Anomaly Detection: Sparse Representation for High Dimensional Data

The Thesis Submitted to the Electrical and Electronic Engineering Department

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Guangyu Zhou

Imperial College London

United Kingdom

I hereby declare that to the best of my knowledge, this thesis is my own work and all else sources of contents have been acknowledged and appropriately referenced.

Abstract

In this thesis, I investigated in three different anomaly aware sparse representation approaches.

The first approach focuses on algorithmic development for the low-rank matrix completion problem. It has been shown that in the ℓ_0-search for low-rank matrix completion, the singular points in the objective function are the major reasons for failures. While different methods have been proposed to handle singular points, rigorous analysis has shown that there is a need for further improvement. To address the singularity issue, we propose a new objective function that is continuous everywhere. The new objective function is a good approximation of the original objective function in the sense that in the limit, the lower level sets of the new objective function are the closure of those of the original objective function. We formulate the matrix completion problem as the minimization of the new objective function and design a quasi-Newton method to solve it. Simulations demonstrate that the new method achieves excellent numerical performance.

The second part discusses dictionary learning algorithms to solve the blind source separation (BSS) problem. For the proof of concepts, the focus is on the scenario where the number of mixtures is not less than that of sources. Based on the assumption that the sources are sparsely represented by some

dictionaries, we present a joint source separation and dictionary learning algorithm (SparseBSS) to separate the noise corrupted mixed sources with very little extra information. We also discuss the singularity issue in the dictionary learning process which is one major reason for algorithm failure. Finally, two approaches are presented to address the singularity issue.

The last approach focuses on algorithmic approaches to solve the robust face recognition problem where the test face image can be corrupted by arbitrary sparse noise. The standard approach is to formulate the problem as a sparse recovery problem and solve it using ℓ_1-minimization. As an alternative, the approximate message passing (AMP) algorithm had been tested but resulted in pessimistic results. The contribution of this part is to successfully solve the robust face recognition problem using the AMP framework. The recently developed adaptive damping technique has been adopted to address the issue that AMP normally only works well with Gaussian matrices. Statistical models are designed to capture the nature of the signal more authentically. Expectation maximization (EM) method has been used to learn the unknown hyper-parameters of the statistical model in an online fashion. Simulations demonstrate that our method achieves better recognition performance than the already impressive benchmark ℓ_1-minimization, is robust to the initial values of hyper-parameters, and exhibits low computational cost.

Acknowledgement

Firstly, I would like to express my sincere appreciation to my Ph.D. supervisor Dr. Wei Dai for his encouragement, expert guidance and meticulously support. He has very strong mathematical ability, rigorous logical thinking and creative talent, which drive me to the right track towards my Ph.D. Moreover, he was always very supportive on finding funding for his students, which helped me focus on my research and pass the very hard time during my Ph.D. I believe his positive effect will influence my life firmly.

I also want to thank Dr. Cong Ling and Dr. Moez Draief, both of them are Senior Lecturers in Electrical and Electronic Engineering Department at Imperial College London, for their valuable comments and in-depth guidance through each stage of my Ph.D. study.

I would also like to express my gratitude to Prof. Mark Plumbley from Surrey University and Dr. Cong Ling, who would like to be my Ph.D. external and internal examiners, respectively.

I am also indebted to my colleagues and friends in Imperial College London. Special thanks to Mr. Xiaochen Zhao and Mr. Yang Lu in our group for their kind help and priceless collaboration.

Last but not the least, I cannot express enough thanks my family. My parents always show their endless support spiritually and economically. Also, my

beloved wife, Jing Zhou, for your deep belief, patience and sacrifices. Without them, this thesis would not have been possible.

