as

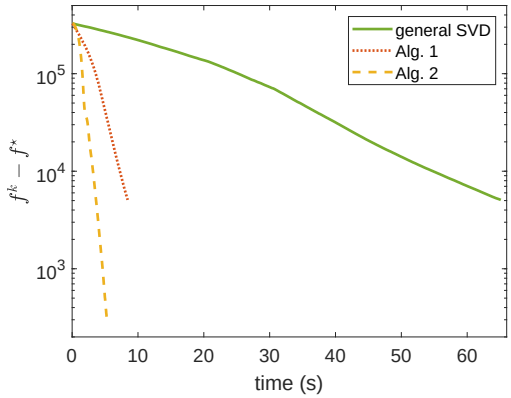
$$\text{PSNR} := 10 \log_{10} \frac{\text{Peak\_Val}^2}{\text{MSE}}.$$

Here Peak_Val is the largest value allowed at a pixel (255 in our case), and MSE is the mean squared error between the recovered image and the true image. The numerical results are shown in Fig. 2.3. From Fig. 2.3(A-C), we can see that our proposed model performs better than Shen et al. (Shen et al., 2019). For the proposed model, we also compare the speed of three algorithms: Alg. 2.1, Alg. 2.1 with standard SVD, and Alg. 2.2 in Fig. 2.3(D). For both plots, we can see that the Gauss-Newton approach increases the speed comparing to the standard SVD approach. From the decrease of the objective function value, we can see that the accelerated algorithm Alg. 2.2 is faster than the nonaccelerated Alg. 2.1.

(a) Corrupted image
RE: 0.4760, PSNR: 12.76

(b) Recovered by Shen et al.
RE: 0.1736, PSNR: 21.52

(c) Recovered by Alg. 2.2
RE: 0.0457, PSNR:33.11



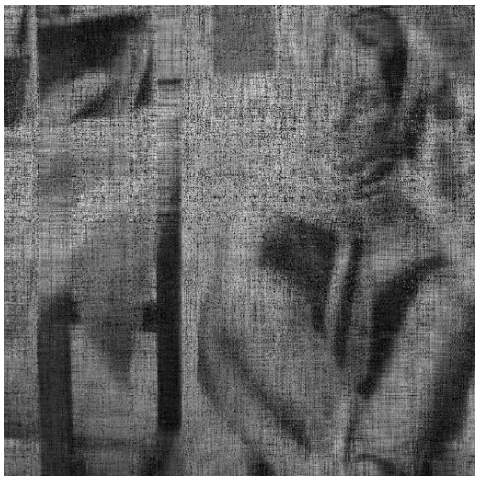
(d) Comparison of the objective function value vs time for three algorithms

Figure 2.3: The numerical experiment on the 'cameraman' image. (A-C) show that the proposed model performs better than Shen et al.'s both visually and in terms of RE and PSNR. (D) compares the objective values vs time for general SVD, Alg. 2.1, and Alg. 2.2. Here $f^\star$ is the value obtained by Alg. 2.2 with more iterations. It shows the fast speed with the Gauss-Newton approach and acceleration. With the Gauss-Newton approach, the computation time for Alg. 2.1 is reduced to about 1/7 of the one with standard SVD (from 65.11s to 8.43s). The accelerated Alg. 2.2 requires 5.2s, though the number of iterations is reduced from 3194 to 360.

Next, we use the original 512×512 barbara image (the pixel values are from 0 to 255) without modification and add the same two types of noise as in the cameraman image. Because the original image is not low-rank, we choose the upper bound of rank $p = 50$. We let $\lambda = 0.03$ for Shen et al. and ($\mu = 0.5,\ \lambda = 0.06$) for our model. The comparison

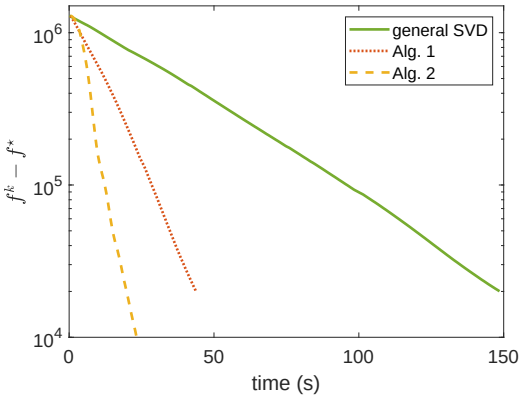
result is shown in Fig. 2.4, and it is similar to the cameraman image. We also applied the acceleration to Shen et al.'s algorithm and obtained a better image with RE = 0.1447 and PSNR = 22.37.

(a) Corrupted image
RE: 0.4821, PSNR: 11.91



(b) Recovered by Shen et al
RE: 0.3368, PSNR: 15.03

(c) Recovered by Alg. 2.2
RE: 0.1317, PSNR: 23.18



(d) Comparison of the objective function value vs time for three algorithms

Figure 2.4: The numerical experiment on the 'Barbara' image. (A-C) show that the proposed model performs better than Shen et al.'s both visually and in terms of RE and PSNR. (D) compares the objective values vs time for general SVD, Alg. 2.1, and Alg. 2.2. Here $f^\star$ is the value obtained by Alg. 2.2 with more iterations. It shows the fast speed with the Gauss-Newton approach and acceleration. With the Gauss-Newton approach, the computation time for Alg. 2.1 is reduced to less than 1/3 of the one with standard SVD (from 148.6s to 43.7s). The accelerated Alg. 2.2 requires 23.3s, though the number of iterations is reduced from 3210 to 300.

2.4 Concluding remarks

In this chapter, we introduced a new model for RPCA when an upper bound of the rank is provided. For the unconstrained RPCA problem, we formulate it as the sum of one smooth function and one nonsmooth nonconvex function. Then we derive an algorithm based on proximal-gradient. This proposed algorithm has the alternating minimization algorithm (Shen et al., 2019) as a special case. Because of the connection between this algorithm and proximal gradient, we adopted an acceleration approach and proposed an accelerated algorithm. Both proposed algorithms have two advantages comparing to existing algorithms. First, different from algorithms that require accurate rank estimations, the proposed algorithms are robust to the upper bound of the rank. Second, we apply the Gauss-Newton algorithm to avoid the computation of singular values for large matrices, so our algorithm is faster than those algorithms that require SVD. Except for problem (2.7), this algorithm can be generalized to solve many other variants.

2.4.1 Nonconvex penalties on the singular values

In the problem (2.7), we choose the convex nuclear norm for the low-rank component in the objective function, which is the ℓ_1 norm on the singular values. The ℓ_1 norm pushes all singular values toward zero for the same amount, bringing bias in the solution. To promote the low-rankness of the low-rank component (or sparsity of its singular values), we can choose nonconvex regularization terms for the singular values. The idea for nonconvex regularization is to reduce the bias by pushing less on larger singular values. Some examples of nonconvex regularization are ℓ_p $(0 \leq p < 1)$ (Chartrand, 2007), smoothly clipped absolute deviation (SCAD) (Fan and Li, 2001), minimax concave penalty (MCP) (Zhang et al., 2010), nonconvex weighted ℓ_1 (Huang et al., 2015), etc. When these regularization terms are applied, the only difference is in the third step for finding $\mathbf{X}$ in Lemma 2.2.3. Currently, we have to apply the soft thresholding on the singular values. When nonconvex regularization is

used, we apply the corresponding thresholding on the singular values. In this case, all the convergence results stay valid.

2.4.2 Other regularization on the sparse component

We can also replace the ℓ_1 norm of the sparse component with other regularization terms. Similarly to the penalty on the singular values, the ℓ_1 norm on the sparse component brings bias, and we can use nonconvex regularization terms. Wen et al. (2019) uses both nonconvex regularization terms for the low-rank and sparse components. When different regularization terms are used on the sparse component, the new function f_λ (see (2.12) for the definition) may not be differentiable any more. In this case, the convergence results do not hold.

2.4.3 Constrained problems

When there is no noise in the measurements, the problem becomes constrained, and the previous algorithm can not be applied directly. Shen et al. (2019) uses the penalty method and gradually increases the weight for the penalization to approximate the constrained problem. Here, we introduce a new method based on ADMM. We consider the following constrained problem

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}}\ \mu\|\mathbf{L}\|_* + \|\mathbf{S}\|_1,\ \text{subject to}\ \text{rank}(\mathbf{L}) \le p,\ \mathbf{D} = \mathbf{L} + \mathbf{S}. \tag{2.28}$$

When we apply ADMM, the steps are

$$\mathbf{L}^{k+1} = \underset{\mathbf{L}:\text{rank}(\mathbf{L})\le p}{\arg\min}\ \mu\|\mathbf{L}\|_* + \frac{\alpha}{2}\|\mathbf{D} - \mathbf{L} - \mathbf{S}^k + \frac{\mathbf{Z}^k}{\alpha}\|_F^2; \tag{2.29a}$$

$$\mathbf{S}^{k+1} = \underset{\mathbf{S}}{\arg\min}\ \|\mathbf{S}\|_1 + \frac{\alpha}{2}\|\mathbf{D} - \mathbf{L}^{k+1} - \mathbf{S} + \frac{\mathbf{Z}^k}{\alpha}\|_F^2; \tag{2.29b}$$

$$\mathbf{Z}^{k+1} = \mathbf{Z}^k - \alpha(\mathbf{L}^{k+1} + \mathbf{S}^{k+1} - D). \tag{2.29c}$$

The first step is exactly the proximal operator that can be solved from Lemma 2.2.3. The other two steps are easy to compute. This algorithm has only one parameter α, while penalty

methods, such as that in (Shen et al., 2019), require additional parameters to increase the weight for the penalization.

CHAPTER 3

ROBUST PRINCIPAL COMPONENT ANALYSIS FOR SEISMIC EVENT DETECTION

3.1 Introduction

Reflected seismic data is contaminated by both random and coherence noise. Random noise is usually caused by environmental inferences, and coherent noise is mostly generated by the source. Denoising is an important preprocessing step because noisy seismic data may lead to unrealistic artifacts in the inversion or imaging results. But, it is challenging to effectively and efficiently eliminate noise from noisy seismic data.

Various seismic denoising methods have been developed to remove random noise (Yu et al., 2015; Fomel and Liu, 2013; Kreimer and Sacchi, 2012) and coherent noise (Weglein, 2016; Liu and Fomel, 2013; Herman and Perkins, 2006). Our technique belongs to the sparse-transform-based methods. In this type of methods, it was shown that the signal could be sparsely represented by a basis/dictionary or it is sparse after some transform such as wavelet transform. In particular, different methods based on sparsity are proposed (Chen et al., 2016; Rubinstein et al., 2010). Since the signal received at different receivers are correlated, the measurements lie in a low dimensional space. Dimension reduction techniques are also applied to model the signal. They include empirical mode decomposition based methods (Kopsinis and McLaughlin, 2009; Chen et al., 2017), singular spectral analysis (Qiao et al., 2017), RPCA (Candès et al., 2011; Cheng et al., 2015), and Cadzow filtering.

Though RPCA-based denoising methods achieve promising results in many applications (Cheng et al., 2015; Sun et al., 2014; Duarte et al., 2012), existing algorithms are slow and yield a large amount of computational time, especially for large-scale data set. Therefore, we develop new algorithms for RPCA and its nonconvex variants. To verify the performance of these algorithms, we apply them to both synthetic and field data. Through

the numerical experiments, our proposed algorithms significantly improve computational efficiency and yield comparable or better denoising results.

3.2 Theory

In this section, we focus on RPCA, though the technique can be applied to other methods. Given the noisy data denoted as an $n_1 \times n_2$ matrix $\mathbf{D}$, where n_1 and n_2 are the number of receivers and measurements at each receiver, respectively. The goal of RPCA is to get a low-rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$ from this noisy matrix such that $\mathbf{L} + \mathbf{S} = \mathbf{D}$. Ma and Aybat (2018) reviewed several forms of RPCA and efficient algorithms. One formula is

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \text{rank}(\mathbf{L}) + \lambda \|\mathbf{S}\|_0, \ \text{subject to} \ \mathbf{L} + \mathbf{S} = \mathbf{D}, \tag{3.1}$$

where $\text{rank}(\mathbf{L})$ is the rank of the matrix $\mathbf{L}$, $\|\mathbf{S}\|_0$ is the number of nonzero elements in the matrix $\mathbf{S}$, and λ is a parameter to balance these two terms. This model assumes that the data is corrupted by sparse noise, which is modeled as a sparse matrix $\mathbf{S}$. However, in practice, especially in the collected seismic data, $\mathbf{D}$ often includes other types of noise such as Gaussian noise, denoted as $\mathbf{N}$. We set the random noise level to be σ, namely, $\|\mathbf{N}\|_F^2 \leq \sigma$, where $\|\cdot\|_F$ is the Frobenious norm. Hence, the corresponding formula becomes

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \text{rank}(\mathbf{L}) + \lambda \|\mathbf{S}\|_0, \ \text{subject to} \ \|\mathbf{D} - \mathbf{L} - \mathbf{S}\|_F^2 \leq \sigma. \tag{3.2}$$

This problem is NP-hard (Ma and Aybat, 2018), and direct numerical calculation is impossible. One direction is convexification. It has been proved in (Zhou et al., 2010b) that a relaxed version –principal component pursuit– can be defined as

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \ \text{subject to} \ \|\mathbf{D} - \mathbf{L} - \mathbf{S}\|_F^2 \leq \sigma. \tag{3.3}$$

The above minimization problem yields a stable estimate of the low-rank matrix and the sparse matrix under some conditions for $\mathbf{D}$. Here, $\|\cdot\|_*$ is the nuclear norm defined as the sum of all singular values, and $\|\cdot\|_1$ is the sum of the absolute values of all elements.

The constrained problem (3.3) is equivalent to the following unconstrained one

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \ \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \tfrac{1}{2\mu}\|\mathbf{D}-\mathbf{L}-\mathbf{S}\|_F^2. \tag{3.4}$$

That is, given σ in (3.3), we can find μ such that the optimal solutions for both (3.3) and (3.4) are equivalent, and vice versa. Therefore, we focus on solving Eq. (3.4) instead of Eq. (3.3). This problem is convex, and many existing convex optimization algorithms are applied. Some examples are proximal gradient method (PGM), accelerated PGM, and alternating direction methods of multipliers (ADMM). We provide some brief introduction over those methods.

We consider the two matrices $\mathbf{L}$ and $\mathbf{S}$ together. The last function in (3.4) is differentiable with respect to $\mathbf{L}$ and $\mathbf{S}$, while the first two functions are separable. One iteration of PGM can be expressed as

$$\begin{cases} \bar{\mathbf{L}}^k = \mathbf{L}^k - \frac{t}{\mu}(\mathbf{L}^k + \mathbf{S}^k - \mathbf{D}), \\ \bar{\mathbf{S}}^k = \mathbf{S}^k - \frac{t}{\mu}(\mathbf{L}^k + \mathbf{S}^k - \mathbf{D}), \\ \mathbf{L}^{k+1} = \arg\min_{\mathbf{L}} \ t\|\mathbf{L}\|_* + \frac{1}{2}\|\mathbf{L} - \bar{\mathbf{L}}^k\|_F^2, \\ \mathbf{S}^{k+1} = \arg\min_{\mathbf{L}} \ t\lambda\|\mathbf{S}\|_1 + \frac{1}{2}\|\mathbf{S} - \bar{\mathbf{S}}^k\|_F^2, \end{cases} \tag{3.5}$$

where $t \in (0, \mu)$ is the stepsize.

PGM has the convergence rate of $O(1/k)$ for general convex functions. Acceleration techniques improve the convergence rate to $O(1/k^2)$. Some examples are provided in (Nesterov, 2013, 2005; Kim and Fessler, 2016; Beck and Teboulle, 2009a,b). Beck and Teboulle (2009a) develop fast iterative shrinkage-thresholding algorithm (FISTA) by applying the proximal operator on an extrapolated point. By denoting one iteration of PGM as $(\mathbf{L}^{k+1}, \mathbf{S}^{k+1}) = \text{PGM}(\mathbf{L}^k, \mathbf{S}^k)$, the iteration of FISTA is equivalent to

$$\begin{aligned} (\hat{\mathbf{L}}^k, \hat{\mathbf{S}}^k) &= (\mathbf{L}^k, \mathbf{S}^k) + \tfrac{\theta_{k-1}-1}{\theta_k}(\mathbf{L}^k - \mathbf{L}^{k-1}, \mathbf{S}^k - \mathbf{S}^{k-1}), \\ (\mathbf{L}^{k+1}, \mathbf{S}^{k+1}) &= \text{PGM}(\hat{\mathbf{L}}^k, \hat{\mathbf{S}}^k), \end{aligned}$$

where $\theta_k = (1 + \sqrt{1 + 4\theta_{k-1}^2})/2$ and $\theta_0 = 1$. FISTA requires fewer iterations than PGM with similar per-iteration cost, which is demonstrated in our numerical experiments.

ADMM (Boyd et al., 2011; Yuan and Yang, 2013) solves a constrained problem. By introducing a new matrix $\mathbf{Z}$, an equivalent formulation can be obtained

$$\underset{\mathbf{Z},\mathbf{L},\mathbf{S}}{\text{minimize}} \ \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \tfrac{1}{2\mu}\|\mathbf{Z}\|_F^2 \text{ s.t. } \mathbf{Z}+\mathbf{S}+\mathbf{L}=\mathbf{D}. \tag{3.6}$$

Though the convergence of three-block ADMM is not guaranteed for general problems, this ADMM for RPCA converges with an appropriately chosen parameter (Wang et al., 2019).

These algorithms are computationally efficient for small matrices, but they suffer slow convergent rates when handling medium to large-scale data sets, such as seismic data. The efficiency of the algorithms are directly related to the number of the unknown variables. PGM and FISTA have two matrices as unknown variables, while ADMM has three. In this abstract, we use infimal convolution to reduce unknown variables to be one matrix and obtain faster algorithms than existing ones.

Besides solving the convex formula (3.4), people also solve nonconvex ones (Zhou and Tao, 2011, 2013) because of their better performance. In this abstract, we also consider a nonconvex model by replacing the L_1 term with a nonconvex term to show the robustness of our algorithms.

Methodology and Algorithms

3.2.1 New algorithms with infimal convolution

The problem (3.4) has two unknown matrices $\mathbf{L}$ and $\mathbf{S}$, and only the last two terms have $\mathbf{S}$. So we can eliminate $\mathbf{S}$ by finding the optimal $\mathbf{S}$ with a given $\mathbf{L}$

$$h(\mathbf{D}-\mathbf{L}) := \min_{\mathbf{S}} \ \lambda\|\mathbf{S}\|_1 + \tfrac{1}{2\mu}\|\mathbf{D}-\mathbf{L}-\mathbf{S}\|_F^2. \tag{3.7}$$

By defining $f(\mathbf{S}) = \lambda\|\mathbf{S}\|_1$ and $g(\mathbf{X}) = \frac{1}{2\mu}\|\mathbf{X}\|_F^2$, we will have h as the infimal convolution of f and g, which is defined as $f\square g : x \to \min_y f(y) + g(x-y)$. Hence, the problem (3.4) can be reduced to

$$\underset{\mathbf{L}}{\text{minimize}} \ \|\mathbf{L}\|_* + f\square g(\mathbf{D}-\mathbf{L}), \tag{3.8}$$

which contains only one unknown matrix $\mathbf{L}$. The infimal convolution $f\square g(\mathbf{D}-\mathbf{L})$ is differentiable with respect to $\mathbf{L}$, and its gradient is $\frac{1}{\mu}$-Lipschitz continuous. So we apply PGM and FISTA to solve the problem (3.8) with $\mathbf{L}$ only. We name the corresponding algorithms as IC-PGM and IC-FISTA. We also apply ADMM with two blocks and name the new algorithm as IC-ADMM. Its convergence is well studied, and there is no restriction in choosing its parameters. For simplicity, we only provide the iteration of IC-PGM as below

$$\begin{cases} \bar{\mathbf{S}}^k = \mathbf{S}^k - (\mathbf{L}^k + \mathbf{S}^k - \mathbf{D}), \\ \mathbf{S}^{k+1} = \arg\min\limits_{\mathbf{L}} \ \mu\lambda\|\mathbf{S}\|_1 + \frac{1}{2}\|\mathbf{S} - \bar{\mathbf{S}}^k\|_F^2, \\ \bar{\mathbf{L}}^k = \mathbf{L}^k - \frac{t}{\mu}(\mathbf{L}^k + \mathbf{S}^{k+1} - \mathbf{D}), \\ \mathbf{L}^{k+1} = \arg\min\limits_{\mathbf{L}} \ t\|\mathbf{L}\|_* + \frac{1}{2}\|\mathbf{L} - \bar{\mathbf{L}}^k\|_F^2. \end{cases} \tag{3.9}$$

Here, the stepsize of $t \in (0, 2\mu)$, which is larger than PGM. The derivation of IC-FISTA is similar to IC-PGM.

3.2.1.1 Comparison between PGM and IC-PGM

Comparing the steps in (3.5) and (3.9), we notice that PGM updates $\mathbf{L}$ and $\mathbf{S}$ simultaneously, while IC-PGM updates $\mathbf{S}$ first and use the updated $\mathbf{S}$ to update $\mathbf{L}$. The improved performance of IC-PGM over conventional PGM comes from two folds. Firstly, alternative update is faster than simultaneous update, which is similar to the improvement of Gauss-Seidel over Jacobian methods for solving linear equations. Secondly, IC-PGM essentially solves the problem with $\mathbf{L}$ only, which allows a larger stepsize than the conventional PGM.