Contents

Abstract **5**

Acknowledgement **7**

List of Publications **13**

List of Figures **15**

List of Tables **19**

Abbreviations **21**

1 Introduction **25**

1.1 The Challenges of Large-scale Data Processing 25

1.2 From Compressive Sensing to Low Rank Matrices 27

1.3 Sparse Representations for Anomaly Detection 30

1.4 Organization of the Thesis 36

2 Low Rank Matrix Completion **37**

2.1 Introduction 37

2.2 Backgrounds 40

2.2.1 An Optimization Framework for ℓ_0-Search 40

2.2.2 Singularity Issue 42
2.3 Smoothed Objective Function 48
2.3.1 A Choice of Parameter ρ_is 50
2.3.2 Gradient of the Smoothed Objective Function 53
2.3.3 An Illustration of the Smoothed Function 54
2.4 Algorithm Implementation 55
2.4.1 Optimization Methods in Euclidean Space 55
2.4.2 Why Optimize on Grassmann Manifold 56
2.4.3 From Euclidean Space to Grassmann Manifold 59
2.4.4 BFGS Algorithm in Global Coordinates 64
2.4.5 BFGS Algorithm in Local Coordinates 64
2.5 Performance Study 67
2.6 Proofs 72
2.6.1 Analytical Results for Example 1: Minimizing f_u 72
2.6.2 Proof of Theorem 2.3.1 74
2.6.3 Proof of Proposition 2.3.3 76
2.6.4 Proof of Singular Values in Example 2.2.2 78
2.6.5 Proof of Lemma 2.4.2 78
2.6.6 Proof of Lemma 2.4.3 79

3 Blind Source Separation 81

3.1 Introduction 81
3.2 Background 84
3.3 Framework of Dictionary Learning Based BSS Problem 87
3.3.1 Separation with Dictionaries Known in Advance 87
3.3.2 Separation with Unknown Dictionaries 88

3.3.2.1 SparseBSS Algorithm Framework 88
3.3.2.2 Implementation Details of SparseBSS 92
3.3.3 Blind MMCA and Its Comparison to SparseBSS 94
3.4 Dictionary Learning and the Singularity Issue 96
3.4.1 Brief Introduction of Dictionary Learning Algorithms . . 96
3.4.2 Singularity Issue and Its Impacts 97
3.4.3 Regularized SimCO 100
3.4.4 Smoothed SimCO 100
3.4.5 Implementation of Smoothed SimCO 103
3.5 Algorithm Testing on Practical Applications 106
3.5.1 Empirical Tests for Smoothed SimCO 106
3.5.2 Algorithm Testing on BSS Problem 109
3.6 Conclusions 117

4 Robust Face Recognition 119

4.1 Introduction 119
4.2 Preliminary Research 123
4.2.1 Robust Face Recognition 123
4.2.2 AMP/GAMP for SRC 124
4.3 An AMP Based Method 126
4.3.1 Dual BG-GAMP 126
4.3.2 Adaptive Damping 127
4.3.3 Dual Expectation Maximization 129
4.4 Experiments 131
4.5 Conclusions 135

5 Conclusion and Future Work 137

References **141**

List of Publications

List of publications arising directly from this thesis

1. Guangyu Zhou, Xiaochen Zhao and Wei Dai. "Low Rank Matrix Completion: A Smoothed ℓ_0-Search," 50th Annual Allerton Conference on Communication, Control, and Computing, 2012.

2. Guangyu Zhou, Xiaochen Zhao (Co-first author), Wei Dai and Wenwu Wang. "Blind Source Separation Based on Dictionary Learning: A Singularity Aware Approach," In book Blind Source Separation: Advance in Theory, Algorithms and Applications. Published in 2014.

3. Guangyu Zhou, Xiaochen Zhao and Wei Dai, "Low-Rank Matrix Completion with Smoothed ℓ_0-Search," this journal paper is in preparation.

4. Guangyu Zhou and Wei Dai, "An Approximate Message Passing Algorithm for Robust Face Recognition," 24th European Signal Processing Conference (EUSIPCO), 2016.

Other publications

1. Xiaochen Zhao, Guangyu Zhou, Wenwu Wang and Wei Dai. "Weighted SimCO: A Novel Algorithm for Dictionary Update," Sensor Signal Processing for Defence (SSPD), 2012.

2. Xiaochen Zhao, Guangyu Zhou and Wei Dai. "Smoothed SimCO for Dictionary Learning: Handling The Singularity Issue," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013.

3. Xiaochen Zhao, Guangyu Zhou, Wei Dai, Tao Xu and Wenwu Wang. "Joint Image Separation and Dictionary Learning," 18th International Conference on Digital Signal Processing (DSP), 2013.