A new nonconvex model

Though IC-FISTA solves RPCA efficiently, RPCA is still a convex relaxed model. In order to obtain better performance, we consider nonconvex models that we can apply the infimal convolution technique to get fast algorithms. There are many nonconvex penalties, please see (Huang and Yan, 2018; Wen et al., 2018). In this abstract, we choose Minimax Concave

Penalty (MCP) (Zhang et al., 2010) to replace the L_1 term $\lambda\|\mathbf{S}\|_1$ in RPCA. When we apply IC-PGM or IC-FISTA to solve this nonconvex problem, we just need to replace the step for updating $\mathbf{S}$:

$$\mathbf{S}^{k+1} = \arg\min_{\mathbf{L}} \ \mu r(\mathbf{S}) + \tfrac{1}{2}\|\mathbf{S} - \bar{\mathbf{S}}^k\|_F^2,$$

where $r(\mathbf{S}) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} g_{\lambda,b}(\mathbf{S}_{i,j})$ is the MCP function with

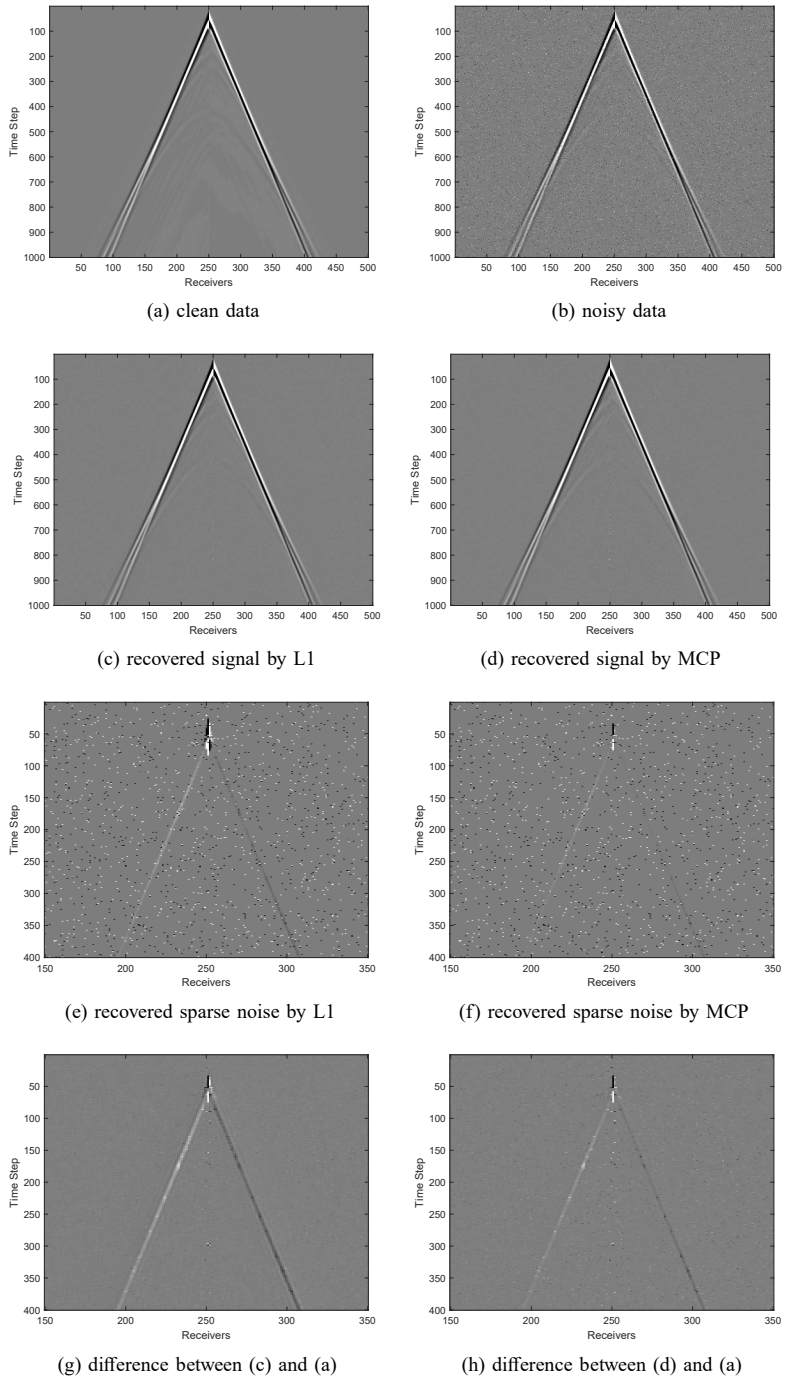
$$g_{\lambda,b}(\mathbf{S}_{i,j}) = \begin{cases} \lambda|\mathbf{S}_{i,j}| - \frac{\mathbf{S}_{i,j}^2}{2b}, & |\mathbf{S}_{i,j}| \le b\lambda, \\ \frac{b\lambda^2}{2}, & |\mathbf{S}_{i,j}| > b\lambda. \end{cases} \tag{3.10}$$

Here b is a parameter. When b goes to infinity, $r(\mathbf{S})$ becomes the L_1 term.

3.3 Results

3.3.1 Synthetic seismic data

We first test our denoising algorithms on synthetic seismic data. The seismic measurements are collections of synthetic seismograms obtained by implementing forward modeling on a velocity model with a few layers. One common-shot gather of synthetic seismic data with 500 receivers is posed at the top surface of the model. The interval between two receivers is 5 m. We use a Ricker wavelet with a center frequency of 25 Hz as the source time function and a staggered-grid finite-difference scheme with a perfectly matched layered absorbing boundary condition to generate 2D synthetic seismic reflection data (Tan and Huang, 2014). The synthetic trace at each receiver is a collection of time-series data of length $1,000$. We add two types of noise onto the seismic data. First, we add 25.2 dB Gaussian noise. Then, we choose 2% of the data and reset their values with random numbers uniformly distributed in $[-u, u]$ with u being three multiply the largest value in the clean data. The overall signal to noise ratio (SNR) is -26.2 dB.



(a) clean data (b) noisy data

(c) recovered signal by L1 (d) recovered signal by MCP

(e) recovered sparse noise by L1 (f) recovered sparse noise by MCP

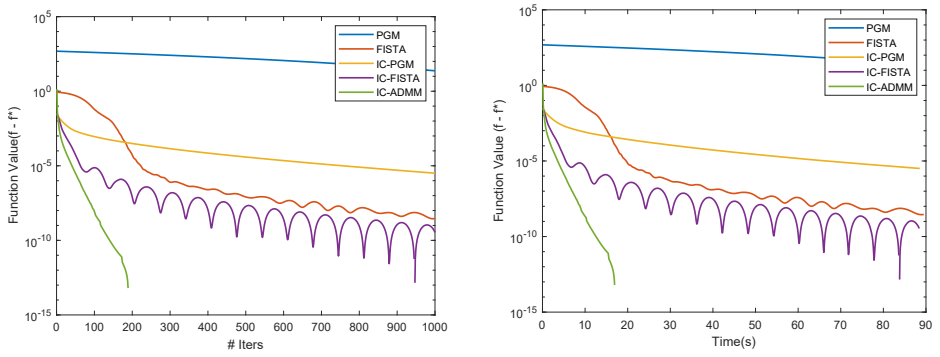
(g) difference between (c) and (a) (h) difference between (d) and (a)

Figure 3.1: Comparison of recovered results on synthetic seismic data with 500 receivers and 1000 measurements at each receiver. (a) simulated clean data. (b) noisy data (-26.2 dB). (c) recovered data by L_1 (13.4 dB). (d) recovered data by MCP (13.9 dB). (e) recovered sparse noise by L_1. (f) recovered sparse noise by MCP. (g) the difference between the clean data and the recovered one by L_1. (h) the difference between the clean data and the recovered one by L_1. (e-h) zoom-in over receivers 150-350 and measurements 1-400.

We first compare the recovery results for convex and nonconvex models in Fig. 3.1. We

manually tune the parameters to obtain the best denoising results for both models. For the L_1 penalty, we choose $\mu = 3 \times 10^{-5}$ and $\lambda = 0.12$, while for MCP, we choose $\mu = 1 \times 10^{-5}$, $\lambda = 0.135$, and $b = 10$. Both models can remove noise from the noisy data. The SNR values of recovered data for both L_1 and MCP are 13.4 dB and 13.9 dB, respectively. The figure confirms that the nonconvex model performs slightly better than the convex one.

Next, we compare the efficiency of all algorithms on the convex RPCA model. We first run IC-ADMM for 1,000 iterations to obtain an estimation for the minimal objective value. Then we compare their performance in function values with respect to the iteration number and time in Fig 3.2. IC-ADMM has the fastest convergent rate and smallest computational time among all five algorithms. By applying our IC technique, traditional algorithms (PGM and FISTA) yield better convergent rates. Comparing Fig. 3.2(a) and Fig. 3.2(b), we note that the time for each iteration is similar for all algorithms.



(a) performance-iteration (b) performance-time

Figure 3.2: Comparison of five algorithms (PGM, FISTA, IC-PGM, IC-FISTA, IC-ADMM) for the convex RPCA on synthetic data. IC-ADMM has the fastest convergence rate and smallest computational time. IC technique improves the performance of PGM and FISTA significantly.

3.3.2 Field seismic data

In this section, we apply our denoising algorithms to the field data collected from the IRIS Community Wavefield Experiment in Oklahoma (Kent et al., 2016). Fig. 3.3 shows the data collected by 220 major seismic sensors to detect earthquakes. Before applying RPCA to

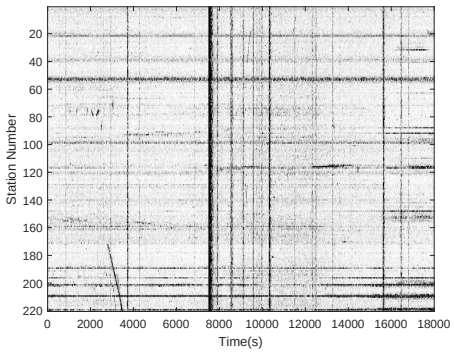


Figure 3.3: Noisy data generated in Oklahoma.

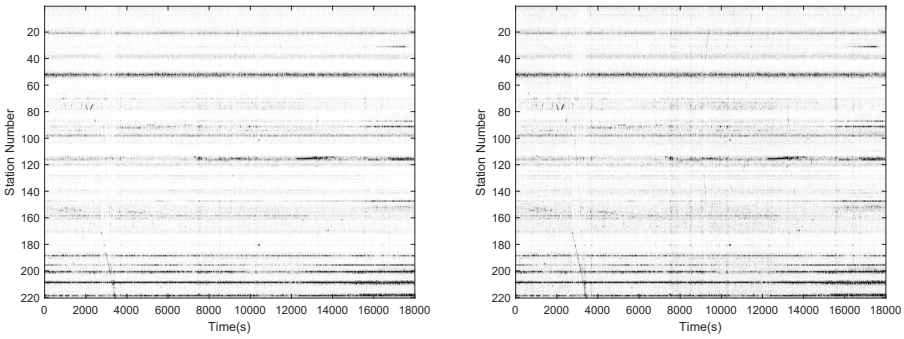
recover the data, we applied discrete cosine transform on the time domain. The comparison between L_1 and MCP is in Fig. 3.4. We set $\mu = 3 \times 10^4$, $\lambda = 3 \times 10^{-3}$, and $b = 3 \times 10^6$. The stopping criteria is $\max(\|\mathbf{S}^{k+1} - \mathbf{S}^k\|_F^2/\|\mathbf{S}^k\|_F^2, \|\mathbf{L}^{k+1} - \mathbf{L}^k\|_F^2/\|\mathbf{L}^k\|_F^2) < 10^{-3}$. As shown in Fig. 3.4, both models successfully separate the horizontal signal (we do not want) and vertical signal (we want).

In addition, we compare the computation time and total numbers of iterations in Table 2.1. IC technique improves the performance of conventional algorithms. They are 4-10x

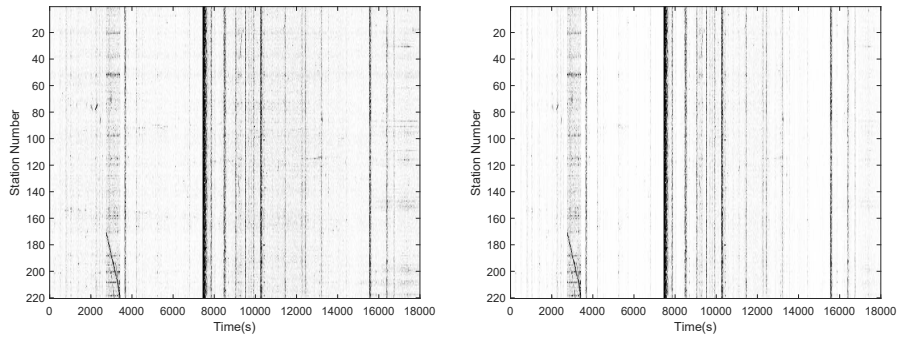
Algorithm	# Iters	Time (s)	Function vlaue
PGM	940	9.23×10^3	4.92×10^7
FISTA	166	1.63×10^3	4.92×10^7
IC-PGM	81	1.07×10^3	4.92×10^7
IC-FISTA	43	6.41×10^2	4.92×10^7
IC-ADMM	25	2.69×10^2	4.92×10^7
MCP	49	5.74×10^2	3.78×10^7

Table 3.1: Comparison of six algorithms. IC-ADMM is the fastest, which is the same as synthetic data. The function value for MCP is smaller because of a different model.

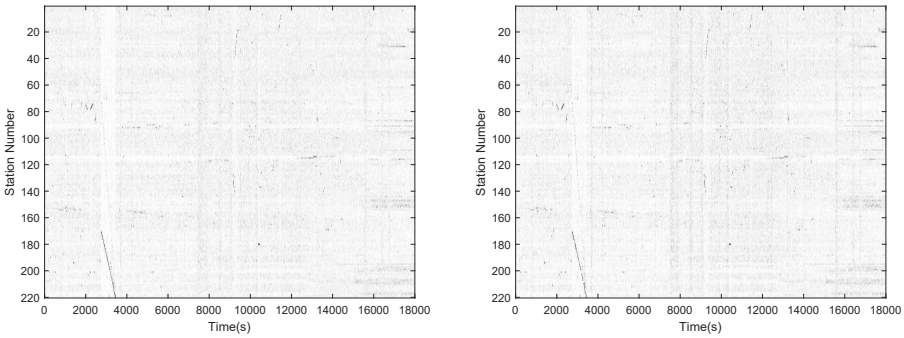
faster than non-IC algorithms. It also shows that the non-convex MCP model has comparable performance as the convex one, while the algorithm for this nonconvex model is also fast.



(a) recovered horizontal signal by L1 (b) recovered horizontal signal by MCP



(c) recovered vertical signal by L1 (d) recovered vertical signal by MCP



(e) recovered random noise by L1 (f) recovered random noise by MCP

Figure 3.4: Recovered results of the real data with two models.

3.4 Conclusion

In this section, we developed new seismic denoising algorithms based on RPCA. In particular, we applied infimal convolution to solve the convex and nonconvex optimization problems. Our technique not only allows a large stepsize, but also reduces the number of unknown variables to solve. All these characteristics of our new algorithms result in a significantly improved computational efficiency by comparing to other conventional algorithms such as PGM

and FISTA. We verified the performance of our algorithms using both synthetic reflection seismic data and field data. We observe at least a speed-up ratio of 4-10x over conventional algorithms with a comparable or better denoised results.

CHAPTER 4

MANIFOLD DENOISING BY NONLINEAR ROBUST PRINCIPAL COMPONENT ANALYSIS

4.1 Introduction

Manifold and graph learning are nowadays widely used in computer vision, image processing, and biological data analysis on tasks such as classification, anomaly detection, data interpolation, and denoising. In most applications, graphs are learned from high dimensional data, and successfully learned graphs allow traditional data analysis methods (PCA, Fourier analysis, clustering algorithm, neural networks) to be performed in conjunction with prior knowledge of the graph connectivity (Hammond et al., 2011; Jianbo Shi and Malik, 2000; Jiang et al., 2013; Meila and Shi, 2001). However, the quality of the learned manifold or graph may be greatly jeopardized by outliers, in ways that affect the stability of various manifold learning methods.

In recent years, several methods have been proposed to handle outliers in nonlinear data (Li et al., 2009; Zhigang Tang et al., 2010; Du et al., 2013). Despite the success of those methods, they only aim at finding the outliers instead of correcting them. In addition, few theoretical results characterize their statistical performances. In this paper, we propose a novel non task-driven algorithm for the mixed noise model in (4.1) and provide theoretical guarantees to control its estimation error. Specifically, we consider the mixed noise model as

$$\tilde{X}_i = X_i + S_i + \epsilon_i, \quad i = 1, \ldots, n, \tag{4.1}$$

where $X_i \in \mathbb{R}^p$ is the noiseless data independently drawn from some manifold $\mathcal{M}$ with an intrinsic dimension $d < p$, ϵ_i is the i.i.d. Gaussian noise with small magnitudes, and S_i is the sparse noise with possible large magnitudes. If S_i has a large entry, then the corresponding $\tilde{X}_i$ is usually considered as an outlier. The goal of this chapter is to simultaneously recover

X_i and S_i from $\tilde{X}_i$, $i = 1, .., n$.

There are several benefits in recovering the noise term S_i along with the signal X_i. First, the support of S_i indicates the locations of the anomaly, which is informative in many applications. For example, if X_i is the gene expression data from the ith patient, the nonzero elements in S_i indicate the differentially expressed genes that are the candidates for personalized medicine. Similarly, if S_i is a result of malfunctioned hardware, its nonzero elements indicate the locations of the malfunction parts. Secondly, the recovery of S_i allows the "outliers" to be pulled back to the data manifold instead of simply being discarded. This prevents waste of information and is especially beneficial in cases where data is insufficient. Thirdly, in some applications, the sparse S_i is part of the actual data rather than a noise term, then the algorithm provides a natural decomposition of the data into a sparse and a non-sparse component that may carry different pieces of information.

Along a similar line of research, robust principle component analysis (RPCA) (Candes et al., 2011) has received considerable attention and has demonstrated its success in separating data from sparse noise in many applications. However, its assumption that the data lies in a low dimensional subspace is somewhat strict. In this chapter, we generalize the Robust PCA idea to the non-linear manifold setting. The major new components in our algorithm are: 1) an incorporation of the curvature information of the manifold into the optimization framework, and 2) a unified way to apply RPCA to a collection of tangent spaces of the manifold.

4.2 Methodology

Let $\tilde{X} = [\tilde{X}_1, \ldots, \tilde{X}_n] \in \mathbb{R}^{p \times n}$ be the noisy data matrix containing n samples. Each sample is a vector in $\mathbb{R}^p$ independently drawn based on (4.1). The overall data matrix $\tilde{X}$ has the representation

$$\tilde{X} = X + S + N$$

where X is the clean data matrix, S is the matrix of the sparse noise, and N is the Gaussian noise. We further assume that the clean data X_i lies on some manifold $\mathcal{M}$ embedded in $\mathbb{R}^p$ with a small intrinsic dimension $d \ll p$ and the sample size is sufficient ($n \geq p$). The small intrinsic dimension assumption ensures that data is locally low dimensional so that the corresponding local data matrix is of low rank. This property allows the data to be separated from the sparse noise.

The key idea behind our method is to handle the data locally. We use the k Nearest Neighbors (kNN) to construct local data matrices, where k is larger than the intrinsic dimension d. For a data point $X_i \in \mathbb{R}^p$, we define the local patch centered at it to be the set consisted of its kNN and itself, and a local data matrix $X^{(i)}$ associated with this patch is $X^{(i)} = [X_{i_1}, X_{i_2}, \ldots, X_{i_k}, X_i]$, where X_{i_j} is the jth-nearest neighbor of X_i. Let $\mathcal{P}_i$ be the restriction operator to the ith patch, i.e., $\mathcal{P}_i(X) = XP_i$ where P_i is the $n \times (k+1)$ matrix that selects the columns of X in the ith patch. Then $X^{(i)} = \mathcal{P}_i(X)$. Similarly, we define $S^{(i)} = \mathcal{P}_i(S)$, $N^{(i)} = \mathcal{P}_i(N)$ and $\tilde{X}^{(i)} = \mathcal{P}_i(\tilde{X})$.

Since each local data matrix $X^{(i)}$ is of low rank and S is sparse, we can decompose the noisy data matrix into low-rank parts and sparse parts through solving the following optimization problem

$$\begin{aligned}(\hat{S}, \{\hat{L}^{(i)}\}_{i=1}^n) &= \arg\min_{S, L^{(i)}} F(S, \{L^{(i)}\}_{i=1}^n) \\ &\equiv \arg\min_{S, S^{(i)}, L^{(i)}} \sum_{i=1}^n \left(\lambda_i \|\tilde{X}^{(i)} - L^{(i)} - S^{(i)}\|_F^2 + \|\mathcal{C}(L^{(i)})\|_* + \beta \|S^{(i)}\|_1\right) \qquad (4.2) \\ &\quad \text{subject to } S^{(i)} = \mathcal{P}_i(S),\end{aligned}$$

here we take $\beta = \max\{k, p\}^{-1/2}$ as in RPCA, $\tilde{X}^{(i)} = \mathcal{P}_i(\tilde{X})$ is the local data matrix on the ith patch and $\mathcal{C}$ is the centering operator that subtract the column mean: $\mathcal{C}(Z) = Z(I - \frac{1}{k}11^T)$, where 1 is the $(k+1)$-dimensional column vector of all ones. Here we are decomposing the data on each patch into a low-rank part $L^{(i)}$ and a sparse part $S^{(i)}$ by imposing the nuclear norm and entry-wise ℓ_1 norm on $L^{(i)}$ and $S^{(i)}$, respectively. There are two key components in this formulation: First, the patches have overlapping components (for example, X_1 may

belong to several patches). Thus, the constraint $S^{(i)} = \mathcal{P}_i(S)$ is particularly important because it ensures the same point (and the sparse noise on that point) belonging to different patches eventually has all its copies coincide with each other. Secondly, we do not have such a requirement on $L^{(i)}$ because the $L^{(i)}$s correspond to local tangent spaces, which will be explained in the next section. Although some of the tangent spaces may be close, there is no reason for a point on the manifold to have the same projection onto two different tangent spaces. This seemingly subtle difference has a large impact on the final result.