List of Figures

1.1.1 A simple example of the incomplete elements in user recommendation system. Each row contains the feedback scores for movies from one customer. Each column contains all the feedback scores for one particular movie from all the customer. The question marks indicate the incomplete feedback scores. 26

1.2.1 An illustration of the mathematical model of CS. Each block indicates one entry of the vectors/ matrix. For the sparse vector $\boldsymbol{x}$, the red blocks present its non-zero entries. These non-zero entries are associated with the corresponding columns (bounded in red) in the measurement matrix $\boldsymbol{A}$. 29

1.3.1 The model of sparse representations for anomaly detection. The observation matrix (left) can be decomposed into its sparse representation (middle) and anomalies (right). 32

1.3.2 An illustration of low rank matrix completion problem. The matrix on the left is the incomplete observations. Each block presents an entry of the matrix. The black blocks indicate the missing entries. The underlying matrix is assumed to be low rank. One is aiming to estimate the full low rank matrix (right) from its incomplete observations (left). 34

2.2.1 Illustration of the singularity issue and possible solutions. Consider Example 1. The contours of the original f_1, the regularized $f_{\mu,1}$ ($\mu = 0.1$), and the smoothed $\tilde{f}_1$ ($\rho = \frac{1}{6}$) are depicted in sub-figures (a), (c), and (d) respectively, where the blue circle and the red cross denote the initial point and the global optimum respectively. Sub-figure (b) illustrates how an infinitesimal gradient descent procedure gets trapped to the singular point when minimizing f. Sub-figures (c) and (d) give the intuitions on why regularization does not solve the singularity issue but the proposed smoothing technique does. 44

2.3.1 The modulation function $g_\rho(\lambda)$. 48

2.3.2 This is an alternative illustration of the singularity issue and the smoothed solution. Left: convex case. It is easy to find the optimal. Middle: singular case. The singular point prevents the searching path to optimal. Right: Our smooth solution. It lets the searching process pass through. 50

2.4.1 An illustration of Parallel transport $\bar{\boldsymbol{T}}(-\nabla f_k)$ of tangent vector $-\nabla f_k$ from tangent space $\mathcal{T}_k$ to $\mathcal{T}_{k+1}$ on Grassmann manifold $\mathcal{M}$, where $\triangle_k$ is the decent direction. (After [28]) 63

2.4.2 An illustration of tangent vectors in local coordinates on Grassmann manifold. In local coordinates, the tangent vector $\boldsymbol{D}_{f_k}$ is constant in both tangent spaces $\mathcal{T}_k$ and $\mathcal{T}_{k+1}$. (After [28]) 66

2.5.1 Performance improvement of SSE compared with SET. 68

2.5.2 Performance comparison: noiseless case. 69

2.5.3 Performance comparison: noisy case. 70

2.5.4 Performance comparison: large matrix case. 70

3.5.1 The performance comparison. 107

3.5.2 Noiseless case: success rate. 108

3.5.3 Two speech sources and the corresponding noisy mixtures (20 dB Gaussian noise). 110

3.5.4 Relation of the parameter λ to the estimation error of the mixing matrix under different noise levels. The signal-to-noise ratio (SNR) is defined as $\rho = 10\log_{10}\|\boldsymbol{AS}\|_F^2 / \|\boldsymbol{V}\|_F^2$ dB. 113

3.5.5 Two classic images, *Lena* and *Boat* were selected as the source images, which are shown in (a). The mixtures are shown in (b). The separation results are shown in (c)-(f). We compared SparseBSS with other benchmark algorithms: FastICA [42], GMCA [10] and BMMCA [3]. We set the overlap percentage equal to 50% for both BMMCA and SparseBSS. The recovered source images by the SparseBSS tend to be less blurred as compared to the other three algorithms. 114

3.5.6 Compare the performance of estimating the mixing matrix for all the methods in different noise standard deviation σs. In this experiment, σ varies from 2 to 20. The performance of GMCA is better than that of FastICA. The curve for BMMCA is not available as the setting for the parameters is too sophisticated and inconsistent for different σ to obtain a good result. SparseBSS outperforms the compared algorithms. 115

3.5.7 The two source images *Lena* and *Texture* are shown in (a). The separation results are shown in (b) and (c). The comparison results demonstrate the importance of the singularity aware process. 116

4.2.1 Overview of the SRC framework. The test image (left), which is occluded by a sunglasses. It is equal to the sparse linear combinations of the training images (middle) plus error image (right). The sparse coefficient (red) indicate the corresponding true identity, which is bounded in a red box in the training images (middle). This graph is only for demonstration. There are hundreds or even thousands of training images in the test. . 122

4.4.1 Recognition rates for different algorithms under different fractions of noise corruptions. 134

4.4.2 Comparison of our algorithm with DALM under different fractions of corrupted entries c = 60%, 70%, 80%. Our method: red lines. DALM algorithm: blue lines. 135

List of Tables

3.1 Separation performance of the SparseBSS algorithm as compared to FastICA and QJADE. The proposed SparseBSS algorithm performs better than the benchmark algorithms. Table 3.1a. For the same algorithm, the $\triangle SDR$ and $\triangle SIR$ are the same in noiseless case. The $\triangle SDRs$ and $\triangle SIRs$ for all the tested algorithms are large and similar, suggesting that all the compared algorithms perform very well. The artifact introduced by SparseBSS is small as its $\triangle SAR$ is positive. Table 3.1b. In the presence of noise with SNR = 20 dB, SparseBSS excels the other algorithms in $\triangle SDR$, $\triangle SIR$ and $\triangle SAR$. One interesting phenomenon is that the $\triangle SDRs$ are much smaller than those in the noiseless case, implying that the distortion introduced by the noise is trivial. However, SparseBSS still has better performance. 111

3.2 Achieved MSEs of the algorithms in a noiseless case. 113

4.1 Definitions: the input and output probabilistic relationships $P_{X|Y}$ and $P_{Z|Y}$. The associated BG input scalar estimation function and the AWGN output scalar estimation function [72][64]. . 128

Abbreviations

AD Adaptively Damping

ADMiRA Atomic Decomposition for Minimum Rank Approximation

ADMM Alternating Direction Method of Multipliers

AMP Approximate Message Passing

AWGN Additive White Gaussian Noise

BFGS Broyden-Fletcher-Goldfarb-Shanno

BG Bernoulli-Gaussian

BMMCA Blind Morphological Component Analysis

BSS Blind Source Separation

CG Conjugate Gradient

CS Compressive Sensing

DCT Discrete Cosine Transform

EM Expectation Maximization

FastICA Fast Independent Component Analysis

FOCUSS FOCal Underdetermined System Solver

GAMP Generalized Approximate Message Passing

GM Gaussian Mixture

GMCA Generalized Morphological Component Analysis

ICA Independent Component Analysis

ILS-DLA Iterative Least Squares based Dictionary Learning Algorithms

JADE Joint Approximate Diagonalisation of Eigen-matrices

JPEG Joint Photographic Experts Group

K-SVD K-means clustering Singular Value Decomposition

MCA Morphological Component Analysis

MMCA Multichannel Morphological Component Analysis

MOD Method of Optimal Directions

MSE Mean Squared Error

NMF Non-negative Matrix Factorization

OMP Orthogonal Matching Pursuit

PALM Primal Augmented Lagrangian Method

PCA Principle Component Analysis

PDIPA Primal-Dual Interior-Point Algorithm

PF PowerFactorization

PSNR Peak Signal-to-Noise Ratio

RIP Restricted Isometry Property

RLS-DLA Recursive Least Squares based Dictionary Learning Algorithms

RTRMC Riemannian Trust Region method for low-rank Matrix Completion

SAR Source-to-Artifact Ratio

SCA Sparse Component Analysis

SCI Sparsity Concentration Index

SDR Source-to-Distortion Ratio

SET Subspace Evolution and Transfer

SimCO Simultaneous Codeword Optimization

SIR Source-to-Interference Ratio

SNR Signal-to-Noise Ratio

SP Subspace Pursuit

SparseBSS Sparse Blind Source Separation

SPICA Sparse Independent Component Analysis

SRC Sparse Representation based Classification

SSE Smoothed Subspace Evolution

SVD Singular Value Decomposition

SwAMP Swept Approximate Message Passing

TNIPM Truncated Newton Interior-Point Method

Chapter 1

Introduction

Nowadays, we capture, transmit, analyse and store more and more data. Big data processing has played an important role from research to our daily life, e.g., signal processing, machine learning, computer vision, user recommendation system, etc. For example, book retailers can predict preferences of their costumers and recommend books to them by using the data that collected from their costumers.