If the data has no Gaussian noise, i.e., $N = 0$, then $\hat{X} \equiv \tilde{X} - \hat{S}$ is the final estimation for X. If $N \neq 0$, we can no longer only remove the sparse noise from $\tilde{X}$ and use $\tilde{X} - \hat{S}$ to approximate the clean data. Instead, we use the supposedly cleaner (See §3) tangent spaces $\hat{L}^{(i)}$ to construct a final estimate $\hat{X}$ of X via fitting it to $\hat{L}^{(i)}$

$$\hat{X} = \underset{Z \in \mathbb{R}^{p \times n}}{\arg\min} \sum_{i=1}^{n} \lambda_i \|\mathcal{P}_i(Z) - \hat{L}^{(i)}\|_F^2. \tag{4.3}$$

The following discussion revolves around (4.2) and (4.3), and the structure of the chapter is as follows. In §4.3, we explain the geometric meaning of each term in (4.2). The choice of λ requires the information of the curvature of the manifold. Optimization algorithm is presented is §4.4 and numerical experiments are in §4.5.

4.3 Geometric explanation

We provide a geometric intuition for the formulation (4.2). Let us write the local clean data matrix $X^{(i)}$ into its Taylor expansion along the manifold,

$$X^{(i)} = X_i 1^T + T^{(i)} + R^{(i)}, \tag{4.4}$$

where the Taylor series is expanded at X_i (the point around which the ith patch is constructed), $T^{(i)}$ stores the first order term whose columns lie in the tangent space of the manifold at X_i, and $R^{(i)}$ contains all the higher order terms. The sum of the first two terms $X_i 1^T + T^{(i)}$ is the linear approximation to $X^{(i)}$ that is unknown if the tangent space is not given. This linear approximation precisely corresponds to the $L^{(i)}$s in (4.2), i.e.,

$L^{(i)} = X_i 1^T + T^{(i)}$. Since the tangent space has the same dimensionality d as the manifold, with randomly chosen points, we have with probability one, that $\text{rank}(T^{(i)}) = d$. As a result, $\text{rank}(L^{(i)}) = \text{rank}(X_i 1^T + T^{(i)}) \leq d + 1$. By the assumption that $d < \min\{p, k\}$, we know that $L^{(i)}$ is indeed low rank.

Combing (4.4) with $\tilde{X}^{(i)} = X^{(i)} + S^{(i)} + N^{(i)}$, we find the misfit term $\tilde{X}^{(i)} - L^{(i)} - S^{(i)}$ in (4.2) equals $N^{(i)} + R^{(i)}$. This implies that the misfit contains the high order residue (i.e., the linear approximation error) and the Gaussian noise.

4.4 Optimization algorithm

To solve the convex optimization problem (4.2) in a memory-economic way, we first write $L^{(i)}$ as a function of S and eliminate them from the problem. We can do so by fixing S and minimizing the objective function with respect to $L^{(i)}$

$$\begin{aligned}\hat{L}^{(i)} &= \arg\min_{L^{(i)}} \lambda_i \|\tilde{X}^{(i)} - L^{(i)} - S^{(i)}\|_F^2 + \|\mathcal{C}(L^{(i)})\|_* \\ &= \arg\min_{L^{(i)}} \lambda_i \|\mathcal{C}(L^{(i)}) - \mathcal{C}(\tilde{X}^{(i)} - S^{(i)})\|_F^2 + \|\mathcal{C}(L^{(i)})\|_* \\ &\quad + \lambda_i \|(I - \mathcal{C})(L^{(i)} - (\tilde{X}^{(i)} - S^{(i)}))\|_F^2.\end{aligned} \tag{4.5}$$

Notice that $L^{(i)}$ can be decomposed as $L^{(i)} = \mathcal{C}(L^{(i)}) + (I - \mathcal{C})(L^{(i)})$, set $A = \mathcal{C}(L^{(i)}), B = (I - \mathcal{C})(L^{(i)})$, then (4.5) is equivalent to

$$(\hat{A}, \hat{B}) = \arg\min_{A,B} \ \lambda_i \|A - \mathcal{C}(\tilde{X}^{(i)} - S^{(i)})\|_F^2 + \|A\|_* + \lambda_i \|B - (I - \mathcal{C})(\tilde{X}^{*(i)} - S^{(i)}))\|_F^2,$$

which decouples into

$$\begin{aligned}\hat{A} &= \arg\min_{A} \ \lambda_i \|A - \mathcal{C}(\tilde{X}^{(i)} - S^{(i)})\|_F^2 + \|A\|_*, \\ \hat{B} &= \arg\min_{B} \lambda_i \|B - (I - \mathcal{C})(\tilde{X}^{(i)} - S^{(i)})\|_F^2.\end{aligned}$$

The problems above have closed form solutions

$$\hat{A} = \mathcal{T}_{1/2\lambda_i}(\mathcal{C}(\tilde{X}^{(i)} - \mathcal{P}_i(S))), \ \hat{B} = (I - \mathcal{C})(\tilde{X}^{(i)} - \mathcal{P}_i(S)), \tag{4.6}$$

where $\mathcal{T}_\mu$ is the soft-thresholding operator on the singular values

$$\mathcal{T}_\mu(Z) = U \max\{\Sigma - \mu I, 0\} V^*, \text{ where } U\Sigma V^* \text{ is the SVD of } Z.$$

Combing $\hat{A}$ and $\hat{B}$, we have derived the closed form solution for $\hat{L}^{(i)}$

$$\hat{L}^{(i)}(S) = \mathcal{T}_{1/2\lambda_i}(\mathcal{C}(\tilde{X}^{(i)} - \mathcal{P}_i(S))) + (I - \mathcal{C})(\tilde{X}^{(i)} - \mathcal{P}_i(S)). \tag{4.7}$$

Plugging (4.7) into F in (4.2), the resulting optimization problem solely depends on S. Then we apply FISTA (Beck and Teboulle, 2009c; Sha et al., 2019) to find the optimal solution $\hat{S}$ with

$$\hat{S} = \arg\min_S F(\hat{L}^{(i)}(S), S). \tag{4.8}$$

Once $\hat{S}$ is found, if the data has no Gaussian noise, then the final estimation for X is $\hat{X} \equiv \tilde{X} - \hat{S}$; if there is Gaussian noise, we use the following denoised local patches $\hat{L}^{(i)}_{\tau^*}$

$$\hat{L}^{(i)}_{\tau^*} = H_{\tau^*}(\mathcal{C}(\tilde{X}^{(i)} - \mathcal{P}_i(\hat{S}))) + (I - \mathcal{C})(\tilde{X}^{(i)} - \mathcal{P}_i(\hat{S})), \tag{4.9}$$

where H_{τ^*} is the singular value hard thresholding Operator with the optimal threshold as defined in (Gavish and Donoho, 2014). This optimal thresholding removes the Gaussian noise from $\hat{L}^{(i)}_{\tau^*}$. With the denoised $\hat{L}^{(i)}_{\tau^*}$, we solve (4.3) to obtain the denoised data

$$\hat{X} = \left(\sum_{i=1}^{n} \lambda_i \hat{L}^{(i)}_{\tau^*} P_i^T \right) (\sum_{i=1}^{n} \lambda_i P_i P_i^T)^{-1}. \tag{4.10}$$

The proposed nonlinear robust principle component analysis (NRPCA) algorithm is summarized in Algorithm 4.1.

There is one caveat in solving (4.2): the strong sparse noise may result in a wrong neighborhood assignment when constructing the local patches. Therefore, once $\hat{S}$ is obtained and removed from the data, we update the neighborhood assignment and re-compute $\hat{S}$. This procedure is repeated T times.

4.5 Numerical experiments

We evaluate the performance of the proposed algorithm on simulated and real-world data sets.

Algorithm 4.1: Nonlinear RPCA

Input: Noisy data matrix $\tilde{X}$, k (number of neighbors in each local patch), T (number of neighborhood updates iterations)

Output: the denoised data $\hat{X}$, the estimated sparse noise $\hat{S}$

Estimate the curvature;
Estimate λ_i, $i = 1, \ldots, n$ as in §5, set β as in (4.2);
$\hat{S} \leftarrow 0$;
for *iter = 1: T* **do**
    Find the kNN for each point using $\tilde{X} - \hat{S}$ and construct the restriction operators $\{\mathcal{P}_i\}_{i=1}^n$;
    Construct the local data matrices $\tilde{X}^{(i)} = \mathcal{P}_i(\tilde{X})$ using $\mathcal{P}_i$ and the noisy data $\tilde{X}$;
    $\hat{S} \leftarrow$ minimizer of (4.8) iteratively using FISTA;
end
Compute each $\hat{L}_{\tau^*}^{(i)}$ from (4.9) and assign $\hat{X}$ from (4.10).

Simulated Swiss roll: We demonstrate the superior performance of NRPCA on a synthetically generated dataset following the mixed noise model (4.1). We sampled 2000 noiseless data X_i uniformly from a 3D Swiss roll and generated the Gaussian noise matrix with i.i.d. entries obey $\mathcal{N}(0, 0.25)$. The sparse noise matrix S is generated by randomly replacing 100 entries of a zero $p \times n$ matrix with i.i.d. samples generated from $(-1)^y \cdot z$ where $y \sim$ Bernoulli(0.5) and $z \sim \mathcal{N}(2, 0.09)$. We applied NRPCA to the simulated data with patch size $k = 16$. Figure 4.1 reports the denoising effect in the original space (3D) looking down from above. We observed a visible reduction of the noise. A similar experiment on the high dimensional Swiss roll is in the appendix, where the differences between $\tilde{X} - \hat{S}$ and $\hat{X}$ are much more apparent.

MNIST: We observe some interesting dimension reduction results of the MNIST dataset with the help of NRPCA. It is well-known that the handwritten digits 4 and 9 have so high a similarity that the popular dimension reduction methods Isomap and Laplacian Eigenmaps are not able to separate them into two clusters (first column of Figure 4.2). We conjecture that the overlapping parts are caused by personalized writing styles with different beginning or finishing strokes. This type of differences can be better modelled by sparse noise than Gaussian or Poisson noises. The right column of Figure 4.2 confirms this conjecture: after

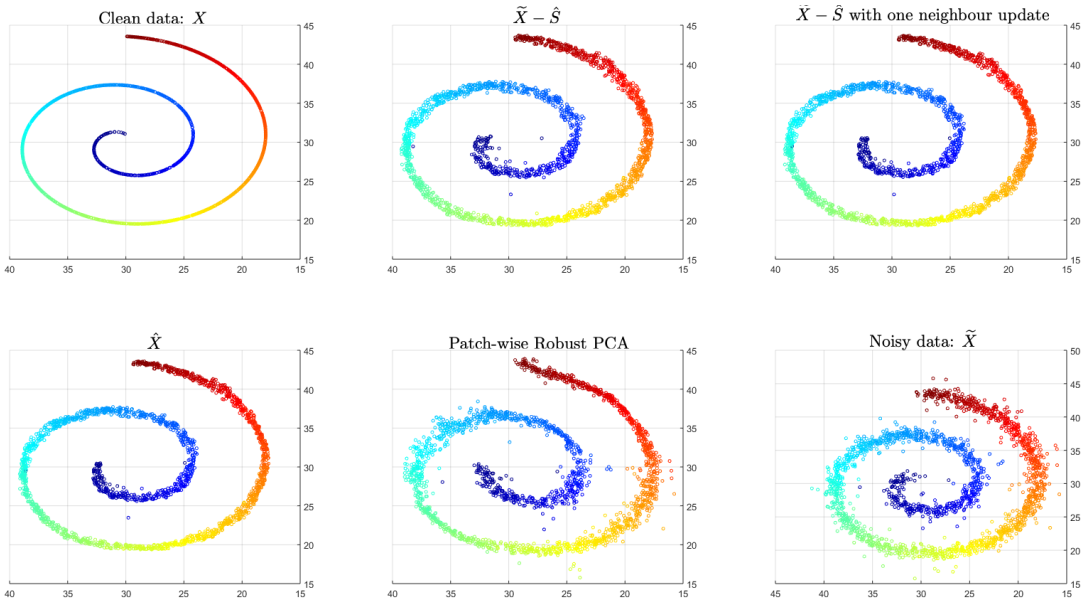


Figure 4.1: NRPCA applied to the noisy Swiss roll data set. $\tilde{X} - \hat{S}$ is the result after subtracting the estimated sparse noise via NRPCA with $T = 1$; "$\tilde{X} - \hat{S}$ with one neighbor update" is that with $T = 2$, i.e., patches are reassigned once; $\hat{X}$ is the denoised data obtained via NRPCA with $T = 2$; "Patch-wise Robust PCA" refers to the ad-hoc application of the vanilla RPCA to each local patch independently, whose performance is clearly worse than the proposed joint-recovery formulation.

the NRPCA denoising (with $k = 16$), we see a much better separability of the two digits using the first two coordinates of Isomap and Laplacian Eigenmaps. In addition, these new embedding results seem to suggest that some trajectory patterns may exist in the data. We provide additional plots in the appendix to support this observation.

4.6 Conclusion

In this chapter, we proposed the first outlier correction method for nonlinear data analysis that can correct outliers caused by the addition of large sparse noise. The method is a generalization of the Robust PCA method to the nonlinear setting. We provided procedures to treat the non-linearity by working with overlapping local patches of the data manifold and incorporating the curvature information into the denoising algorithm. We demonstrated that the method works equally well when Gaussian noises are present in the data in addition

to the sparse noise. We established a theoretical error bound on the denoised data that holds under conditions only depending on the intrinsic properties of the manifold. We tested our method on both synthetic and real dataset that were known to have nonlinear structures and reported promising results.

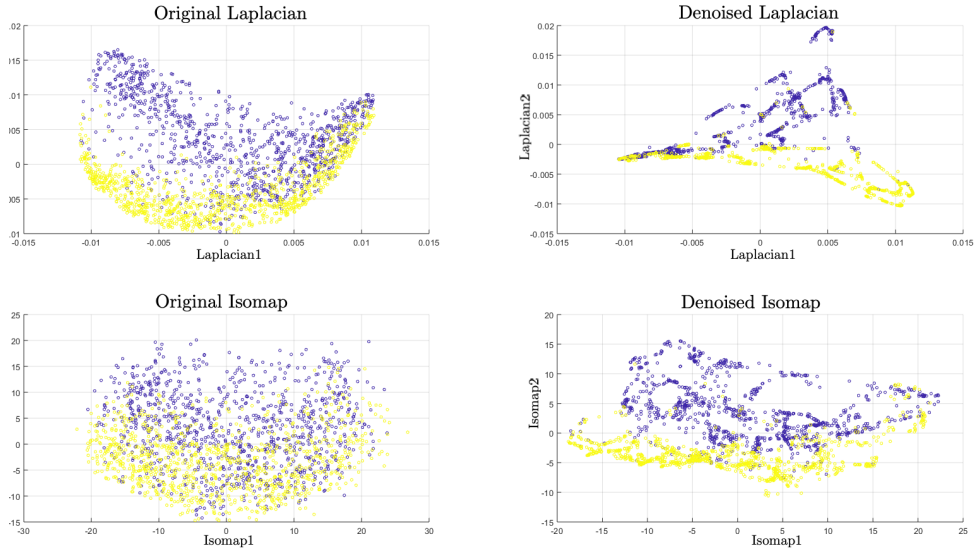


Figure 4.2: Laplacian eigenmaps and Isomap results for the original and the NRPCA denoised digits 4 and 9 from the MNIST dataset.

CHAPTER 5

ONLINE MATRIX COMPLETION WITH QUATERNION MATRIX

5.1 Introduction

Matrix completion problems aim to recover an unknown matrix given that the matrix has a low rank. It has been widely studied and has many applications especially in computer vision (Cabral et al., 2011, 2014), video processing (Ji et al., 2010; Kim et al., 2015), bioinformatics (Li et al., 2017; Lu et al., 2018), etc. There are both convex and non-convex algorithms for solving it. Convex algorithms tend to simplify the problem and use convex penalties (Recht et al., 2010). While it is easier to find an optimal solution, it may cost a lot of computational time. Nonconvex algorithms tend to use matrix factorization. Even if they need less computational time, they may not easily converge to the global optimal solution because of the saddle points and local optimal solution. In this case, a good initialization is very important.

Traditional methods for matrix completion have already achieved good performance when dealing with greyscale images. For each image, we can easily view it as a matrix. As for a video, we can convert each frame as a vector in a matrix. However, the problem arises when applied to color images. Color images have three channels (Red, Green, and Blue) that have a mutual connection. Intuitively, traditional matrix-based methods that treat each channel separately do not work well.

Some tensor-based methods have been developed to solve this multidimensional data problem. In this case, it is converted as a low-rank tensor approximation problem where three channels are considered as a third-order tensor. There are many well-known models like ANDECOMP/PARAFAC (CP) and Tucker (Zhou et al., 2019; Rauhut et al., 2017). Similar to matrix SVD, they describe the tensor as the sum of the outer products of vectors. In a recent paper (Kilmer and Martin, 2011), it proposed t-SVD, which expresses the tensor

as the sum of outer products of matrices.

In this chapter, we consider an alternating way and apply quaternion matrices to this problem. A quaternion matrix has four parts: one real part and three imaginary parts representing the three channels. Quaternion matrices have been applied to other models such as Deep Neural Network (Liu et al., 2019; Zhu et al., 2018; Gaudet and Maida, 2018). They also have been applied on many areas like Natural Language Processing (Parcollet et al., 2018; Tay et al., 2019) and image processing (Wang et al., 2018; Ye et al., 2020).

Also, there is an increasing trend today that we need to deal with large-scale data. The traditional offline matrix completion tends to give good performance. Offline matrix completion model needs to collect all observation data and gets recovery result at once. However, it is not practical on some applications, such as web data analysis. For example, in Figure 5.1, a movie ranking system can be seen as a matrix. Each column is a movie that needs to be rated, and each column corresponds to a user. We need to offer up-to-date recommendations to users. Moreover, the estimate should be better if we continuously get observations.

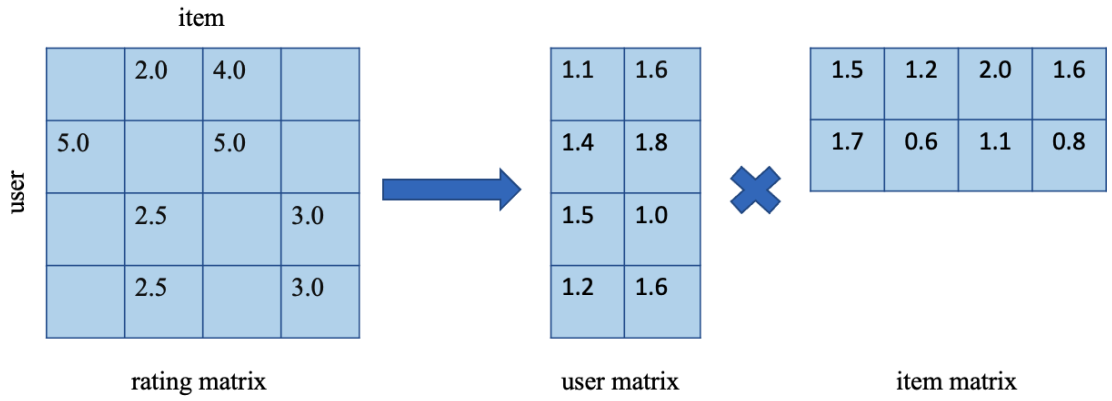


Figure 5.1: A movie rating system. For a given $d_1 \times d_2$ low-rank matrix with missing entries, it can be factorized by a $d_1 \times k$ user matrix and a $k \times d_2$ item matrix where k is the rank of the original matrix.

This chapter combines quaternion matrices and online matrix completion. We will develop an online low-rank quaternion matrix completion model that can be widely used in many cases.

5.2 Introduction on Quaternion Matrices

We introduce quaternion numbers/matrices and their properties in this section.

5.2.1 Quaternion Numbers

A quaternion number $\mathbf{q} \in \mathbb{Q}^n$ is defined as

$$\mathbf{q} = q_r + q_i\mathbf{i} + q_j\mathbf{j} + q_k\mathbf{k}, \tag{5.1}$$

where $q_r, q_i, q_j, q_k \in \mathbb{R}$ and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are three imaginary units satisfying

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1, \mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \mathbf{ki} = -\mathbf{ik} = \mathbf{j}. \tag{5.2}$$

The conjugate and modulus of $\mathbf{q}$ are respectively defined by

$$\mathbf{q}^* = q_r - q_i\mathbf{i} - q_j\mathbf{j} - q_k\mathbf{k} \text{ and } |\mathbf{q}| = \sqrt{q_r^2 + q_i^2 + q_j^2 + q_k^2}. \tag{5.3}$$

Let $\mathbf{x} = [\mathbf{x}_i] \in \mathbb{Q}^n$ be a quaternion vector. We define the following three norms for the quaternion vector $\mathbf{x}$: 1-norm $\|\mathbf{x}\|_1 := \sum_{i=1}^n |\mathbf{x}_i|$, 2-norm $\|\mathbf{x}\|_2 := \sqrt{\sum_{i=1}^n |\mathbf{x}_i|^2}$, and ∞-norm $\|\mathbf{x}\|_\infty := \max_{1\le i\le n} |\mathbf{x}_i|$.

Let $\mathbf{A} = [\mathbf{a}_{ij}] \in \mathbb{Q}^{n_1\times n_2}$ be a quaternion matrix. We define the following norms: 1-norm $\|\mathbf{A}\|_1 = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} |\mathbf{a}_{ij}|$, Forbinoes norm $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} |\mathbf{a}_{ij}|^2} = \sqrt{\mathrm{Tr}(\mathbf{A}^*\mathbf{A})}$, where $\mathbf{A}^*$ is the conjugate transpose (or Hermitian transpose) of $\mathbf{A}$, and ∞-norm $\|\mathbf{A}\|_\infty = \max_{1\le i\le n_1, 1\le j\le n_2} |\mathbf{a}_{ij}|$. The rank of matrix $\mathbf{A}$ is the number of independent rows/columns in $\mathbf{A}$. A square quaternion matrix is unitary if $\mathbf{A}^*\mathbf{A} = \mathbf{A}\mathbf{A}^* = \mathbf{I}$. An Hermitian quaternion matrix satisfies $\mathbf{A}^* = \mathbf{A}$, which is an extension of symmetric matrices.