1.1 The Challenges of Large-scale Data Processing

Large-scale data brings more useful information. It offers tremendous insight for us. On the contrary, the cost of big data processing increases with the increasing of the dimension of the data. It is one of the phenomena that is referred as the curse of dimensionality. There are more fundamental phenomena referred to the curse of dimensionality in other fields, such as data acquisition, data sampling, etc. However, we are particularly interested in the big data

Figure 1.1.1: A simple example of the incomplete elements in user recommendation system. Each row contains the feedback scores for movies from one customer. Each column contains all the feedback scores for one particular movie from all the customer. The question marks indicate the incomplete feedback scores.

processing in this thesis.

We are facing the challenges in leveraging more useful data in real time applications. For example, in statistical anomaly detection techniques, we may need more parameters and the statistical model is complex, when the dimension of the data is high. Another problem of handling high dimensional data is that the data is often incomplete or even corrupted as anomalies. For example, in the user recommendation system, the vendors collect the feedback scores of their products (such as movies, books, etc.) from their customers. However, in practice, the data always has "missing" elements. In other words, we are not able to collect the feedbacks from all the customers for all the products. The aim of a recommendation system is to predict the missing entries from the incomplete observations. A simple example is shown in Fig. 1.1.1. In face recognition problem, some of the collected face images are under extreme lighting conditions, corrupted by the noise or occlusions, etc.

1.2 From Compressive Sensing to Low Rank Matrices

One may utilise the sparsity property to handle large-scale data. In the high dimensional data, there exists correlation and redundancy. Those properties make the data more compressible/sparse. A very common example is the image compression standard JPEG, which is an abbreviation for the Joint Photographic Experts Group. Under this standard, the raw image can be compressed by using Discrete Cosine Transform (DCT) with a small number of significant non-zero coefficients while the majority of the coefficients are all zero or close to zero. Moreover, a higher dimension may bring more sparse feature. For example, under the same scene of a video, the two adjacent frames are highly correlated. Hence, we do not have to keep the redundant information between those correlated frames. The compression rate of a video can be made much higher than a single image. We benefit from the sparsity to handle the high dimensional data. Typically, there are two types of sparse representations: one is the signal is sparse under the some fixed basis, such as DCT, standard Gaussian; the other is the that the basis is not fixed, for example, dictionary learning. A more detailed discussion about the fixed basis case is presented in this Subsection.

Compressive sensing (CS) was first introduced by Candes, Tao [15] and Donoho [25]. In the past decades, it has been studied and widely applied to many fields, such as Magnetic Resonance Imaging (MRI), Single-pixel camera, face recognition, etc. The acquired signal is sampled under a very low measurement rate, which is below the Nyquist sampling rate, while still preserving

a good quality of the signal. The mathematical model is as follows,

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x},$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the known measurement matrix, $\boldsymbol{x} \in \mathbb{R}^n$ is the unknown sparse signal, and observation vector $\boldsymbol{y} \in \mathbb{R}^m$ $(m < n)$. CS targets at finding a sparse solution $\boldsymbol{x}$ from its measurement $\boldsymbol{y}$. Then, this problem is presented as,

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{1.2.1}$$

where $\|\cdot\|_0$ is the ℓ_0 pseudo-norm, which denotes the number of non-zero elements of $(\cdot)$. An illustration of this problem is presented in Fig. 1.2.1. However, this ℓ_0 problem is NP-hard. A relaxed convex optimization approach to the problem (1.2.1) was proposed,

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{1.2.2}$$

where $\|\cdot\|_1$ is the ℓ_1-norm, which is defined as the summation of the absolute value of each element in of vector. The solution of ℓ_1 problem gives the exact solution to the ℓ_0 problem in (1.2.1) if the measurement matrix satisfies some properties, such as Restricted Isometry Property (RIP) [15].