A color image with R (red), G (green), B (blue) channels can be represented by a quaternion matrix without the real part. That is,

$$\mathbf{A}_{ij} = R_{ij}\mathbf{i} + G_{ij}\mathbf{j} + B_{ij}\mathbf{k}, \tag{5.4}$$

where R_{ij}, G_{ij}, and B_{ij} are the corresponding R, G, and B channels.

5.2.2 Basic Properties

Some properties of quaternion matrices are the same as real matrices. For example,

- The definition of the inner product $\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{Tr}\langle \mathbf{A}^* \mathbf{B} \rangle$;
- Spectral norm and the Frobenius norm $\|\mathbf{A}^*\| = \|\mathbf{A}\|$, $\|\mathbf{A}^*\|_F = \|\mathbf{A}\|_F$;
- $\|\mathbf{AB}\|_F \leq \|\mathbf{A}\|\|\mathbf{B}\|_F$;
- $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$;
- $\|\mathbf{A}\| \leq \|\mathbf{A}\|_F$.
- Cauchy-Schwarz inequality $\mathrm{Re}(\mathrm{tr}(\mathbf{A}^*\mathbf{B})) \leq \|\mathbf{A}\|_F \|\mathbf{B}\|_F$.

However, we need to be careful about one thing. For real matrices, the matrices in a trace of a product can be switched without changing the result. However for quarternion matrices $\mathbf{A}$ and $\mathbf{B}$, we can only have

$$\mathrm{Re}(\mathrm{tr}(\mathbf{AB})) = \mathrm{Re}(\mathrm{tr}(\mathbf{BA})). \tag{5.5}$$

For example, given two quaternion matrices

$$\mathbf{A} = \begin{bmatrix} 1,1 \\ 1,0 \end{bmatrix} + \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix} \mathbf{i} + \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix} \mathbf{j} + \begin{bmatrix} 1,1 \\ 1,0 \end{bmatrix} \mathbf{k},$$

and

$$\mathbf{B} = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix} + \begin{bmatrix} 1,1 \\ 1,0 \end{bmatrix} \mathbf{i} + \begin{bmatrix} 1,1 \\ 1,0 \end{bmatrix} \mathbf{j} + \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix} \mathbf{k},$$

we have

$$\mathrm{tr}(\mathbf{AB}) = -10 + 20\mathbf{i} + 6\mathbf{j} + 10\mathbf{k},$$

and

$$\mathrm{tr}(\mathbf{BA}) = -10 + 6\mathbf{i} + 20\mathbf{j} + 10\mathbf{k}.$$

In this case, $\mathrm{tr}(\mathbf{AB}) \neq \mathrm{tr}(\mathbf{BA})$. The reason is that the product of two quaternion numbers may not be the same if the order is changed. For example, let $\mathbf{a} = 1 + 1\mathbf{i} + 2\mathbf{j} + 3\mathbf{k}$ and $\mathbf{b} = 3 + 2\mathbf{i} + 2\mathbf{j} + 1\mathbf{k}$, then we have

$$\mathbf{a} * \mathbf{b} = -6 + 1\mathbf{i} + 13\mathbf{j} + 8\mathbf{k}, \ \mathbf{b} * \mathbf{a} = -6 + 9\mathbf{i} + 3\mathbf{j} + 12\mathbf{k}.$$

This difference may cause some difficulties for the theoretical proof, which will be explained in the following sections.

5.2.3 Singular Value Decomposition

According to the work by Zhang (1997), we can define singular value decomposition for quaternion matrices. For any quaternion matrix $\mathbf{L} \in \mathbb{Q}^{d_1 \times d_2}$ with rank k, there exists an unitary quaternion matrix $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_{d_1}] \in \mathbb{Q}^{d_1 \times d_1}$ and another unitary quaternion matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_{d_2}] \in \mathbb{Q}^{d_2 \times d_2}$ such that

$$\mathbf{L} = \mathbf{U}\Sigma_k\mathbf{V}^* \tag{5.6}$$

where $\Sigma_k \in \mathbb{R}^{d_1 \times d_2}$ consists of all singular values of $\mathbf{L}$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$, on its diagonal entries. Then the spectral norm $\|\mathbf{L}\| := \max\{\sigma_1, \ldots, \sigma_k\}$. The condition number κ is defined as $\kappa = \frac{\max\{\sigma_1, \ldots, \sigma_k\}}{\min\{\sigma_1, \ldots, \sigma_k\}}$.

What's more, for any Hermitian quaternion matrix $\mathbf{L} \in \mathbb{Q}^{d \times d}$ with rank k, there exists an unitary quaternion matrix $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_d]$, with $\mathbf{u}_i$ being its i-th column, such that

$$\mathbf{L} = \mathbf{U}\Sigma_k\mathbf{U}^* \tag{5.7}$$

where $\Sigma_k = \mathrm{diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0) \in \mathbb{R}^{d \times d}$ consists of all singular values of $\mathbf{L}$, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$.

5.2.4 Incoherence Condition

Let $\mathbf{W} \in \mathbb{Q}^{d\times k}$ be an orthonormal basis of a subspace of $\mathbb{R}^d$ of dimension k, then the projection to the subspace is $\mathcal{P}_{\mathbf{W}} = \mathbf{W}\mathbf{W}^*$. We define the coherence of $\mathbf{W}$ as

$$\mu(\mathbf{W}) = \frac{d}{k}\max_{1\le i\le d}\|\mathcal{P}_{\mathbf{W}}\mathbf{e}_i\|^2 = \frac{d}{k}\max_{1\le i\le d}\|\mathbf{e}_i^*\mathbf{W}\|^2, \tag{5.8}$$

where $\mathbf{e}_i$ is the vector with the i-th component being 1 and others being 0.

Definition 5.2.1. We assume $\mathbf{M}$ is μ-incoherent, i.e.,

$$\max_i \|\mathbf{X}^*\mathbf{e}_i\|^2 \le \frac{\mu k}{d_1},\ \ \max_i \|\mathbf{Y}^*\mathbf{e}_i\|^2 \le \frac{\mu k}{d_2} \tag{5.9}$$

and

$$\|\mathbf{X}\mathbf{Y}^*\|_\infty \le \sqrt{\frac{\mu k}{d_1 d_2}}, \tag{5.10}$$

where $\mathbf{X} \in \mathbb{Q}^{d_1\times k}$, $\mathbf{Y} \in \mathbb{Q}^{d_2\times k}$ are the left and right singular vectors of $\mathbf{M}$.

5.2.5 Sampling Scheme

We consider the Bernoulli model for uniform sampling. Let $\Omega \subset [d_1]\times[d_2]$. Given a matrix $\mathbf{M}$, we define the matrix $\mathbb{P}_\Omega$ as

$$[\mathbb{P}_\Omega(\mathbf{M})]_{ij} = \begin{cases} \mathbf{M}_{ij} & \text{if } (i,j)\in\Omega, \\ 0 & \text{if } (i,j)\notin\Omega. \end{cases}$$

Every time (i,j) is uniformly sampled from Ω. Our goal is to recover $\mathbf{M}$ given $\mathbb{P}_\Omega(\mathbf{M})$.

5.3 Online Matrix Completion Algorithms and its Theoretical Analysis

We first consider a Hermitian quaternion matrix $\mathbf{M} \in \mathbb{Q}^{d\times d}$, i.e., there exists a quaternion matrix $\mathbf{U}$ such that $\mathbf{M} = \mathbf{U}\mathbf{U}^*$. Also, we write $\mathbf{U}$ with its columns as

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d].$$

Moreover, we define $f(\mathbf{U})$ as

$$\begin{aligned} f(\mathbf{U}) &= \|\mathbf{M} - \mathbf{U}\mathbf{U}^*\|_F^2 \\ &= \langle \mathbf{M}, -\mathbf{U}\mathbf{U}^* \rangle + \langle \mathbf{M}, \mathbf{M} \rangle + \langle -\mathbf{U}\mathbf{U}^*, \mathbf{M} \rangle + \langle \mathbf{U}\mathbf{U}^*, \mathbf{U}\mathbf{U}^* \rangle. \end{aligned}$$

The stochastic gradient of $f(\mathbf{U})$ given the (i, j) component is

$$SG(\mathbf{U}) = 2d^2[(\mathbf{U}\mathbf{U}^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^* + (\mathbf{U}\mathbf{U}^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*]\mathbf{U}. \tag{5.11}$$

Note that $(\mathbf{U}\mathbf{U}^* - \mathbf{M})_{ij}^* = (\mathbf{U}\mathbf{U}^* - \mathbf{M})_{ji}$ because both $\mathbf{M}$ and $\mathbf{U}\mathbf{U}^*$ are Hermitian. The expectation of $SG(\mathbf{U})$ is

$$\begin{aligned} \mathbb{E}SG(\mathbf{U}) &= \sum_{i=1,j=1}^{d,d} \frac{1}{d^2} 2d^2((\mathbf{U}\mathbf{U}^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^* + (\mathbf{U}\mathbf{U}^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*)\mathbf{U} \\ &= 4(\mathbf{U}\mathbf{U}^* - \mathbf{M})\mathbf{U}. \end{aligned}$$

Let $\nabla f(\mathbf{U}) := 4(\mathbf{U}\mathbf{U}^* - \mathbf{M})\mathbf{U}$, which is one descent direction of $f(\mathbf{U})$.

We assume $\|\mathbf{M}\| = 1$. Then $\kappa = \frac{1}{\sigma_{\min}(\mathbf{M})}$, namely, $\sigma_{\min}(\mathbf{M}) = \frac{1}{\kappa}$. We also denote

$$\mathrm{SVD}(\mathbf{M}) = \mathbf{X}\mathbf{S}\mathbf{X}^*, \quad \mathrm{SVD}(\mathbf{U}\mathbf{U}^*) = \mathbf{W}\mathbf{D}\mathbf{W}^*.$$

Here $\mathbf{X} \in \mathbb{Q}^{d\times k}$ with k orthogonal columns and $\mathbf{S} \in \mathbb{R}^{k\times k}$ is a diagonal square matrix.

First, we prepare with a few lemmas about properties of $f(\mathbf{U})$ in a local Frobenious ball around optimal.

Lemma 5.3.1. *Within the region $\mathcal{D} = \{\mathbf{U} | \|\mathbf{U}\| \le \Gamma\}$, we have the function $f(\mathbf{U})$ satisfying for any $\mathbf{U}_1, \mathbf{U}_2 \in \mathcal{D}$:*

$$\|\nabla f(\mathbf{U}_1) - \nabla f(\mathbf{U}_2)\|_F \le \beta \|\mathbf{U}_1 - \mathbf{U}_2\|_F, \tag{5.12}$$

with $\beta = 16 \max\{\|\mathbf{U}_1\|^2, \|\mathbf{U}_2\|^2, 1\}$.

Proof. From the definition of $\nabla f(\mathbf{U})$, we have

$$
\begin{aligned}
&\|\nabla f(\mathbf{U}_1) - \nabla f(\mathbf{U}_2)\|_F \\
=&\|4(\mathbf{U}_1\mathbf{U}_1^* - \mathbf{M})\mathbf{U}_1 - 4(\mathbf{U}_2\mathbf{U}_2^* - \mathbf{M})\mathbf{U}_2\|_F \\
\leq&4\|\mathbf{U}_1\mathbf{U}_1^*\mathbf{U}_1 - \mathbf{U}_2\mathbf{U}_2^*\mathbf{U}_2\|_F + 4\|\mathbf{M}(\mathbf{U}_1 - \mathbf{U}_2)\|_F \\
=&4\|\mathbf{U}_1\mathbf{U}_1^*(\mathbf{U}_1 - \mathbf{U}_2) + \mathbf{U}_1(\mathbf{U}_1 - \mathbf{U}_2)^*\mathbf{U}_2 + (\mathbf{U}_1 - \mathbf{U}_2)\mathbf{U}_2^*\mathbf{U}_2\|_F + 4\|\mathbf{M}(\mathbf{U}_1 - \mathbf{U}_2)\|_F \\
\leq&4\|\mathbf{U}_1\|^2\|\mathbf{U}_1 - \mathbf{U}_2\|_F + 4\|\mathbf{U}_1\|\|\mathbf{U}_2\|\|\mathbf{U}_1 - \mathbf{U}_2\|_F \\
&+ 4\|\mathbf{U}_2\|^2\|\mathbf{U}_1 - \mathbf{U}_2\|_F + 4\|\mathbf{M}\|\|\mathbf{U}_1 - \mathbf{U}_2\|_F \\
\leq&12\max\{\|\mathbf{U}_1\|^2, \|\mathbf{U}_2\|^2\}\|\mathbf{U}_1 - \mathbf{U}_2\|_F + 4\|\mathbf{M}\|\|\mathbf{U}_1 - \mathbf{U}_2\|_F \\
\leq&16\max\{\|\mathbf{U}_1\|^2, \|\mathbf{U}_2\|^2, 1\}\|\mathbf{U}_1 - \mathbf{U}_2\|_F.
\end{aligned}
$$

We have completed the proof. □

Lemma 5.3.2. *Within the region* $\mathcal{D} = \{\mathbf{U}|\sigma_{min}(\mathbf{X}^*\mathbf{U}) \geq \gamma\}$*, the function* $f(\mathbf{U}) = \|\mathbf{M} - \mathbf{U}\mathbf{U}^*\|_F^2$ *satisfies*

$$\|\nabla f(\mathbf{U})\|_F^2 \geq 4\gamma^2 f(\mathbf{U}). \tag{5.13}$$

Proof. Inside the region $\mathcal{D}$, we let $\mathrm{SVD}(\mathbf{U}\mathbf{U}^*) = \mathbf{W}\mathbf{D}\mathbf{W}^*$, thus we have

$$
\begin{aligned}
&\|\nabla f(\mathbf{U})\|_F^2 \\
=&16\|(\mathbf{U}\mathbf{U}^* - \mathbf{M})\mathbf{U}\|_F^2 \\
=&16(\|\mathcal{P}_{\mathbf{W}}(\mathbf{U}\mathbf{U}^* - \mathbf{M})\mathbf{U}\|_F^2 + \|\mathcal{P}_{\mathbf{W}_\perp}(\mathbf{U}\mathbf{U}^* - \mathbf{M})\mathbf{U}\|_F^2) \\
=&16\left(\mathrm{tr}(\mathcal{P}_{\mathbf{W}}(\mathbf{U}\mathbf{U}^* - \mathbf{M})\mathbf{U}\mathbf{U}^*(\mathbf{U}\mathbf{U}^* - \mathbf{M})\mathcal{P}_{\mathbf{W}}) + \|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathbf{U}\|_F^2\right) \\
=&16\left(\mathrm{tr}(\mathcal{P}_{\mathbf{W}}(\mathbf{U}\mathbf{U}^* - \mathbf{M})\mathbf{W}\mathbf{D}\mathbf{W}^*(\mathbf{U}\mathbf{U}^* - \mathbf{M})\mathcal{P}_{\mathbf{W}}) + \|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathbf{U}\|_F^2\right) \\
\geq&16(\sigma_{\min(\mathbf{D})}\|\mathcal{P}_{\mathbf{W}}(\mathbf{U}\mathbf{U}^* - \mathbf{M})\mathcal{P}_{\mathbf{W}}\|_F^2 + \|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathbf{U}\|_F^2) \\
=&16(\sigma_{\min(\mathbf{D})}\|\mathbf{U}\mathbf{U}^* - \mathcal{P}_{\mathbf{W}}\mathbf{M}\mathcal{P}_{\mathbf{W}}\|_F^2 + \|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathbf{U}\|_F^2).
\end{aligned}
$$

The inequality holds because

$$\begin{aligned}
&\mathrm{tr}(\mathcal{P}_{\mathbf{W}}(\mathbf{U}\mathbf{U}^*-\mathbf{M})\mathbf{W}\mathbf{D}\mathbf{W}^*(\mathbf{U}\mathbf{U}^*-\mathbf{M})\mathcal{P}_{\mathbf{W}})\\
\geq&\sigma_{\min(\mathbf{D})}\mathrm{tr}(\mathcal{P}_{\mathbf{W}}(\mathbf{U}\mathbf{U}^*-\mathbf{M})\mathbf{W}\mathbf{W}^*(\mathbf{U}\mathbf{U}^*-\mathbf{M})\mathcal{P}_{\mathbf{W}})\\
=&\sigma_{\min(\mathbf{D})}\mathrm{tr}(\mathcal{P}_{\mathbf{W}}(\mathbf{U}\mathbf{U}^*-\mathbf{M})\mathbf{W}\mathbf{W}^*\mathbf{W}\mathbf{W}^*(\mathbf{U}\mathbf{U}^*-\mathbf{M})\mathcal{P}_{\mathbf{W}})
\end{aligned}$$

The last equality is true because $\mathcal{P}_{\mathbf{W}}\mathbf{U}\mathbf{U}^*\mathcal{P}_{\mathbf{W}} = \mathbf{W}\mathbf{W}^*\mathbf{W}\mathbf{D}\mathbf{W}^*\mathbf{W}\mathbf{W}^* = \mathbf{W}\mathbf{D}\mathbf{W}^* = \mathbf{U}\mathbf{U}^*$.

On the other hand, we have

$$\begin{aligned}
&\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathbf{U}\|_F^2 = \|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{X}\mathbf{S}\mathbf{X}^*\mathbf{U}\|_F^2 \geq \sigma_{\min}^2(\mathbf{X}^*\mathbf{U})\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{X}\mathbf{S}\|_F^2\\
=&\sigma_{\min}^2(\mathbf{X}^*\mathbf{U})\mathrm{tr}(\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}^2\mathcal{P}_{\mathbf{W}_\perp}) = \sigma_{\min}^2(\mathbf{X}^*\mathbf{U})\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\|_F^2,
\end{aligned}$$

and

$$\begin{aligned}
&\sigma_{\min}(\mathbf{D}) = \lambda_{\min}(\mathbf{U}\mathbf{U}^*) = \lambda_{\min}(\mathbf{U}^*\mathbf{U})\\
\geq&\lambda_{\min}(\mathbf{U}^*\mathcal{P}_{\mathbf{X}}\mathbf{U}) = \sigma_{\min}^2(\mathbf{X}^*\mathbf{U}).
\end{aligned}$$

Finally, we can get

$$\begin{aligned}
&\|\nabla f(\mathbf{U})\|_F^2\\
\geq&16(\sigma_{\min}(\mathbf{D})\|\mathbf{U}\mathbf{U}^*-\mathcal{P}_{\mathbf{W}}\mathbf{M}\mathcal{P}_{\mathbf{W}}\|_F^2+\sigma_{\min}(\mathbf{X}^*\mathbf{U})\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\|_F^2)\\
\geq&16\sigma_{\min}^2(\mathbf{X}^*\mathbf{U})(\|\mathbf{U}\mathbf{U}^*-\mathcal{P}_{\mathbf{W}}\mathbf{M}\mathcal{P}_{\mathbf{W}}\|_F^2+\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\|_F^2))\\
=&16\sigma_{\min}^2(\mathbf{X}^*\mathbf{U})(\|\mathbf{U}\mathbf{U}^*-\mathcal{P}_{\mathbf{W}}\mathbf{M}\mathcal{P}_{\mathbf{W}}\|_F^2+\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}(\mathcal{P}_{\mathbf{W}}+\mathcal{P}_{\mathbf{W}_\perp})\|_F^2))\\
=&16\sigma_{\min}^2(\mathbf{X}^*\mathbf{U})(\|\mathbf{U}\mathbf{U}^*-\mathcal{P}_{\mathbf{W}}\mathbf{M}\mathcal{P}_{\mathbf{W}}\|_F^2+\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathcal{P}_{\mathbf{W}}\|_F^2+\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathcal{P}_{\mathbf{W}_\perp}\|_F^2)\\
\geq&4\sigma_{\min}^2(\mathbf{X}^*\mathbf{U})(\|\mathbf{U}\mathbf{U}^*-\mathcal{P}_{\mathbf{W}}\mathbf{M}\mathcal{P}_{\mathbf{W}}\|_F^2+\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathcal{P}_{\mathbf{W}}\|_F^2+\|\mathcal{P}_{\mathbf{W}}\mathbf{M}\mathcal{P}_{\mathbf{W}_\perp}\|_F^2\\
&+\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathcal{P}_{\mathbf{W}_\perp}\|_F^2)\\
=&4\sigma_{\min}^2(\mathbf{X}^*\mathbf{U})\|\mathbf{U}\mathbf{U}^*-\mathbf{M}\|_F^2\\
\geq&4\gamma^2\|\mathbf{U}\mathbf{U}^*-\mathbf{M}\|_F^2.
\end{aligned}$$

The third inequality holds because we have

$$\|\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathcal{P}_{\mathbf{W}}\|_F = \|\mathcal{P}_{\mathbf{W}}\mathbf{M}\mathcal{P}_{\mathbf{W}_\perp}\|_F. \tag{5.14}$$

The third equality holds because the inner product between each pair of $\mathbf{U}\mathbf{U}^* - \mathcal{P}_{\mathbf{W}}\mathbf{M}\mathcal{P}_{\mathbf{W}}$, $\mathcal{P}_{\mathbf{W}}\mathbf{M}\mathcal{P}_{\mathbf{W}_\perp}$, $\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathcal{P}_{\mathbf{W}}$ and $\mathcal{P}_{\mathbf{W}_\perp}\mathbf{M}\mathcal{P}_{\mathbf{W}_\perp}$ is 0. □

Lemma 5.3.3. *Within the region* $\mathcal{D} = \{\mathbf{U} | \|\mathbf{M} - \mathbf{U}\mathbf{U}^*\|_F \leq \frac{1}{10}\sigma_k(\mathbf{M})\}$*, we have*

$$\|\mathbf{U}\| \leq \sqrt{2\|\mathbf{M}\|}, \qquad \sigma_{\min}(\mathbf{X}^*\mathbf{U}) \geq \sqrt{\sigma_k(\mathbf{M})/2}. \tag{5.15}$$

Proof. From the definition of the spectral norm, we have

$$\begin{aligned}\|\mathbf{U}\|^2 =& \|\mathbf{U}\mathbf{U}^*\| \leq \|\mathbf{M}\| + \|\mathbf{M} - \mathbf{U}\mathbf{U}^*\| \\ \leq& \|\mathbf{M}\| + \|\mathbf{M} - \mathbf{U}\mathbf{U}^*\|_F \leq 2\|\mathbf{M}\|.\end{aligned}$$

We have the following lower bound for the smallest nonzero singular value of $\mathbf{U}^*\mathbf{U}$,

$$\begin{aligned}\sigma_{\min}(\mathbf{U}^*\mathbf{U}) =& \sigma_k(\mathbf{U}\mathbf{U}^*) = \sigma_k(\mathbf{M} - (\mathbf{M} - \mathbf{U}\mathbf{U}^*)) \\ \geq& \sigma_k(\mathbf{M}) - \|\mathbf{M} - \mathbf{U}\mathbf{U}^*\| \geq \frac{9}{10}\sigma_k(\mathbf{M}).\end{aligned}$$

The first inequality holds because $\forall i, j \in \mathbb{N}$, we have

$$\sigma_i(\mathbf{A}) \geq \sigma_{i+j-1}(\mathbf{A} + \mathbf{B}) - \sigma_j(\mathbf{B}).$$

On the other hand, we denote the orthogonal complementary space of $\mathbf{X}$ as $\mathbf{X}_\perp$. Then we have

$$\begin{aligned}\frac{9}{10}\sigma_k(\mathbf{M})\|\mathbf{X}_\perp^*\mathbf{W}\|^2 \leq& \frac{9}{10}\sigma_{\min}(\mathbf{U}^*\mathbf{U})\|\mathbf{X}_\perp^*\mathbf{W}\|^2 \leq \sigma_{\min}(\mathbf{D})\|\mathbf{X}_\perp^*\mathbf{W}\|^2 \\ \leq& \|\mathbf{X}_\perp^*\mathbf{W}\mathbf{D}\mathbf{W}^*\mathbf{X}_\perp\| \leq \|\mathbf{X}_\perp^*\mathbf{U}\mathbf{U}^*\mathbf{X}_\perp\|_F \\ =& \|\mathcal{P}_{\mathbf{X}_\perp}(\mathbf{M} - \mathbf{U}\mathbf{U}^*)\mathcal{P}_{\mathbf{X}_\perp}\|_F \\ \leq& \|\mathbf{M} - \mathbf{U}\mathbf{U}^*\|_F \leq \frac{1}{10}\sigma_k(\mathbf{M}).\end{aligned}$$

The last equality is true because

$$\mathrm{tr}(\mathbf{X}_\perp^*\mathbf{U}\mathbf{U}^*\mathbf{X}_\perp\mathbf{X}_\perp^*\mathbf{U}\mathbf{U}^*\mathbf{X}_\perp) = \mathrm{tr}(\mathbf{X}_\perp\mathbf{X}_\perp^*\mathbf{U}\mathbf{U}^*\mathbf{X}_\perp\mathbf{X}_\perp^*\mathbf{X}_\perp\mathbf{X}_\perp^*\mathbf{U}\mathbf{U}^*\mathbf{X}_\perp\mathbf{X}_\perp^*)$$

and

$$\mathrm{tr}(\mathbf{X}_\perp\mathbf{X}_\perp^*\mathbf{M}\mathbf{X}_\perp\mathbf{X}_\perp^*\mathbf{X}_\perp\mathbf{X}_\perp^*\mathbf{M}\mathbf{X}_\perp\mathbf{X}_\perp^*) = \mathrm{tr}(\mathbf{X}_\perp\mathbf{X}_\perp^*\mathbf{X}\mathbf{S}\mathbf{X}^*\mathbf{X}_\perp\mathbf{X}_\perp^*\mathbf{X}\mathbf{S}\mathbf{X}^*\mathbf{X}_\perp\mathbf{X}_\perp^*) = 0$$

Let the principal angle between $\mathbf{X}$ and $\mathbf{W}$ be θ. According to the inequality above, $\sin^2\theta = \|\mathbf{X}_\perp^*\mathbf{W}\|^2 \le \frac{1}{9}$. So $\cos^2\theta = \sigma_{\min}^2(\mathbf{X}^*\mathbf{W}) \ge \frac{8}{9}$. We have

$$\begin{aligned}\sigma_{\min}^2(\mathbf{X}^*\mathbf{U}) =& \sigma_{\min}(\mathbf{X}^*\mathbf{U}\mathbf{U}^*\mathbf{X}) = \sigma_{\min}(\mathbf{X}^*\mathbf{W}\mathbf{D}\mathbf{W}^*\mathbf{X})\\ \ge& \sigma_{\min}(\mathbf{D})\sigma_{\min}^2(\mathbf{X}^*\mathbf{W}) \ge \frac{9}{10}\sigma_k(\mathbf{M}) \times \frac{8}{9} \ge \sigma_k(\mathbf{M})/2.\end{aligned}$$

The lemma is proved. □

Now we are well prepared for the main theorem.

Theorem 5.3.4. *Let $f(\mathbf{U}) = \|\mathbf{U}\mathbf{U}^* - \mathbf{M}\|_F^2$ and $g_i(\mathbf{U}) = \|\mathbf{e}_i^*\mathbf{U}\|^2$. Suppose after initialization, we have*

$$f(\mathbf{U}_0) \le \left(\frac{1}{20\kappa}\right)^2, \qquad \max_i g_i(\mathbf{U}_0) \le \frac{10\mu k\kappa^2}{d}.$$

Then, there exists some absolute constant c such that for any learning rate $\eta < \frac{c}{\mu dk\kappa^3\log d}$, with at least $1 - \frac{T}{d^{10}}$ probability, we will have for all $t \le T$ that

$$f(\mathbf{U}_t) \le \left(1 - \frac{\eta}{2\kappa}\right)^t\left(\frac{1}{10\kappa}\right)^2, \qquad \max_i g_i(\mathbf{U}_t) \le \frac{20\mu k\kappa^2}{d}.$$

Proof. Let the filtration be $\mathcal{F}_t = \sigma\{SG(\mathbf{U}_0), SG(\mathbf{U}_1), \cdots, SG(\mathbf{U}_{t-1})\}$, i.e., an increasing sequence of $\sigma-$field.

We define event $\epsilon_t = \left\{\forall\tau \le t, f(\mathbf{U}_\tau) \le (1-\frac{\eta}{2\kappa})^t(\frac{1}{10\kappa})^2, \max_i g_i(\mathbf{U}_\tau) \le \frac{20\mu k\kappa^2}{d}\right\}$. We aim to prove that this event happens with high probability. Conditioned on ϵ_t, we have $\|\mathbf{U}_t\| \le \sqrt{2}$, $\sigma_{\min}(\mathbf{X}^*\mathbf{U}_t) \ge \frac{1}{\sqrt{2\kappa}}$ and $\sigma_{\min}(\mathbf{U}_t^*\mathbf{U}_t) \ge \frac{1}{2\kappa}$ based on Lemma 5.3.3.

Construction of supermartingale G: Let $g_i(\mathbf{U}) = \mathbf{e}_i^*\mathbf{U}\mathbf{U}^*\mathbf{e}_i$, for any change $\Delta\mathbf{U}$, we have

$$\begin{aligned}g_i(\mathbf{U}+\Delta\mathbf{U}) =& \mathbf{e}_i^*(\mathbf{U}+\Delta\mathbf{U})(\mathbf{U}+\Delta\mathbf{U})^*\mathbf{e}_i\\ =& g_i(\mathbf{U}) + \mathbf{e}_i^*(\Delta\mathbf{U}\mathbf{U}^* + \mathbf{U}\Delta\mathbf{U}^*)\mathbf{e}_i + \|\mathbf{e}_i^*\Delta\mathbf{U}\|^2.\end{aligned}$$

For any $l \in [d]$:

$$\begin{aligned}
&\mathbb{E}\|\mathbf{e}_l^* SG(\mathbf{U})\|^2 \mathbf{1}_{\epsilon_t} \\
=&4d^4 \mathbb{E}\| \left(\mathbf{e}_l^*(\mathbf{u}_i^*\mathbf{u}_j - \mathbf{M}_{ij})\mathbf{e}_i\mathbf{e}_j^* + \mathbf{e}_l^*(\mathbf{u}_j^*\mathbf{u}_i - \mathbf{M}_{ji})\mathbf{e}_j\mathbf{e}_i^*\right)\mathbf{U}\|^2 \mathbf{1}_{\epsilon_t} \\
\leq&16d^4 \mathbb{E}\|\mathbf{e}_l^*(\mathbf{u}_i^*\mathbf{u}_j - \mathbf{M}_{ij})\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}\|^2 \mathbf{1}_{\epsilon_t} \\
\leq&16d^4 \mathbb{E}\delta_{il}|\mathbf{u}_i^*\mathbf{u}_j - \mathbf{M}_{ij}|^2 \max_i \|\mathbf{e}_i^*\mathbf{U}\|^2 \mathbf{1}_{\epsilon_t} \\
=&16d^4 \frac{1}{d^2}\sum_{i,j}\delta_{il}|\mathbf{u}_i^*\mathbf{u}_j - \mathbf{M}_{ij}|^2 \max_i \|\mathbf{e}_i^*\mathbf{U}\|^2 \mathbf{1}_{\epsilon_t} \\
=&16d^2 \|\mathbf{e}_l^*(\mathbf{U}\mathbf{U}^* - \mathbf{M})\|^2 \max_i \|\mathbf{e}_i^*\mathbf{U}\|^2 \mathbf{1}_{\epsilon_t} \\
\leq&O(\mu^2 k^2 \kappa^4).
\end{aligned}$$

The last inequality holds because we have

$$\begin{aligned}
\|\mathbf{e}_l^*(\mathbf{U}\mathbf{U}^* - \mathbf{M})\| \leq& \|\mathbf{e}_l^*\mathbf{U}\mathbf{U}^*\| + \|\mathbf{e}_l^*\mathbf{M}\| \\
\leq& \|\mathbf{U}\|\|\mathbf{U}^*\mathbf{e}_l\| + \|\mathbf{e}_l^*\mathbf{X}\| \\
\leq& \sqrt{2}\sqrt{\frac{20\mu k\kappa^2}{d}} + \sqrt{\frac{\mu k}{d}} \\
\leq& O(\sqrt{\frac{\mu k\kappa^2}{d}}).
\end{aligned}$$

On the other hand, we know

$$\begin{aligned}
&\mathbb{E}[g_i(\mathbf{U}_{t+1})\mathbf{1}_{\epsilon_t}|\mathcal{F}_t] \\
=&\mathbb{E}[g_i(\mathbf{U}_t - \eta SG(\mathbf{U}_t))\mathbf{1}_{\epsilon_t}|\mathcal{F}_t] \\
=&[g_i(\mathbf{U}_t) - \eta\mathbf{e}_i^*\mathbb{E}SG(\mathbf{U}_t)\mathbf{U}_t^*\mathbf{e}_i - \eta\mathbf{e}_i^*\mathbf{U}_t\mathbb{E}SG(\mathbf{U}_t)^*\mathbf{e}_i + \eta^2\mathbb{E}\mathbf{e}_i^*SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\mathbf{e}_i]\mathbf{1}_{\epsilon_t} \\
=&[g_i(\mathbf{U}_t) - 2\eta\mathrm{Re}(\mathbf{e}_i^*\mathbb{E}SG(\mathbf{U}_t)\mathbf{U}_t^*\mathbf{e}_i) + \eta^2\mathbb{E}\|\mathbf{e}_i^*SG(\mathbf{U}_t)\|^2]\mathbf{1}_{\epsilon_t} \\
=&[\mathbf{e}_i^*\mathbf{U}_t\mathbf{U}_t^*\mathbf{e}_i - 8\eta\mathrm{Re}(\mathbf{e}_i^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})\mathbf{U}_t\mathbf{U}_t^*\mathbf{e}_i) + \eta^2\mathbb{E}\|\mathbf{e}_i^*SG(\mathbf{U}_t)\|^2]\mathbf{1}_{\epsilon_t} \\
=&[tr(\mathbf{U}_t^*\mathbf{e}_i\mathbf{e}_i^*\mathbf{U}_t) - 8\eta\mathrm{Re}(tr(\mathbf{U}_t^*\mathbf{e}_i\mathbf{e}_i^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})\mathbf{U}_t)) + \eta^2\mathbb{E}\|\mathbf{e}_i^*SG(\mathbf{U}_t)\|^2]\mathbf{1}_{\epsilon_t} \\
=&[\mathrm{Re}(tr(\mathbf{U}_t^*\mathbf{e}_i\mathbf{e}_i^*(\mathbf{I} - 8\eta(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}))\mathbf{U}_t)) + \eta^2\mathbb{E}\|\mathbf{e}_i^*SG(\mathbf{U}_t)\|^2]\mathbf{1}_{\epsilon_t} \\
=&[\mathrm{Re}(tr(\mathbf{U}_t^*\mathbf{e}_i\mathbf{e}_i^*\mathbf{U}_t(\mathbf{I} - 8\eta\mathbf{U}_t^*\mathbf{U}_t))) + 8\eta\mathrm{Re}(tr(\mathbf{U}_t^*\mathbf{e}_i\mathbf{e}_i^*\mathbf{M}\mathbf{U}_t)) + \eta^2\mathbb{E}\|\mathbf{e}_i^*SG(\mathbf{U}_t)\|^2]\mathbf{1}_{\epsilon_t} \\
\leq&[(1 - 8\eta\sigma_{min}(\mathbf{U}_t^*\mathbf{U}_t))g_i(\mathbf{U}_t) + 8\eta\mathrm{Re}(tr(\mathbf{U}_t^*\mathbf{e}_i\mathbf{e}_i^*\mathbf{M}\mathbf{U}_t)) + \eta^2\mathbb{E}\|\mathbf{e}_i^*SG(\mathbf{U}_t)\|^2]\mathbf{1}_{\epsilon_t}
\end{aligned}$$

For the middle term, we have

$$\begin{aligned}
8\eta\mathrm{Re}(tr(\mathbf{U}_t^*\mathbf{e}_i\mathbf{e}_i^*\mathbf{M}\mathbf{U}_t)) =& 4\eta(tr(\mathbf{U}_t^*\mathbf{e}_i\mathbf{e}_i^*\mathbf{M}\mathbf{U}_t) + tr(\mathbf{U}_t^*\mathbf{M}\mathbf{e}_i\mathbf{e}_i^*\mathbf{U}_t)) \\
\leq& 8\eta\|\mathbf{U}_t^*\mathbf{e}_i\|_2\|\mathbf{e}_i^*\mathbf{M}\mathbf{U}_t\|_2 \\
\leq& 8\eta\|\mathbf{U}_t^*\mathbf{e}_i\|_2\|\mathbf{U}_t\|\|\mathbf{e}_i^*\mathbf{M}\|_2 \\
\leq& 8\eta\sqrt{\frac{20\mu k\kappa^2}{d}}\sqrt{2}\|\mathbf{e}_i^*\mathbf{X}\mathbf{S}\mathbf{X}^*\|_2 \\
\leq& 16\eta\sqrt{\frac{10\mu k\kappa^2}{d}}\|\mathbf{e}_i^*\mathbf{X}\mathbf{S}\|_2 \\
\leq& 16\eta\sqrt{\frac{10\mu k\kappa^2}{d}}\|\mathbf{e}_i^*\mathbf{X}\|_2 \\
\leq& 16\eta\sqrt{\frac{10\mu k\kappa^2}{d}}\sqrt{\frac{\mu k}{d}} = \frac{16\sqrt{10}\eta\mu k\kappa}{d}.
\end{aligned}$$

The first inequality holds because

$$\begin{aligned}
&\mathrm{tr}(\mathbf{U}_t^*\mathbf{e}_i\mathbf{e}_i^*\mathbf{M}\mathbf{U}_t) + \mathrm{tr}(\mathbf{U}_t^*\mathbf{M}\mathbf{e}_i\mathbf{e}_i^*\mathbf{U}_t) = 2\mathrm{Re}(\mathrm{tr}(\mathbf{e}_i^*\mathbf{M}\mathbf{U}_t\mathbf{U}_t^*\mathbf{e}_i)) \\
\leq& 2\|\mathbf{U}_i^*\mathbf{e}_i\|_2\|\mathbf{e}_i^*\mathbf{M}\mathbf{U}_t\|_2.
\end{aligned}$$

The fourth inequality holds because $\|\mathbf{X}\| = 1$. The fifth inequality holds because $\|\mathbf{S}\| = \|\mathbf{M}\| = 1$. The last inequality comes from the incoherence definition of $\mathbf{M}$:

$$\|\mathbf{e}_i^*\mathbf{X}\|_2 \leq \sqrt{\frac{\mu k}{d}}.$$

In this case,

$$\begin{aligned}
&\mathbb{E}[g_i(\mathbf{U}_{t+1})1_{\epsilon_t}|\mathcal{F}_t] \\
\leq& [(1 - \frac{4\eta}{\kappa})g_i(\mathbf{U}_t) + \frac{16\sqrt{10}\eta\mu k\kappa}{d} + \eta^2 O(\mu^2k^2\kappa^4)]\mathbf{1}_{\epsilon_t} \\
\leq& \left[(1 - \frac{4\eta}{\kappa})g_i(\mathbf{U}_t) + 60\frac{\eta\mu k\kappa}{d}\right]\mathbf{1}_{\epsilon_t}.
\end{aligned}$$

We can get the last inequality if the stepsize η is small enough.

We let $G_{it} = \left(1 - \frac{4\eta}{\kappa}\right)^{-t} \left(g_i(\mathbf{U}_t)\mathbf{1}_{\epsilon_{t-1}} - 15\frac{\mu k\kappa^2}{d}\right)$ and have

$$\begin{aligned}
\mathbb{E}[G_{i(t+1)}|\mathcal{F}_t] =& \left(1 - \frac{4\eta}{\kappa}\right)^{-t-1} \left[\mathbb{E}[g_i(\mathbf{U}_{t+1})\mathbf{1}_{\epsilon_t}|\mathcal{F}_t] - 15\frac{\mu k\kappa^2}{d}\right] \\
\leq& \left(1 - \frac{4\eta}{\kappa}\right)^{-t} \left[g_i(\mathbf{U}_t)\mathbf{1}_{\epsilon_t} + \frac{60\eta\mu k\kappa^2}{(\kappa - 4\eta)d}\mathbf{1}_{\epsilon_t} - \frac{15\mu k\kappa^3}{(\kappa - 4\eta)d}\right] \\
\leq& G_{it}.
\end{aligned}$$

The last inequality is true because we have

$$\mathbf{1}_{\epsilon_t} \leq \mathbf{1}_{\epsilon_{t-1}}.$$

That's to say, G_{it} is a supermartingale.

Probability 1 bound for G: We know that

$$\begin{aligned}
G_{i(t+1)} - E[G_{i(t+1)}|\mathcal{F}_t] =& \left(1 - \frac{4\eta}{\kappa}\right)^{-t-1} (g_i(\mathbf{U}_{t+1})\mathbf{1}_{\epsilon_t} - \mathbb{E}[g_i(\mathbf{U}_{t+1})\mathbf{1}_{\epsilon_t}|\mathcal{F}_t]) \\
=& \left(1 - \frac{4\eta}{\kappa}\right)^{-t-1} [-\eta\mathbf{e}_i^*[\mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^* \\
& - (\mathbf{U}_t\mathbb{E}SG(\mathbf{U}_t)^* + \mathbb{E}SG(\mathbf{U}_t)\mathbf{U}_t^*)]\mathbf{e}_i \\
& + \eta^2[\|\mathbf{e}_i^* SG(\mathbf{U}_t)\|^2 - \mathbb{E}\|\mathbf{e}_i^* SG(\mathbf{U}_t)\|^2]] \mathbf{1}_{\epsilon_t}.
\end{aligned}$$

Let $l \in [d]$, we need to approximate the upper bounds for $\mathbf{e}_l^*[[SG(\mathbf{U}_t)]\mathbf{U}_t^* + \mathbf{U}_t SG(\mathbf{U}_t)^*]\mathbf{e}_l$ and $\|\mathbf{e}_l^* SG(\mathbf{U}_t)\|^2\mathbf{1}_{\epsilon_t}$. For the first term, we have

$$\begin{aligned}
& \mathbf{e}_l^*[[SG(\mathbf{U}_t)]\mathbf{U}_t^* + \mathbf{U}_t SG(\mathbf{U}_t)^*]\mathbf{e}_l \\
=& \mathbf{e}_l^*[(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^* + (\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*]\mathbf{U}_t\mathbf{U}_t^*\mathbf{e}_l + \\
& \mathbf{e}_l^*\mathbf{U}_t\mathbf{U}_t^*[(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^* + (\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*]\mathbf{e}_l.
\end{aligned}$$

We consider the upper bounds for different conditions.

- If $l \neq i$ and $l \neq j$, we have $\mathbf{e}_l^*[[SG(\mathbf{U}_t)]\mathbf{U}_t^* + \mathbf{U}_t SG(\mathbf{U}_t)^*]\mathbf{e}_l = 0$.