The matrix rank minimization problem is closely related to the CS problem. It assumes the signal has a low rank structure, which is known as spectral sparsity. The sparsity is in the eigen space rather than the signal itself. Let $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ be an unknown matrix. The singular value decomposition (SVD) of $\boldsymbol{X}$ gives,

$$SVD(\boldsymbol{X}) = \sum_i \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T,$$

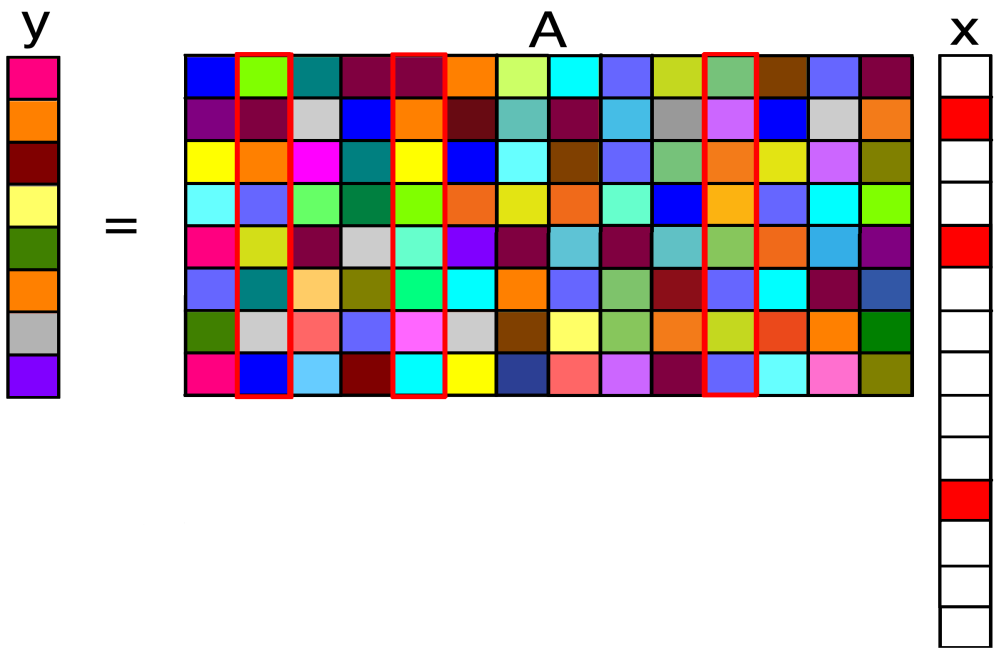


Figure 1.2.1: An illustration of the mathematical model of CS. Each block indicates one entry of the vectors/ matrix. For the sparse vector $\boldsymbol{x}$, the red blocks present its non-zero entries. These non-zero entries are associated with the corresponding columns (bounded in red) in the measurement matrix $\boldsymbol{A}$.

where σ_is are the singular values in descending order and $\boldsymbol{u}_i$s and $\boldsymbol{v}_i$s are the corresponding left and right singular vectors, respectively. The rank is the number of non-zero σ_is. In other words, the matrix rank minimization problem is aiming to find the minimization of the ℓ_0 pseudo-norm of the singular value vector.

The matrix rank minimization problem is defined as follows,

$$\min_{\boldsymbol{X}} \operatorname{rank}(\boldsymbol{X}) \quad s.t.\ \mathcal{P}(\boldsymbol{X}) = \boldsymbol{b}, \tag{1.2.3}$$

where the linear map $\mathcal{P} : \mathbb{R}^{m\times n} \to \mathbb{R}^p$. Similar with the problem (1.2.1), this problem is also NP-hard. For the matrix minimization problem, we consider the following convex relaxation,

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_* \quad s.t.\ \mathcal{P}(\boldsymbol{X}) = \boldsymbol{b},$$

where $\|\cdot\|_*$ is the nuclear norm, which denotes the sum of the singular values.

1.3 Sparse Representations for Anomaly Detection

For statistical anomaly detection techniques, the major issue is that we may need many parameters and the statistical model is complex when the dimension of the data is high. The spectral anomaly detection techniques are based on the assumption that data can be projected into a lower subspace, which the abnormal data differ from the normal data. One effective way of solving the high dimensional problem is to learn the subspace structure of data,

which leads to reducing the dimension of the data. Some dimension reduction methods have been proposed, e.g. Principle Component Analysis (PCA) [2], Non-negative Matrix Factorization (NMF) [52], etc. Low rank models (spectral sparse) capture the intrinsic structure of the high dimensional data. The low rank representation of the data can help us in saving storage space and reducing computational complexity. For example, PCA is a low rank matrix approximation method that captures the hidden important aspects of the data. The low rank matrix approximation of $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ can be written as follows

$$\min \left\| \boldsymbol{X} - \hat{\boldsymbol{X}} \right\|_F^2, \text{ s.t. } \operatorname{rank}(\hat{\boldsymbol{X}}) \leq r$$

where $\hat{\boldsymbol{X}} \in \mathbb{R}^{m \times n}$ is the low rank approximation, $r < \min(m, n)$ and $\|\cdot\|_F$ denotes the Frobenius norm.

One can use SVD to find the rank r approximation $\boldsymbol{L}$ of the data matrix $\boldsymbol{X}$ by

$$\boldsymbol{L} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T,$$

where $\boldsymbol{L}$ is the optimal solution of the low rank matrix completion problem. Then, the dimension of the matrix can be significantly reduced if the value of the rank r is low.

Most interestingly, one can detect [6, 62] or predict changes from data sets by using statistic tools. In many industry and business, fast and precise detection or prediction play a very important role. Unfortunately, most of the data analysis are based on parametric statistical models [50]. High dimensional data sets increase the number of unknown parameters and computational complexity. Laan and Rose [50] argue that the next generation of statisticians must build new tools for massive data sets. However, for some particular case, not

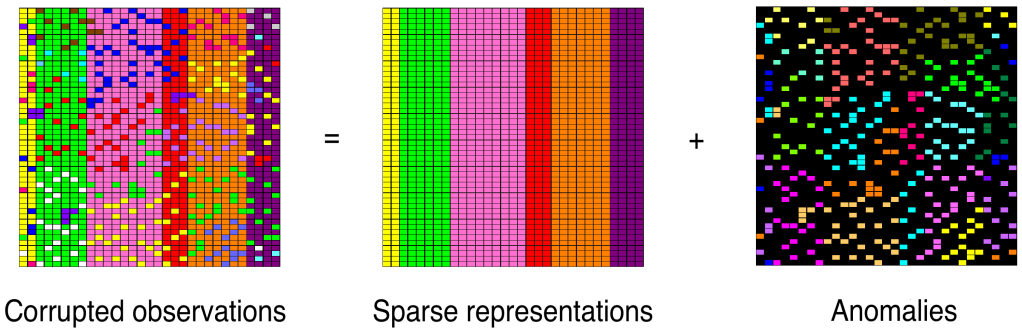


Figure 1.3.1: The model of sparse representations for anomaly detection. The observation matrix (left) can be decomposed into its sparse representation (middle) and anomalies (right).

all of the data sets are useful and some of them are even redundant. If it is possible to distill (i.e. random measurements) useful information from big data, and then one could apply statistical tools. An efficient and robust statistical approach is needed to process the sparse signal.

In this thesis, we assume that the observation is the summation of the under-lying sparse representation and the anomalies. The model of sparse representations for anomaly detection are shown in Fig. 1.3.1. The observation matrix can be decomposed into its sparse representation and anomalies. We aim to acquire the sparse representation of the observation and separate the anomalies. We shall adopt the state of the art sparse representation techniques, such as matrix rank minimization, CS, etc., leveraging the anomalies in real-time applications.

Low rank matrix completion is a typical application of the matrix rank minimization. It has been receiving considerable interest in recent years due to the wide applicability, ranging from collaborative filtering to anomaly detection. It aims at recovering the underlying low rank matrix from its incomplete

observations. A very famous example of low rank matrix completion is the Netflix problem [1]. Netflix is a company that provides online streaming service. Netflix collects the user preferences, e.g., feedbacks, browsing histories, etc. Then, let user recommendation system to predict the potential user interests. The users are more likely to purchase the personalised recommendations if the predictions are more accurate. In 2009, in order to improve the accuracy of their predictions, Netflix established one million dollars grand prize to award the person or team who achieved 10% improvement compared with their own recommendation system on the same train data. It is noteworthy that the low rank structure is determined by the several key factors. From the point of view of movies, it can be affected by the themes, regions, directors, actresses/actors, etc. For the users, it is determined by the ages, genders, nationalities, etc. For the low-rank matrix completion problem, we are aiming to find a complete matrix with the lowest rank from its subset of entries. An illustration of this problem is shown in Fig. 1.3.2.