- if $l = j \neq i$, we have

$$\begin{aligned}
&\mathbf{e}_l^*[[SG(\mathbf{U}_t)]\mathbf{U}_t^* + \mathbf{U}_t SG(\mathbf{U}_t)^*]\mathbf{e}_l \\
=&2d^2(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ji}\mathbf{e}_i^*\mathbf{U}_t\mathbf{U}_t^*\mathbf{e}_j + \mathbf{e}_j^*\mathbf{U}_t\mathbf{U}_t^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i \\
=&2d^2\mathrm{Re}[(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i^*\mathbf{U}_t\mathbf{U}_t^*\mathbf{e}_j] \\
\leq&2d^2\|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}\|_\infty \max_i \|\mathbf{e}_i^*\mathbf{U}_t\|^2 \leq O(\mu^2k^2\kappa^4).
\end{aligned}$$

  The last second inequality is true because we have

$$\begin{aligned}
&\|\mathbf{U}\mathbf{U}^* - \mathbf{M}\|_\infty \leq \|\mathbf{U}\mathbf{U}^*\|_\infty + \|\mathbf{M}\|_\infty \\
\leq& \max_i \|\mathbf{e}_i^*\mathbf{U}\|^2 + \max_i |\mathbf{e}_i\mathbf{X}\mathbf{S}\mathbf{X}^*\mathbf{e}_i| \\
\leq& \frac{20\mu k\kappa^2}{d} + \|\mathbf{M}\|\frac{\mu k}{d} \leq \frac{21\mu k\kappa^2}{d}.
\end{aligned}$$

- if $l = j = i$, we have

$$\begin{aligned}
&\mathbf{e}_l^*[[SG(\mathbf{U}_t)]\mathbf{U}_t^* + \mathbf{U}_t SG(\mathbf{U}_t)^*]\mathbf{e}_l \\
=&2d^2(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ii}\mathbf{e}_i^*\mathbf{U}_t\mathbf{U}_t^*\mathbf{e}_i + 2d^2\mathbf{e}_i^*\mathbf{U}_t\mathbf{U}_t^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ii}\mathbf{e}_i \\
=&4d^2(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ii}\|\mathbf{e}_i^*\mathbf{U}_t\|^2 \\
\leq&4d^2\|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}\|_\infty \max_i \|\mathbf{e}_i^*\mathbf{U}_t\|^2 \leq O(\mu^2k^2\kappa^4).
\end{aligned}$$

Therefore, we can always have $\mathbf{e}_l^*[[SG(\mathbf{U}_t)]\mathbf{U}_t^* + \mathbf{U}_t SG(\mathbf{U}_t)^*]\mathbf{e}_l\mathbf{1}_{\epsilon_t} \leq O(\mu^2k^2\kappa^4)\mathbf{1}_{\epsilon_t}$. For the second term, we have

$$\begin{aligned}
&\|\mathbf{e}_l^*SG(\mathbf{U}_t)\|^2\mathbf{1}_{\epsilon_t} \\
\leq&4d^4|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}|_{ij}^2(\|\mathbf{e}_l^*\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}_t^*\| + \|\mathbf{e}_l^*\mathbf{e}_j\mathbf{e}_i^*\mathbf{U}_t^*\|)^2\mathbf{1}_{\epsilon_t} \\
\leq&16d^4\|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}\|_\infty^2 \max_i \|\mathbf{e}_i^*\mathbf{U}_t\|^2\mathbf{1}_{\epsilon_t} \leq O(\mu^3dk^3\kappa^6)\mathbf{1}_{\epsilon_t}
\end{aligned}$$

For any $l \in [d]$, we have $\mathbf{e}_l^*[[SG(\mathbf{U}_t)]\mathbf{U}_t^* + \mathbf{U}_t SG(\mathbf{U}_t)^*]\mathbf{e}_l \leq O(\mu^2k^2\kappa^4)$ and $\|\mathbf{e}_l^*SG(\mathbf{U}_t)\|^2 \leq O(\mu^3dk^3\kappa^6)$. Since the (i, j) component is randomly sampled from $\mathbf{M}$. In this case, we also have

$$\mathbb{E}(\mathbf{e}_l^*[[SG(\mathbf{U}_t)]\mathbf{U}_t^* + \mathbf{U}_t SG(\mathbf{U}_t)^*]\mathbf{e}_l) \leq O(\mu^2k^2\kappa^4),$$

and

$$\mathbb{E}(\|\mathbf{e}_l^* SG(\mathbf{U}_t)\|^2 \mathbf{1}_{\epsilon_t}) \leq O(\mu^3 d k^3 \kappa^6) \mathbf{1}_{\epsilon_t}.$$

In fact, we have that $\mathbb{E}[G_{i(t+1)}|\mathcal{F}_t] \geq 0$. By letting η small enough, we have with probability 1,

$$G_{i(t+1)} - \mathbb{E}[G_{i(t+1)}|\mathcal{F}_t] \leq \left(1 - \frac{4\eta}{\kappa}\right)^{-t-1} \eta O(\mu^2 k^2 \kappa^4) \mathbf{1}_{\epsilon_t}. \tag{5.16}$$

Variance bound for G: We need to approximate an upper bound for two variances: $\mathrm{Var}(\mathbf{e}_l^*[[SG(\mathbf{U}_t)]\mathbf{U}_t^* + \mathbf{U}_t SG(\mathbf{U}_t)^*]\mathbf{e}_l \mathbf{1}_{\epsilon_t})$ and $\mathrm{Var}(\|\mathbf{e}_l^* SG(\mathbf{U}_t)\|^2 \mathbf{e}_l \mathbf{1}_{\epsilon_t})$. For the first term, we have:

$$\begin{aligned}
&\mathrm{Var}(\mathrm{Re}(\mathbf{e}_l^*[SG(\mathbf{U}_t)]\mathbf{U}_t^*\mathbf{e}_l) \cdot \mathbf{1}_{\epsilon_t}|\mathcal{F}_t) \\
\leq &\mathbb{E}[(\mathrm{Re}(\mathbf{e}_l^*[SG(\mathbf{U}_t)]\mathbf{U}_t^*\mathbf{e}_l))^2 \cdot \mathbf{1}_{\epsilon_t}|\mathcal{F}_t] \\
= &\mathbb{E}[(\mathrm{Re}(\mathrm{tr}(\mathbf{U}_t^*\mathbf{e}_l\mathbf{e}_l^*[SG(\mathbf{U}_t)])))^2 \cdot \mathbf{1}_{\epsilon_t}|\mathcal{F}_t] \\
\leq &4d^4\mathbb{E}[(\mathrm{Re}(\mathrm{tr}(\mathbf{U}_t^*\mathbf{e}_l\mathbf{e}_l^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}_t)) \\
&+ \mathrm{Re}(\mathrm{tr}(\mathbf{U}_t^*\mathbf{e}_l\mathbf{e}_l^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*\mathbf{U}_t)))^2 \cdot \mathbf{1}_{\epsilon_t}|\mathcal{F}_t] \\
\leq &8d^4\mathbb{E}[(\mathrm{Re}(\mathrm{tr}(\mathbf{U}_t^*\mathbf{e}_l\mathbf{e}_l^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}_t)))^2 \cdot \mathbf{1}_{\epsilon_t}|\mathcal{F}_t] \\
&+ 8d^4\mathbb{E}[(\mathrm{Re}(\mathrm{tr}(\mathbf{U}_t^*\mathbf{e}_l\mathbf{e}_l^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*\mathbf{U}_t)))^2 \cdot \mathbf{1}_{\epsilon_t}|\mathcal{F}_t] \\
= &16d^2\sum_{i,j}[(\mathrm{Re}(\mathrm{tr}(\mathbf{U}_t^*\mathbf{e}_l\mathbf{e}_l^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}_t)))^2 \cdot \mathbf{1}_{\epsilon_t}|\mathcal{F}_t] \\
\leq &16d^2\sum_j |(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{lj}|^2 \max_i \|\mathbf{U}_t^*\mathbf{e}_i\|^4 \mathbf{1}_{\epsilon_t} \\
= &16d^2\|\mathbf{e}_l(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})\|^2 \max_j \|\mathbf{e}_j^*\mathbf{U}_t\|^4 \mathbf{1}_{\epsilon_t} \leq O\left(\frac{\mu^3 k^3 \kappa^6}{d}\right) \mathbf{1}_{\epsilon_t}.
\end{aligned}$$

The second equality is based on the definition of expectation. For the fourth inequality, we need to consider different conditions.

- if $l \neq i$, $(\mathrm{Re}(\mathrm{tr}(\mathbf{U}_t^*\mathbf{e}_l\mathbf{e}_l^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}_t)))^2 = 0$.

- if $l = i$, we have

$$\begin{aligned}&(\mathrm{Re}(\mathrm{tr}(\mathbf{U}_t^*\mathbf{e}_l\mathbf{e}_l^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}_t)))^2 = (\mathrm{Re}(\mathrm{tr}(\mathbf{U}_t^*\mathbf{e}_l(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{lj}\mathbf{e}_j^*\mathbf{U}_t)))^2\\ \leq&\|\mathbf{U}_t^*\mathbf{e}_l\|_2^2\|(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{lj}\mathbf{e}_j^*\mathbf{U}_t\|_2^2 \leq |\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{lj}|^2 \max_i \|\mathbf{U}_t^*\mathbf{e}_i\|^4.\end{aligned}$$

For the second term, we have:

$$\begin{aligned}&\mathrm{Var}(\|\mathbf{e}_l^* SG(\mathbf{U}_t)\|^2\mathbf{1}_{\epsilon_t}|\mathcal{F}_t)\\ \leq&\mathbb{E}(\|\mathbf{e}_l^* SG(\mathbf{U}_t)\|^4\mathbf{1}_{\epsilon_t}|\mathcal{F}_t)^2\\ =&16\frac{1}{d^2}\sum_{i,j} d^8\|\mathbf{e}_l^*((\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^* + (\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*)\mathbf{U}_t\|^4\mathbf{1}_{\epsilon_t}\\ \leq&128d^6\sum_{i,j}\|\mathbf{e}_l^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}_t\|^4\mathbf{1}_{\epsilon_t} + 128d^6\sum_{i,j}\|\mathbf{e}_l^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*\mathbf{U}_t\|^4\mathbf{1}_{\epsilon_t}\\ =&256d^6\sum_j|(\mathbf{U}\mathbf{U}^* - \mathbf{M})_{lj}|^4\|\mathbf{e}_j^*\mathbf{U}_t\|^4\mathbf{1}_{\epsilon_t}\\ \leq&O(1)d^6\|\mathbf{U}\mathbf{U}^* - \mathbf{M}\|_\infty^2\|\mathbf{e}_l^*(\mathbf{U}\mathbf{U}^* - \mathbf{M})\|^2\max_i\|\mathbf{e}_i^*\mathbf{U}\|^4\mathbf{1}_{\epsilon_t}\\ \leq&O(\mu^5 dk^5\kappa^{10})\mathbf{1}_{\epsilon_t}.\end{aligned}$$

The second inequality follows from $(a+b)^4 \leq 8a^4 + 8b^4$, and the second equality holds by considering the two cases $l \neq i$ and $l = i$.

Therefore, we can choose a small η and obtain

$$\mathrm{Var}(G_{i(t+1)}|\mathcal{F}_t) \leq \left(1 - \frac{4\eta}{\kappa}\right)^{-2t-2}\eta^2 O\left(\frac{\mu^3k^3\kappa^6}{d}\right)\mathbf{1}_{\epsilon_t}. \tag{5.17}$$

Berstein's inequality for G: Let $\sigma^2 = \sum_{\tau=1}^t \mathrm{Var}(G_{i\tau}|\mathcal{F}_{\tau-1})$, and there exists R such that, $|G_{i\tau} - E[G_{i\tau}|\mathcal{F}_{\tau-1}]| \leq R$, $\tau = 1, \dots t.$ with probability 1. Then by standard Berstein concentration inequality,

$$P(G_{it} \geq G_{i0} + s) \leq \exp\left(-\frac{s^2/2}{\sigma^2 + Rs/3}\right). \tag{5.18}$$

Since $G_{i0} = g_i(\mathbf{U}_0) - 15\frac{\mu k\kappa^2}{d}$, let $\tilde{s} = O(1)\left(1 - \frac{4\eta}{\kappa}\right)^t[\sqrt{\sigma^2\log d} + R\log d]$, we know

$$P\left(g_i(\mathbf{U}_t)1_{\epsilon_{t-1}} \geq 15\frac{\mu k\kappa^2}{d} + \left(1 - \frac{4\eta}{\kappa}\right)^t(g_i(\mathbf{U}_0) - 15\frac{\mu k\kappa^2}{d}) + \tilde{s}\right) \leq \frac{1}{2d^{11}}. \tag{5.19}$$

Based on (5.16), we know $R = (1-\frac{4\eta}{\kappa})^{-t}\eta O(\mu^2k^2\kappa^4)$ satisfies $G_{i\tau} - \mathbb{E}[G_{i\tau}|\mathcal{F}_{\tau-1}] \leq R$ where $\tau = 1, \ldots, t$. Also, by the variance bound for G in (5.17), we can have

$$\begin{aligned}\left(1-\frac{4\eta}{\kappa}\right)^t\sqrt{\sigma^2\log d} \leq& \eta O\left(\sqrt{\frac{\mu^3k^3\kappa^6\log d}{d}}\right)\sqrt{\sum_{\tau=1}^{t}\left(1-\frac{4\eta}{\kappa}\right)^{2t-2\tau}}\\ \leq& \eta O\left(\sqrt{\frac{\mu^3k^3\kappa^6\log d}{d}}\right)\sqrt{\frac{\kappa}{\eta}} \leq \sqrt{\eta}O\left(\sqrt{\frac{\mu^3k^3\kappa^7\log d}{d}}\right).\end{aligned}$$

By choosing $\eta < \frac{c}{\mu dk\kappa^3\log d}$ and choosing c to be small enough, we have

$$\tilde{s} \leq \sqrt{\eta}O\left(\sqrt{\frac{\mu^3k^3\kappa^7\log d}{d}}\right) + \eta O(\mu^2k^2\kappa^4\log d) \leq O\left(\frac{\mu k\kappa^2}{d}\right) + O\left(\frac{\mu k\kappa}{d}\right) \leq \frac{\mu k\kappa^2}{d}.$$

Since we have initialization $\max_i g_i(\mathbf{U}_0) \leq \frac{10\mu k\kappa^2}{d}$, by the Bernstein's inequality, we have

$$P\left(g_i(\mathbf{U}_t)\mathbf{1}_{\epsilon_{t-1}} \geq 20\frac{\mu k\kappa^2}{d}\right) \leq \frac{1}{2d^{11}}.$$

Namely,

$$P\left(\epsilon_{t-1} \cap \left\{g_i(\mathbf{U}_t)\mathbf{1}_{\epsilon_{t-1}} \geq 20\frac{\mu k\kappa^2}{d}\right\}\right) \leq \frac{1}{2d^{11}}.$$

We also need to construct another supermartingale F.

Construction of supermatingale F: From the definition of $SG(\mathbf{U}_t)$, we have

$$\begin{aligned}&\mathbb{E}\|SG(\mathbf{U}_t)\|_F^2\mathbf{1}_{\epsilon_t}\\ \leq& 16d^4\mathbb{E}(\mathbf{U}\mathbf{U}^* - \mathbf{M})_{ij}^2\max_i\|\mathbf{e}_i^*\mathbf{U}_t\|^2\mathbf{1}_{\epsilon_t} \qquad (5.20)\\ \leq& 16d^2\|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}\|_F^2\max_i\|\mathbf{e}_i^*\mathbf{U}_t\|^2\mathbf{1}_{\epsilon_t} \leq O(\mu dk\kappa^2)f(\mathbf{U}_t)\mathbf{1}_{\epsilon_t}.\end{aligned}$$

By the update equation

$$\mathbf{U}_{t+1} = \mathbf{U}_t - \eta SG(\mathbf{U}_t), \qquad (5.21)$$

we can have

$$\begin{aligned}
&\mathbb{E}[f(\mathbf{U}_{t+1})\mathbf{1}_{\epsilon_t}|\mathcal{F}_t] \\
\leq&[f(\mathbf{U}_t) - \mathbb{E}\langle\nabla f(\mathbf{U}_t), \eta SG(\mathbf{U}_t)\rangle + \eta^2\mathbb{E}\|SG(\mathbf{U}_t)\|_F^2]\mathbf{1}_{\epsilon_t} \\
=&[f(\mathbf{U}_t) - \eta\|\nabla f(\mathbf{U}_t)\|_F^2 + \eta^2\mathbb{E}\|SG(\mathbf{U}_t)\|_F^2]\mathbf{1}_{\epsilon_t} \\
\leq&\left[\left(1 - \frac{2\eta}{\kappa}\right) f(\mathbf{U}_t) + \eta^2 O(\mu k d\kappa^2) f(\mathbf{U}_t)\right]\mathbf{1}_{\epsilon_t} \\
\leq&\left(1 - \frac{\eta}{\kappa}\right) f(\mathbf{U}_t)\mathbf{1}_{\epsilon_t}.
\end{aligned}$$

The first inequality comes from second order Taylor expansion, and we choose a small η and increase the coefficient on the second order term from $\eta^2/2$ to η^2. The second inequality uses Lemma 5.3.2. The last inequality holds when we choose a small η.

Let $F_t = (1 - \frac{\eta}{\kappa})^{-t} f(\mathbf{U}_t)\mathbf{1}_{\epsilon_{t-1}}$, then

$$\mathbb{E}[F_{t+1}|\mathcal{F}_t] = \left(1 - \frac{\eta}{\kappa}\right)^{-t-1}\mathbb{E}[f(\mathbf{U}_{t+1})\mathbf{1}_{\epsilon_t}|\mathcal{F}_t] \leq \left(1 - \frac{\eta}{\kappa}\right)^{-t} f(\mathbf{U}_t)\mathbf{1}_{\epsilon_t} \leq F_t.$$

Therefore, F_t is a supermartingale.

Probability 1 bound for F: From the definition of F, we have

$$\begin{aligned}
F_{t+1} &= \left(1 - \frac{\eta}{\kappa}\right)^{-t-1} f(\mathbf{U}_{t+1})\mathbf{1}_{\epsilon_t} \\
&= \left(1 - \frac{\eta}{\kappa}\right)^{-t-1} \|\mathbf{U}_{t+1}\mathbf{U}_{t+1}^* - \mathbf{M}\|_F^2\mathbf{1}_{\epsilon_t} \\
&= \left(1 - \frac{\eta}{\kappa}\right)^{-t-1} \|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M} - \eta(\mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*) \\
&\qquad + \eta^2 SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\|_F^2\mathbf{1}_{\epsilon_t}.
\end{aligned}$$

Define $\hat{f}(\eta) := f(\mathbf{U}_t - \eta SG(\mathbf{U}_t))$. By the second order Taylor expansion with respect to η,

we can have

$$\begin{aligned}
f(\mathbf{U}_{t+1}) =& \hat{f}(0) + \eta\nabla\hat{f}(0) + \frac{\eta^2}{2}\nabla^2\hat{f}(\xi) \\
=& \|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}\|_F^2 + \eta(-2\mathrm{Re}\langle \mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*\rangle) \\
&+ \frac{\eta^2}{2}(2\langle \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*, \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*\rangle \\
&+ 4\mathrm{Re}\langle \mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle \\
&- 12\xi\mathrm{Re}\langle \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*, SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle \\
&+ 12\xi^2\langle SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*, SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle).
\end{aligned}$$

where

$$\nabla\hat{f}(0) = -2\mathrm{Re}\langle \mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*\rangle,$$

and

$$\begin{aligned}
\nabla^2\hat{f}(\xi) =& 2\langle \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*, \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*\rangle \\
&+ 4\mathrm{Re}\langle \mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle \\
&- 12\xi\mathrm{Re}\langle \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*, SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle \\
&+ 12\xi^2\langle SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*, SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle \\
=& 4\mathrm{Re}\langle (\mathbf{U}_t - \xi SG(\mathbf{U}_t))(\mathbf{U}_t - \xi SG(\mathbf{U}_t))^* - \mathbf{M}, SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle \\
&+ 2\|SG(\mathbf{U}_t)(\mathbf{U}_t - \xi SG(\mathbf{U}_t))^* + (\mathbf{U}_t - \xi SG(\mathbf{U}_t))SG(\mathbf{U}_t)^*\|_F^2.
\end{aligned}$$

Then we need to bound

$$F_{t+1} - \mathbb{E}[F_{t+1}|\mathcal{F}_t] = \left(1 - \frac{\eta}{\kappa}\right)^{-t-1}[f(\mathbf{U}_{t+1}) - \mathbb{E}(f(\mathbf{U}_{t+1})|\mathcal{F}_t)]\mathbf{1}_{\epsilon_t}.$$

Firstly,

$$
\begin{aligned}
&\|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}\|_\infty \mathbf{1}_{\epsilon_t} \\
=&\max_{ij} |\mathbf{e}_i(\mathbf{U}_t\mathbf{U}_t - \mathbf{M})\mathbf{e}_j| \mathbf{1}_{\epsilon_t} \\
=&\max_{ij} |\mathbf{e}_i(\mathcal{P}_{\mathbf{X}} + \mathcal{P}_{\mathbf{X}\perp})(\mathbf{U}_t\mathbf{U}_t - \mathbf{M})\mathbf{e}_j| \mathbf{1}_{\epsilon_t} \\
\leq&\max_{ij} |\mathbf{e}_i\mathcal{P}_{\mathbf{X}}(\mathbf{U}_t\mathbf{U}_t - \mathbf{M})\mathbf{e}_j| \mathbf{1}_{\epsilon_t} + \max_{ij} |\mathbf{e}_i\mathcal{P}_{\mathbf{X}_\perp}\mathbf{U}_t\mathbf{U}_t\mathbf{e}_j| \mathbf{1}_{\epsilon_t} \\
\leq&\max_{ij} \|\mathbf{e}_i^*\mathbf{X}\|\|\mathbf{X}^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})\mathbf{e}_j\| \mathbf{1}_{\epsilon_t} + \|\mathbf{e}_j^*\mathbf{W}\|\|\mathbf{W}^*\mathbf{W}\mathbf{D}\mathbf{W}^*\mathcal{P}_{\mathbf{X}_\perp}\mathbf{e}_i\| \\
\leq&\max_{ij} \|\mathbf{e}_i^*\mathbf{X}\|\|(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})\mathbf{e}_j\| + \|\mathbf{e}_j^*\mathbf{W}\|\|\mathbf{W}\mathbf{D}\mathbf{W}^*\mathcal{P}_{\mathbf{X}_\perp}\mathbf{e}_i\| \\
\leq&\max_{i} \|\mathbf{e}_i^*\mathbf{X}\|\|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}\|_F + \|\mathbf{e}_j^*\mathbf{W}\|\|(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})\mathcal{P}_{\mathbf{X}_\perp}\mathbf{e}_i\| \\
\leq&O\left(\sqrt{\frac{\mu k}{d}}\right)\sqrt{f(\mathbf{U}_t)} + O\left(\sqrt{\frac{\mu k\kappa^3}{d}}\right)\sqrt{f(\mathbf{U}_t)} \\
\leq&O\left(\sqrt{\frac{\mu k\kappa^3}{d}}\right)\sqrt{f(\mathbf{U}_t)}.
\end{aligned}
$$

The fifth inequality comes from

$$
\|\mathbf{e}_i^*\mathbf{W}\| \leq \|\mathbf{e}_i\mathbf{W}\mathbf{D}^{\frac{1}{2}}\|\frac{1}{\lambda_{\min}^{\frac{1}{2}}(\mathbf{D})} = \frac{\|\mathbf{e}_i\mathbf{U}_t\|}{\lambda_{\min}^{\frac{1}{2}}(\mathbf{D})} \leq \sqrt{2\kappa}\sqrt{\frac{20\mu k\kappa^2}{d}}.
$$

As we know, for the first-order derivative,

$$
\begin{aligned}
&\mathrm{Re}\langle \mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*\rangle \\
=&\mathrm{Re}\langle \mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, \mathbf{U}_t SG(\mathbf{U}_t)^*\rangle + \mathrm{Re}\langle \mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, SG(\mathbf{U}_t)\mathbf{U}_t^*\rangle \\
\leq&2\sqrt{2}\|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}\|_F\|SG(\mathbf{U}_t)\|_F \\
=&4\sqrt{2}d^2\sqrt{f(\mathbf{U}_t)}\|(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}_t + (\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*\mathbf{U}_t\|_F \\
\leq&4\sqrt{2}d^2\sqrt{f(\mathbf{U}_t)}(\|(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}_t\|_F + \|(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*\mathbf{U}_t\|_F) \\
\leq&4\sqrt{2}d^2\sqrt{f(\mathbf{U}_t)}\|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}\|_\infty(\|\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}_t\|_F + \|\mathbf{e}_j\mathbf{e}_i^*\mathbf{U}_t\|_F) \\
\leq&8\sqrt{2}d^2\sqrt{f(\mathbf{U}_t)}\|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}^*\|_\infty \max_i \|\mathbf{e}_i^*\mathbf{U}_t\| \\
\leq&O(\mu dk\kappa^{2.5})f(\mathbf{U}_t).
\end{aligned}
$$

Here, the first inequality comes from $\|\mathbf{U}\| \leq \sqrt{2}$.

For the second-order derivative, with a small enough η, we have

$$\begin{aligned}
&4\mathrm{Re}\langle(\mathbf{U}_t - \xi SG(\mathbf{U}_t))(\mathbf{U}_t - \xi SG(\mathbf{U}_t))^* - \mathbf{M}, SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle \\
&+ 2\|SG(\mathbf{U}_t)(\mathbf{U}_t - \xi SG(\mathbf{U}_t))^* + (\mathbf{U}_t - \xi SG(\mathbf{U}_t))SG(\mathbf{U}_t)^*\|_F^2 \\
\leq& O(1)\|SG(\mathbf{U}_t)\|_F^2 \\
\leq& O(1)d^4\|\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}\|_\infty^2 \max_i \|\mathbf{e}_i^*\mathbf{U}_t\|^2 \\
\leq& O(\mu^2 d^2 k^2 \kappa^5) f(\mathbf{U}_t).
\end{aligned}$$

The first inequality holds because when η is small enough, we have an uniform upper bound for $\|\mathbf{U} - \xi SG(\mathbf{U}_t)\|$ and $\|(\mathbf{U}_t - \xi SG(\mathbf{U}_t))(\mathbf{U}_t - \xi SG(\mathbf{U}_t))^* - \mathbf{M}\|$. In this case, if we choose η to be small enought, we can have with probability 1,

$$\begin{aligned}
|F_{t+1} - \mathbb{E}[F_{t+1}|\mathcal{F}_t]| &\leq \left(1 - \frac{\eta}{\kappa}\right)^{-t-1} \eta O(\mu dk\kappa^{2.5}) f(\mathbf{U}_{t+1})\mathbf{1}_{\epsilon_t} \\
&\leq \left(1 - \frac{\eta}{\kappa}\right)^{-t-1} \left(1 - \frac{\eta}{2\kappa}\right)^{t+1} \eta O(\mu dk\kappa^{0.5}).
\end{aligned} \tag{5.22}$$

Variance bound for F: For the first order derivative,

$$\begin{aligned}
&\mathrm{Var}(\mathrm{Re}\langle\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*\rangle) \\
\leq& \mathbb{E}(\mathrm{Re}\langle\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, \mathbf{U}_t SG(\mathbf{U}_t)^* + SG(\mathbf{U}_t)\mathbf{U}_t^*\rangle)^2 \\
\leq& 2\mathbb{E}(\mathrm{Re}\langle\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, \mathbf{U}_t SG(\mathbf{U}_t)^*)^2 + 2\mathbb{E}(\mathrm{Re}\langle\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M}, SG(\mathbf{U}_t)\mathbf{U}_t^*\rangle)^2 \\
\leq& 2\|(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})\mathbf{U}_t\|_F^2 \mathbb{E}\|SG(\mathbf{U}_t)\|_F^2 + 2\|\mathbf{U}_t^*(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})\|_F^2 \mathbb{E}\|SG(\mathbf{U}_t)\|_F^2 \\
\leq& 8\mathbb{E}\|(\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})\|_F^2 \mathbb{E}\|SG(\mathbf{U}_t)\|_F^2 \\
\leq& O(\mu dk\kappa^2) f^2(\mathbf{U}_t),
\end{aligned}$$

where the last inequality comes from (5.20).

For the second order derivative, when η is small enough, we have

$$
\begin{aligned}
&\mathrm{Var}(4\mathrm{Re}\langle(\mathbf{U}_t-\xi SG(\mathbf{U}_t))(\mathbf{U}_t-\xi SG(\mathbf{U}_t))^*-\mathbf{M},SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle\\
&\quad+2\|SG(\mathbf{U}_t)(\mathbf{U}_t-\xi SG(\mathbf{U}_t))^*+(\mathbf{U}_t-\xi SG(\mathbf{U}_t))SG(\mathbf{U}_t)^*\|_F^2)\\
\le&\mathbb{E}(4\mathrm{Re}\langle(\mathbf{U}_t-\xi SG(\mathbf{U}_t))(\mathbf{U}_t-\xi SG(\mathbf{U}_t))^*-\mathbf{M},SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle\\
&\quad+2\|SG(\mathbf{U}_t)(\mathbf{U}_t-\xi SG(\mathbf{U}_t))^*+(\mathbf{U}_t-\xi SG(\mathbf{U}_t))SG(\mathbf{U}_t)^*\|_F^2)^2\\
=&\mathbb{E}(4\mathrm{Re}\langle(\mathbf{U}_t-\xi SG(\mathbf{U}_t))(\mathbf{U}_t-\xi SG(\mathbf{U}_t))^*-\mathbf{M},SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle)^2\\
&\quad+\mathbb{E}(2\|SG(\mathbf{U}_t)(\mathbf{U}_t-\xi SG(\mathbf{U}_t))^*+(\mathbf{U}_t-\xi SG(\mathbf{U}_t))SG(\mathbf{U}_t)^*\|_F^2)^2)^2\\
&\quad+\mathbb{E}(8\mathrm{Re}\langle(\mathbf{U}_t-\xi SG(\mathbf{U}_t))(\mathbf{U}_t-\xi SG(\mathbf{U}_t))^*-\mathbf{M},SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle\\
&\quad\cdot\|SG(\mathbf{U}_t)(\mathbf{U}_t-\xi SG(\mathbf{U}_t))^*+(\mathbf{U}_t-\xi SG(\mathbf{U}_t))SG(\mathbf{U}_t)^*\|_F^2)\\
\le&\mathbb{E}2(4\mathrm{Re}\langle(\mathbf{U}_t-\xi SG(\mathbf{U}_t))(\mathbf{U}_t-\xi SG(\mathbf{U}_t))^*-\mathbf{M},SG(\mathbf{U}_t)SG(\mathbf{U}_t)^*\rangle)^2\\
&\quad+\mathbb{E}2(2\|SG(\mathbf{U}_t)(\mathbf{U}_t-\xi SG(\mathbf{U}_t))^*+(\mathbf{U}_t-\xi SG(\mathbf{U}_t))SG(\mathbf{U}_t)^*\|_F^2)^2)^2\\
\le&O(1)\mathbb{E}\|SG(\mathbf{U}_t)\|_F^4\\
=&O(d^8)\mathbb{E}\|(\mathbf{U}_t\mathbf{U}_t^*-\mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\mathbf{U}+(\mathbf{U}_t\mathbf{U}_t^*-\mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*\mathbf{U}_t\|_F^4\\
\le&O(d^8)\|\mathbf{U}_t\mathbf{U}_t^*-\mathbf{M}\|_\infty^2\mathbb{E}|(\mathbf{U}_t\mathbf{U}_t^*-\mathbf{M})_{ij}|^2\max_i\|\mathbf{e}_i\mathbf{U}_t\|^4\\
\le&O(d^6)\|\mathbf{U}_t\mathbf{U}_t^*-\mathbf{M}\|\infty^2\|\mathbf{U}_t\mathbf{U}_t^*-\mathbf{M}\|_F^2\max_i\|\mathbf{e}_i\mathbf{U}_t\|^4\\
\le&O(\mu^3d^3k^3\kappa^7)f^2(\mathbf{U}_t).
\end{aligned}
$$

Therefore, we can have with probability 1,

$$
\begin{aligned}
\mathrm{Var}(F_{t+1}|\mathcal{F}_t)&\le\left(1-\frac{\eta}{\kappa}\right)^{-2t-2}\eta^2O(\mu dk\kappa^2)f^2(\mathbf{U}_t)\mathbf{1}_{\epsilon_t}\\
&\le\left(1-\frac{\eta}{\kappa}\right)^{-2t-2}\left(1-\frac{\eta}{2\kappa}\right)^{2t+2}\eta^2O\left(\frac{\mu dk}{\kappa^2}\right)\mathbf{1}_{\epsilon_t}.
\end{aligned}
\tag{5.23}
$$

Berstein's inequality for F: Let $\sigma^2=\sum_{\tau=1}^t\mathrm{Var}(F_\tau|\mathcal{F}_{\tau-1})$ and R satisfies $|F_\tau-\mathbb{E}[F\tau|\mathcal{F}_{\tau-1}]|\le R$ according to (5.22), $\tau=1,\cdots,t$. Then by the standard Bernstein concentration inequality, we know:

$$
P(F_t\ge F_0+s)\le\exp\left(-\frac{s^2/2}{\sigma^2+Rs/3}\right).
$$

Let $\tilde{s} = O(1)\left(1-\frac{\eta}{\kappa}\right)^t[\sqrt{\sigma^2\log d}+R\log d]$. So when $d\geq 2$, we have

$$P\left(f(\mathbf{U}_t)\mathbf{1}_{\epsilon_{t-1}} \geq \left(1-\frac{\eta}{\kappa}\right)^t f(\mathbf{U}_0)+\tilde{s}\right) \leq \frac{1}{2d^{10}}.$$

We can know that $R = \left(1-\frac{\eta}{\kappa}\right)^{-t}\left(1-\frac{\eta}{2\kappa}\right)^t \eta O(\mu dk\kappa^{0.5})$. By the variance bound of F in (5.23), we have

$$\begin{aligned}
&\left(1-\frac{\eta}{\kappa}\right)^t\sqrt{\sigma^2\log d} \leq \eta O\left(\sqrt{\frac{\mu dk\log d}{\kappa^2}}\right)\sqrt{\sum_{\tau=1}^{t}\left(1-\frac{\eta}{\kappa}\right)^{2t-2\tau}\left(1-\frac{\eta}{2\kappa}\right)^{2\tau}} \\
\leq &\left(1-\frac{\eta}{2\kappa}\right)^t \eta O\left(\sqrt{\frac{\mu dk\log d}{\kappa^2}}\right)\sqrt{\sum_{\tau=1}^{t}\left(1-\frac{\eta}{\kappa}\right)^{2t-2\tau}\left(1-\frac{\eta}{2\kappa}\right)^{2\tau-2t}} \\
\leq &\left(1-\frac{\eta}{2\kappa}\right)\sqrt{\eta}O\left(\sqrt{\frac{\mu dk\log d}{\kappa}}\right).
\end{aligned}$$

The last inequality holds because we have

$$\sum_{\tau=1}^{t}\left(1-\frac{\eta}{\kappa}\right)^{2t-2\tau}\left(1-\frac{\eta}{2\kappa}\right)^{2\tau-2t} < \frac{4(\frac{\kappa}{\eta})^2-4\frac{\kappa}{\eta}+1}{4\frac{\kappa}{\eta}-3} \leq \frac{\kappa}{\eta}.$$

By $\eta < \frac{c}{\mu dk\kappa^3\log d}$ and choosing c to be small enough, we have:

$$\tilde{s} = \left(1-\frac{\eta}{2\kappa}\right)^t\left[\sqrt{\eta}O\left(\sqrt{\frac{\mu dk\log d}{\kappa}}\right)+\eta O(\mu dk\kappa^{0.5})\right] \leq \left(1-\frac{\eta}{2\kappa}\right)^t\left(\frac{1}{20\kappa}\right)^2.$$

Since $F_0 = f(\mathbf{U}_0) \leq \frac{1}{(20\kappa)^2}$, we can have

$$P\left(f(\mathbf{U}_t)\mathbf{1}_{\epsilon_{t-1}} \geq \left(1-\frac{\eta}{2\kappa}\right)^t\left(\frac{1}{10\kappa}\right)^2\right) \leq \frac{1}{2d^{10}}.$$

That's to say,

$$P\left(\epsilon_{t-1}\cap\left\{f(\mathbf{U}_t) \geq \left(1-\frac{\eta}{2\kappa}\right)^t\left(\frac{1}{10\kappa}\right)^2\right\}\right) \leq \frac{1}{2d^{10}}.$$

Probability for event ϵ_T: We need to combine the concentration result for martingales G and F. Then we get

$$\begin{aligned}
&P(\epsilon_{t-1}\cap\bar{\epsilon}_t) = P\left(\epsilon_{t-1}\cap\left(\cup_i\left\{g_i(\mathbf{U}_t)\geq 20\frac{\mu k\kappa^2}{d}\right\}\cup\left\{f(\mathbf{U}_t)\geq\left(1-\frac{\eta}{2\kappa}\right)^t\left(\frac{1}{10\kappa}\right)^2\right\}\right)\right) \\
\leq &\sum_{i=1}^{d}P\left(\epsilon_{t-1}\cap\left\{g_i(\mathbf{U}_t)\geq 20\frac{\mu k\kappa^2}{d}\right\}\right)+P\left(\epsilon_{t-1}\cap\left\{f(\mathbf{U}_t)\geq(1-\frac{\eta}{2\kappa})^t(\frac{1}{10\kappa})^2\right\}\right) \leq \frac{1}{d^{10}}.
\end{aligned}$$

The theorem is proved in the Hermitian case. □

5.4 Algorithms

In this section, we give algorithms for both Hermitian and general cases.

5.4.1 The Hermitian Case

For the Hermitian case, we need to find one matrix $\mathbf{U}$ so that $\mathbf{U}\mathbf{U}^* \approx \mathbf{M}$. Given an new observation (i, j), one or two rows of $\mathbf{U}$ is updated for every iteration. The SGD computation is given by (5.11), and η is the stepsize. For the convergence in this chapter, η has to satisfy $\eta < \frac{c}{\mu d k \kappa^3 \log d}$ in our theoretical proof with a small c. However, in practices, we may choose a larger stepsize. T is the total number of required observations. The algorithm is described in Algorithm 5.1.

Algorithm 5.1: Online learning algorithm for the Hermitian matrix $\mathbf{M}$

Input: Initial $\Omega_0 \in \mathbb{Q}^{d\times d}$, learning rate η, iterations T, $\mathbf{U}_0\mathbf{U}_0^* \leftarrow$ top k SVD of $\frac{d^2}{\Omega_0}\mathbb{P}_{\Omega_0}(\mathbf{M})$

Output: $\mathbf{U}$, s.t.$\mathbf{U}\mathbf{U}^* \approx \mathbf{M}$

for *t = 0, 1, 2, 3, . . . , T-1* **do**
  Observe $\mathbf{M}_{ij}$ where $(i, j) \subset \{1, \cdots, d\} \times \{1, \cdots, d\}$ is uniform distributed ;
  $\mathbf{U}_{t+1} = \mathbf{U}_t - 2\eta d^2((\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^* + (\mathbf{U}_t\mathbf{U}_t^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*))\mathbf{U}_t$
end

5.4.2 The General Case

In the general case with the quaternion matrix $\mathbf{M} \in \mathbb{Q}^{d_1\times d_2}$. We need to find two matrices $\mathbf{U} \in \mathbb{Q}^{d_1\times k}$ and $\mathbf{V} \in \mathbb{Q}^{d_2\times k}$ so that $\mathbf{U}\mathbf{V}^* \approx \mathbf{M}$. Ω_0 is the initial coordinates set of observation, from which we construct a good initialization $\mathbf{U}_0$. For each iteration given one observation $\mathbf{M}_{i,j}$, we update the i-th row of $\mathbf{U}_t$ and the j-th row $\mathbf{V}_t$. Both rows are updated by SGD in a similar way as the Hermitian case. The parameter η is the stepsize, and T is the total number of observations. The algorithm is described in Algorithm 5.2.

In practice, we use Algorithm 5.3 to increase the speed just like (Jin et al., 2016). Instead of directly doing SVD on a $d_1 \times d_2$ quaternion matrix $\mathbf{U}\mathbf{V}^*$, we do SVD on two smaller $k \times k$

quaternion matrix $\mathbf{U}_t^*\mathbf{U}_t$ and $\mathbf{V}_t^*\mathbf{V}_t$.

Algorithm 5.2: Online learning algorithm for general $\mathbf{M}$ (theoretical version)

Input: Initial $\Omega_0 \in \mathbb{Q}^{d_1\times d_2}$, learning rate η, iterations T, $\mathbf{U}_0\mathbf{V}_0^* \leftarrow$ top k SVD of $\frac{d_1\times d_2}{\Omega_0}\mathbb{P}_{\Omega_0}(\mathbf{M})$

Output: $\mathbf{U}$, $\mathbf{V}$, s.t.$\mathbf{U}\mathbf{V}^* \approx \mathbf{M}$

for *t = 0, 1, 2, 3, . . . , T-1* **do**

- $\mathbf{W}_U\mathbf{D}\mathbf{W}_V^* \leftarrow \mathrm{SVD}(\mathbf{U}_t\mathbf{V}^*)$;
- $\tilde{\mathbf{U}}_t \leftarrow \mathbf{W}_U\mathbf{D}^{\frac{1}{2}}, \tilde{\mathbf{V}}_t \leftarrow \mathbf{W}_V\mathbf{D}^{\frac{1}{2}}$;
- Observe $\mathbf{M}_{ij}$where$(i,j) \subset \{1,\cdots,d\}\times\{1,\cdots,d\}$ is uniform distributed ;
- $\mathbf{U}_{t+1} \leftarrow \tilde{\mathbf{U}}_t - 2\eta d_1 d_2(\tilde{\mathbf{U}}_t\tilde{\mathbf{V}}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\tilde{\mathbf{V}}_t$;
- $\mathbf{V}_{t+1} \leftarrow \tilde{\mathbf{V}}_t - 2\eta d_1 d_2(\tilde{\mathbf{U}}_t\tilde{\mathbf{V}}_t^* - \mathbf{M})_{ji}\mathbf{e}_i\mathbf{e}_j^*\tilde{\mathbf{V}}_t$

end

Algorithm 5.3: Online learning algorithm for general $\mathbf{M}$ (practical version)

Input: Initial $\Omega_0 \in \mathbb{Q}^{d_1\times d_2}$, learning rate η, iterations T, $\mathbf{U}_0\mathbf{V}_0^* \leftarrow$ top k SVD of $\frac{d_1\times d_2}{\Omega_0}\mathbb{P}_{\Omega_0}(\mathbf{M})$

Output: $\mathbf{U}$, $\mathbf{V}$, s.t.$\mathbf{U}\mathbf{V}^* \approx \mathbf{M}$

for *t = 0, 1, 2, 3, . . . , T-1* **do**

- Observe $\mathbf{M}_{ij}$where$(i,j) \sim \mathrm{Unif}([d_1]\times[d_2])$;
- $\mathbf{R}_U\mathbf{D}_U\mathbf{R}_U^* \leftarrow \mathrm{SVD}(\mathbf{U}_t^*\mathbf{U})$;
- $\mathbf{R}_V\mathbf{D}_V\mathbf{R}_V^* \leftarrow \mathrm{SVD}(\mathbf{V}_t^*\mathbf{V})$;
- $\mathbf{Q}_U\mathbf{D}\mathbf{Q}_V^* \leftarrow \mathrm{SVD}(\mathbf{D}_U^{\frac{1}{2}}\mathbf{R}_U^*\mathbf{R}_V(\mathbf{D}_V^{\frac{1}{2}})^*)$;
- $\mathbf{U}_{t+1} = \mathbf{U}_t - 2\eta d_1 d_2((\mathbf{U}_t\mathbf{V}_t^* - \mathbf{M})_{ij}\mathbf{e}_i\mathbf{e}_j^*\mathbf{V}_t\mathbf{R}_V\mathbf{D}_V^{-\frac{1}{2}}\mathbf{Q}_V\mathbf{Q}_U^*\mathbf{D}_U^{\frac{1}{2}}\mathbf{R}_U^*$;
- $\mathbf{V}_{t+1} = \mathbf{V}_t - 2\eta d_1 d_2((\mathbf{U}_t\mathbf{V}_t^* - \mathbf{M})_{ji}\mathbf{e}_j\mathbf{e}_i^*\mathbf{U}_t\mathbf{R}_U\mathbf{D}_U^{-\frac{1}{2}}\mathbf{Q}_U\mathbf{Q}_V^*\mathbf{D}_V^{\frac{1}{2}}\mathbf{R}_V^*$

end

5.5 Numerical Experiments

In this section, we conduct some numerical experiments for both Hermitian and general cases.

Small Hermitian case: We randomly generate a 10×10 Hermitian quaternion matrix with rank 5. The initialization is obtained with 99% of the matrix. The stepsize η is chosen as $3e^{-5}$, and the total iteration number is 40000. The total number of iteration is large comparing to the size of the matrix, and we use this example to demonstrate the linear

convergence of the proposed algorithm. For this example, each component is chosen for 400 times on average, and the algorithm converges linearly in Figure 5.2, which confirms the theoretical results.

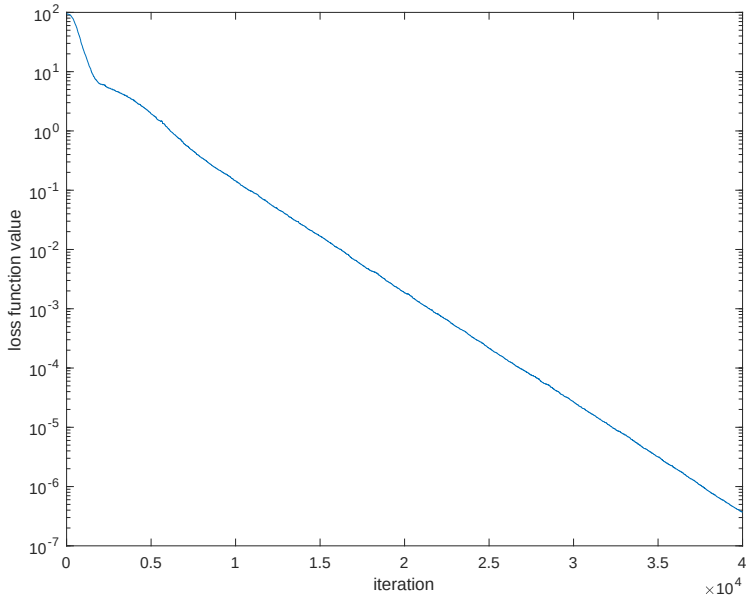


Figure 5.2: Loss function value versus number of iterations for the small Hermitian case. The stepsize is $3e^{-5}$. Within 40000 iterations, the value decreases from nearly 100 to 10^{-6}. The loss function value tends to keep decreasing after these 40000 iterations.

Small general case: We randomly generate a 10×10 general quaternion matrix with rank 5. The initialization is obtained with 99% of the matrix. The stepsize η is chosen as $1e^{-4}$, and the total iteration number is 5000. We also observe the convergence of the proposed algorithm in Figure 5.3. From both example, we can see that the algorithm converges slowly, though it converges linearly.

Color image: For a color image with depth with dimension $259 \times 320 \times 4$, we resize the image to be $156 \times 192 \times 4$. There are four channels including the depth, so we can use a quaternion matrix to describe it with the depth being the real part. We normalize it and force its rank to be 30. The initial stepsize is $1e^{-5}$, and the total iteration number is 10000. In this case, the total number of iterations is smaller than the number of components in the matrix. For every 300 iterations, we reduce the stepsize by 0.95 to help it converge.

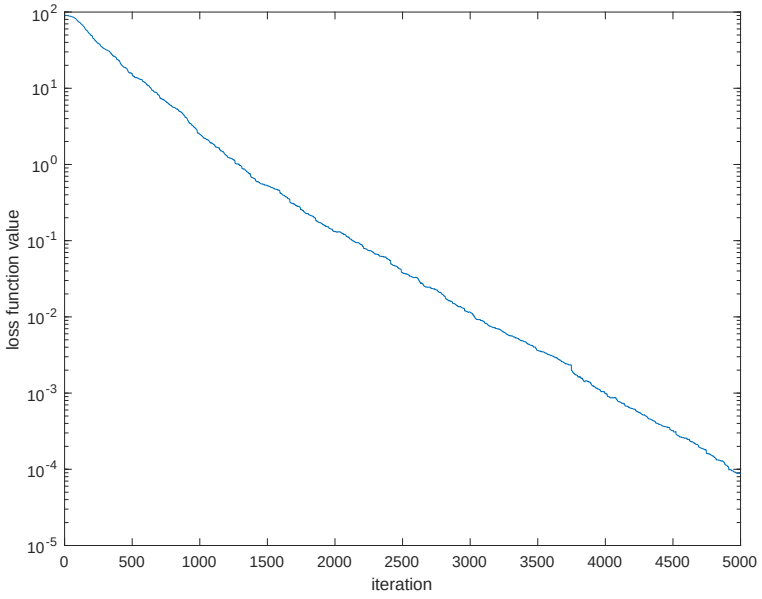


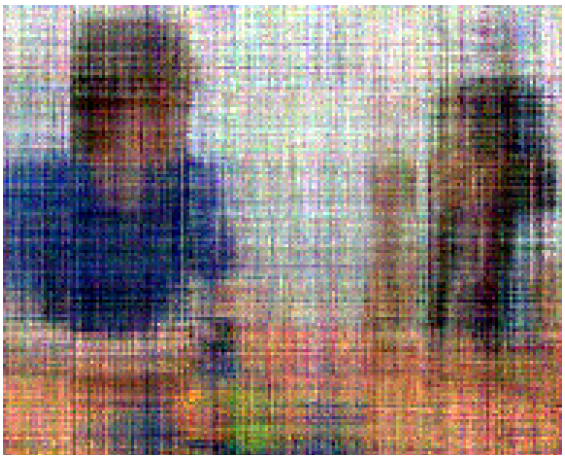
Figure 5.3: Loss function value versus number of iterations for the small general quaternion matrix case. The stepsize is tuned to be $1e^{-4}$. Within 5000 iterations, the value decreases from around 100 to 10^{-4}. The loss function value tends to keep decreasing after these 5000 iterations.

The result is shown in Figure 5.4 and Figure 5.5. We can see that the image can not be recovered well with only 10000 pixels. We are thinking about one possible reason for it. A quaternion matrix typically have three more components than the real matrix. It tends to need more observations to converge. Because only around 1/3 of the pixels are used once, which is far below the computation requires for offline algorithms. In each iteration of an offline algorithm, it goes through all observed pixels, which is about 10000 pixels, and there are many iteration required to get a good observation.

5.6 Conclusion

In this chapter, we introduce quaternion matrix and its properties. Based on previous work for online matrix completion, we set up a provable and efficient framework for online quaternion matrix completion, which can be easily applied on color images. This framework applies nonconvex SGD on quaternion matrix and we can show the performance improvement

(a) True Image.



(b) Recovered Image.

Figure 5.4: Online image recovery result after 10000 iterations. We randomly sampled 1000o observations from (a). We can see that the result for recovered image (b) is not good. We expect that the difference between (a) and (b) can be as small as possible.

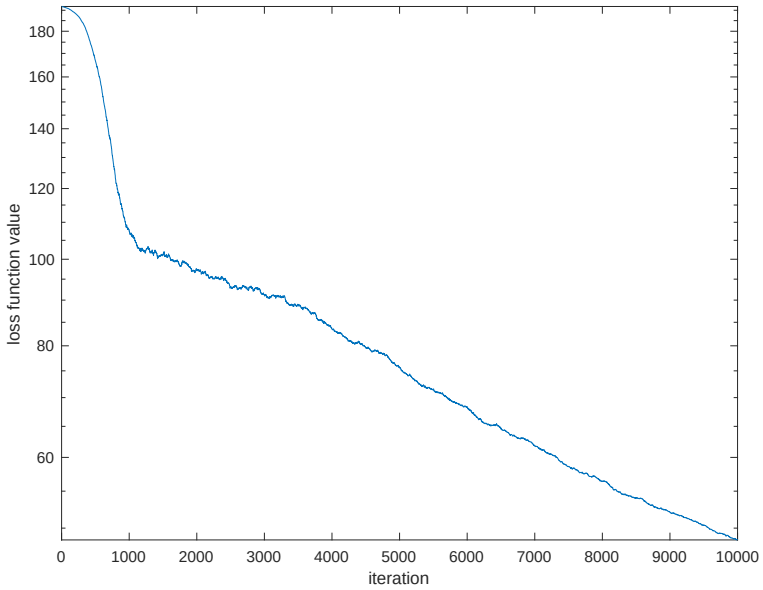


Figure 5.5: Loss function value versus number of iterations for the real color image. The initial stepsize is tuned to be $1e^{-5}$. For every 300 iterations, we multiply the stepsize by 0.95. The loss function value decreases from around 170 to almost 45. At the beginning, the loss function value decreases the most, and it tends to keep decreasing after these 10000 iterations.

based on each updated input. By using martingale theory, we prove that SGD can stay away from saddle points and converges linearly if we have a good initialization.

BIBLIOGRAPHY

BIBLIOGRAPHY

Amaldi, E. and Kann, V. (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1-2):237–260.

Beck, A. and Teboulle, M. (2009a). Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434.

Beck, A. and Teboulle, M. (2009b). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

Beck, A. and Teboulle, M. (2009c). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202.

Bouwmans, T. and Zahzah, E. H. (2014). Robust pca via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122:22–34.

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

Cabral, R., De la Torre, F., Costeira, J. P., and Bernardino, A. (2014). Matrix completion for weakly-supervised multi-label image classification. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):121–135.

Cabral, R. S., Torre, F., Costeira, J. P., and Bernardino, A. (2011). Matrix completion for multi-label image classification. In *Advances in neural information processing systems*, pages 190–198.

Cai, H., Cai, J.-F., and Wei, K. (2019). Accelerated alternating projections for robust principal component analysis. *The Journal of Machine Learning Research*, 20(1):685–717.

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

Candes, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust Principal Component Analysis? *J. ACM*, 58(3):11.

Chartrand, R. (2007). Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Processing Letters*, 14(10):707–710.

Chen, Y., Ma, J., and Fomel, S. (2016). Double-sparsity dictionary for seismic noise attenuation. *Geophysics*, 81(2):V103–V116.

Chen, Y., Zhou, Y., Chen, W., Zu, S., Huang, W., and Zhang, D. (2017). Empirical low-rank approximation for seismic noise attenuation. *IEEE Transactions on Geoscience and Remote Sensing*, 55(8):4696–4711.

Cheng, J., Chen, K., and Sacchi, M. D. (2015). Robust principle component analysis (RPCA) for seismic data denoising. In *GeoConvention 2015*.

Cunningham, J. P. and Ghahramani, Z. (2015). Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research*, 16(1):2859–2900.

Da Costa, J. F. P., Alonso, H., and Roque, L. (2009). A weighted principal component analysis and its application to gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(1):246–252.

De la Torre, F. and Black, M. J. (2001). Robust principal component analysis for computer vision. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 362–369. IEEE.

Du, C., Sun, J., Zhou, S., and Zhao, J. (2013). An Outlier Detection Method for Robust Manifold Learning. In Yin, Z., Pan, L., and Fang, X., editors, *Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2013*, Advances in Intelligent Systems and Computing, pages 353–360. Springer Berlin Heidelberg.

Duarte, L. T., Nadalin, E. Z., Nose Filho, K., Zanetti, R., Romano, J. M., and Tygel, M. (2012). Seismic wave separation by means of robust principal component analysis. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1494–1498. IEEE.

Elhamifar, E. and Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360.

Fomel, S. and Liu, Y. (2013). Seislet transform and seislet frame. *Geophysics*, 75:V25–V38.

Gaudet, C. J. and Maida, A. S. (2018). Deep quaternion networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Gavish, M. and Donoho, D. L. (2014). The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053.

Hammond, D. K., Vandergheynst, P., and Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150.

Herman, G. and Perkins, C. (2006). Predictive removal of scattered noise. *Geophysics*, 71:V41–V49.

Horn, R. A. and Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press.

Huang, X. and Yan, M. (2018). Nonconvex penalties with analytical solutions for one-bit compressive sensing. *Signal Processing*, 144:341–351.

Huang, X.-L., Shi, L., and Yan, M. (2015). Nonconvex sorted ℓ_1 minimization for sparse approximation. *Journal of the Operations Research Society of China*, 3(2):207–229.

Ji, H., Liu, C., Shen, Z., and Xu, Y. (2010). Robust video denoising using low rank matrix completion. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1791–1798. IEEE.

Jianbo Shi and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Jiang, B., Ding, C., Luo, B., and Tang, J. (2013). Graph-Laplacian PCA: Closed-Form Solution and Robustness. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3492–3498.

Jin, C., Kakade, S. M., and Netrapalli, P. (2016). Provable efficient online matrix completion via non-convex stochastic gradient descent. *Advances in Neural Information Processing Systems*, 29:4520–4528.

Kent, A., Sweet, J., and Woodward, B. (2016). Iris community wavefield experiment in Oklahoma. Incorporated Research Institutions for Seismology. Dataset/Seismic Network.

Kilmer, M. E. and Martin, C. D. (2011). Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658.

Kim, D. and Fessler, J. A. (2016). Optimized first-order methods for smooth convex minimization. *Mathematical programming*, 159(1-2):81–107.

Kim, J.-H., Sim, J.-Y., and Kim, C.-S. (2015). Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Transactions on Image Processing*, 24(9):2658–2670.

Kopsinis, Y. and McLaughlin, S. (2009). Development of EMD-based denoising methods inspired by wavelet thresholding. *IEEE Transactions on Signal Processing*, 57(4):1351–1362.

Koren, Y. (2009). The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81(2009):1–10.

Kreimer, N. and Sacchi, M. D. (2012). A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation. *Geophysics*, 77:V113–V122.

Li, G. and Pong, T. K. (2015). Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460.

Li, H. and Lin, Z. (2015). Accelerated proximal gradient methods for nonconvex programming. *Advances in neural information processing systems*, 28:379–387.

Li, J.-Q., Rong, Z.-H., Chen, X., Yan, G.-Y., and You, Z.-H. (2017). Mcmda: Matrix completion for mirna-disease association prediction. *Oncotarget*, 8(13):21187.

Li, X.-R., Li, X.-M., Li, H.-L., and Cao, M.-Y. (2009). Rejecting Outliers Based on Correspondence Manifold. *Acta Automatica Sinica*, 35(1):17–22.

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

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., and Ma, Y. (2012). Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):171–184.

Liu, X., Wen, Z., and Zhang, Y. (2015). An efficient gauss–newton algorithm for symmetric low-rank product matrix approximations. *SIAM Journal on Optimization*, 25(3):1571–1608.

Liu, Y. and Fomel, S. (2013). Seismic data analysis using local time-frequency decomposition. *Geophysical Prospecting*, 61(3):516–525.

Liu, Y., Zheng, Y., Lu, J., Cao, J., and Rutkowski, L. (2019). Constrained quaternion-variable convex optimization: a quaternion-valued recurrent neural network approach. *IEEE transactions on neural networks and learning systems*, 31(3):1022–1035.

Lou, Y. and Yan, M. (2018). Fast l1–l2 minimization via a proximal operator. *Journal of Scientific Computing*, 74(2):767–785.

Lu, C., Yang, M., Luo, F., Wu, F.-X., Li, M., Pan, Y., Li, Y., and Wang, J. (2018). Prediction of lncrna–disease associations based on inductive matrix completion. *Bioinformatics*, 34(19):3357–3364.

Ma, S. and Aybat, N. S. (2018). Efficient optimization algorithms for robust principal component analysis and its variants. *Proceedings of the IEEE*, 106(8):1411–1426.

Meila, M. and Shi, J. (2001). Learning Segmentation by Random Walks. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 873–879. MIT Press.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152.

Nesterov, Y. (2013). Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161.

Parcollet, T., Zhang, Y., Morchid, M., Trabelsi, C., Linarès, G., De Mori, R., and Bengio, Y. (2018). Quaternion convolutional neural networks for end-to-end automatic speech recognition. *arXiv preprint arXiv:1806.07789*.

Qiao, T., Ren, J., Wang, Z., Zabalza, J., Sun, M., Zhao, H., Li, S., Benediktsson, J. A., Dai, Q., and Marshall, S. (2017). Effective denoising and classification of hyperspectral images using curvelet transform and singular spectrum analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 55(1):119–133.

Rauhut, H., Schneider, R., and Stojanac, Ž. (2017). Low rank tensor recovery via iterative hard thresholding. *Linear Algebra and its Applications*, 523:220–262.

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

Rubinstein, R., Zibulevsky, M., and Elad, M. (2010). Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58(3):1553–1564.

Sha, N., Yan, M., and Lin, Y. (2019). Efficient seismic denoising techniques using robust principal component analysis. In *SEG Technical Program Expanded Abstracts 2019*, pages 2543–2547. Society of Exploration Geophysicists.

Shen, Y., Xu, H., and Liu, X. (2019). An alternating minimization method for robust principal component analysis. *Optimization Methods and Software*, 34(6):1251–1276.

Sun, C., Zhang, Q., Wang, J., and Xie, J. (2014). Noise reduction based on robust principal component analysis. *Journal of Computational Information Systems*, 10(10):4403–4410.

Tan, S. and Huang, L. (2014). An efficient finite-difference method with high-order accuracy in both time and space domains for modelling scalar-wave propagation. *Geophysical Journal International*, 197(2):1250–1267.

Tao, M. and Yuan, X. (2011). Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81.

Tay, Y., Zhang, A., Tuan, L. A., Rao, J., Zhang, S., Wang, S., Fu, J., and Hui, S. C. (2019). Lightweight and efficient neural natural language processing with quaternion networks. *arXiv preprint arXiv:1906.04393*.

Trefethen, L. N. and Bau III, D. (1997). *Numerical linear algebra*, volume 50. Siam.

Wang, C., Wang, X., Li, Y., Xia, Z., and Zhang, C. (2018). Quaternion polar harmonic fourier moments for color images. *Information Sciences*, 450:141–156.

Wang, Y., Yin, W., and Zeng, J. (2019). Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63.

Weglein, A. B. (2016). Multiples: Signal or noise? *Geophysics*, 81:V283–V302.

Wen, F., Chu, L., Liu, P., and Qiu, R. C. (2018). A survey on nonconvex regularization-based sparse and low-rank recovery in signal processing, statistics, and machine learning. *IEEE Access*, 6:69883–69906.

Wen, F., Ying, R., Liu, P., and Truong, T.-K. (2019). Nonconvex regularized robust pca using the proximal block coordinate descent algorithm. *IEEE Transactions on Signal Processing*, 67(20):5402–5416.

Wen, Z., Yin, W., and Zhang, Y. (2012). Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361.

Wright, J., Ganesh, A., Rao, S., and Ma, Y. (2009). Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. *Coordinated Science Laboratory Report no. UILU-ENG-09-2210, DC-243*.

Ye, H.-S., Zhou, N.-R., and Gong, L.-H. (2020). Multi-image compression-encryption scheme based on quaternion discrete fractional hartley transform and improved pixel adaptive diffusion. *Signal Processing*, 175:107652.

Yu, S., Ma, J., Zhang, X., and Sacchi, M. D. (2015). Interpolation and denoising of highdimensional seismic data by learning a tight frame. *Geophysics*, 80:V119–V132.

Yuan, X. and Yang, J. (2009). Sparse and low-rank matrix decomposition via alternating direction methods. *preprint*, 12(2).

Yuan, X. and Yang, J. (2013). Sparse and low-rank matrix decomposition via alternating direction methods. *Pacific Journal of Optimization*, 9:167–180.

Zhang, C.-H. et al. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942.

Zhang, F. (1997). Quaternions and matrices of quaternions. *Linear algebra and its applications*, 251:21–57.

Zhigang Tang, Jun Yang, and Bingru Yang (2010). A new Outlier detection algorithm based on Manifold Learning. In *2010 Chinese Control and Decision Conference*, pages 452–457.

Zhou, M., Liu, Y., Long, Z., Chen, L., and Zhu, C. (2019). Tensor rank learning in cp decomposition via convolutional neural network. *Signal Processing: Image Communication*, 73:12–21.

Zhou, T. and Tao, D. (2011). GoDec: Randomized low-rank & sparse matrix decomposition in noisy case. In *International Conference on Machine Learning*, pages 30–40.

Zhou, T. and Tao, D. (2013). Greedy bilateral sketch, completion & smoothing. In *International Conference on Artificial Intelligence and Statistics*, pages 650– 658. JMLR. org.

Zhou, X., Yang, C., Zhao, H., and Yu, W. (2014). Low-rank modeling and its applications in image analysis. *ACM Computing Surveys (CSUR)*, 47(2):1–33.

Zhou, Z., Li, X., Wright, J., Candes, E., and Ma, Y. (2010a). Stable principal component pursuit. In *2010 IEEE international symposium on information theory*, pages 1518–1522. IEEE.

Zhou, Z., Li, X., Wright, J., Candes, E., and Ma, Y. (2010b). Stable principal component pursuit. In *IEEE International Symposium on Information Theory*, pages 1518–1522. IEEE.

Zhu, X., Xu, Y., Xu, H., and Chen, C. (2018). Quaternion convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–647.