In [51], Lakhina et al. firstly introduced PCA technique to the off-line traffic matrix anomaly detection in 2004. The authors shows the correlations of the time series of the traffic data at link level, e.g., there are only 3 or 4 principal components in the time series of a more than 40 links network. Then one can separate anomalies from the normal traffic data by using SVD. Low rank matrix completion technique can also be applied to traffic matrix estimation in the network. A traffic matrix is a measurement matrix that each entry indicates the traffic volume at a link at one time interval. The traffic matrix is low rank [51]. Monitoring all the link data in a large networks can be very costly. Instead, one can measure different subsets of link volume at different time intervals. Then, we can use the partially observed traffic matrix

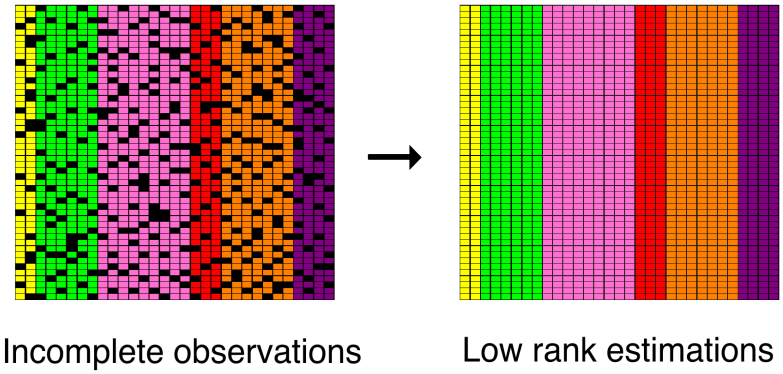


Figure 1.3.2: An illustration of low rank matrix completion problem. The matrix on the left is the incomplete observations. Each block presents an entry of the matrix. The black blocks indicate the missing entries. The underlying matrix is assumed to be low rank. One is aiming to estimate the full low rank matrix (right) from its incomplete observations (left).

to estimate the missing entries.

Pedestrian Detection is a good example of anomaly detection. Given a sequence of video of the public area, even in one frame, pedestrians are always sparse (as anomalies) in contrast to the background (e.g. buildings, trees, etc.). If we vectorized each frame, the whole video can be converted into a matrix, with each frame as a column or row vector. Then, one can assume that this matrix is of low rank. In practice, the rank is larger than one, because of the movement of the pedestrians, illumination changes, etc. Finally, the problem turns to be an optimization of finding a low rank sparse representation of the obtained matrix. Hence, low rank representation is a very interesting problem and one can benefit it from storage reduction to anomaly separation. A recent paper [78] concludes that the traditional methods, e.g., PCA, are sensitive to the outliers. There are some robust algorithms, but they are typically complex. They propose to guess the locations of the outliers and then learn the location

of the outlier and its low rank representation iteratively.

Blind source separation has been investigated during the last two decades, many algorithms have been developed and applied in a wide range of applications including biomedical engineering, medical imaging, speech processing, astronomical imaging and communication systems. One of the most famous phenomena is the cocktail party problem. Suppose there is a cocktail party. We are able to focus on the voice that we are particularly interested even there are lots of other voices or sounds, e.g., someone speaks simultaneously in your neighbors. However, blind source separation remains challenging for a machine. The aim of blind source separation is to isolate on one source while filtering out the other sources or noises in the background as anomalies. It is not limited to separate mixed talks. There are several existing methods, such as PCA, Independent Component Analysis (ICA), etc. In this thesis, we focus on dictionary learning approach for blind source separation. It is worth to note that the dictionary (basis) is not fixed in dictionary learning approaches.

In face recognition problem, it involves in identifying the target from a large set of facial images. In practice, the target image might even be corrupted by the noise or occlusions (sunglasses, scarfs) as anomalies. It makes the identification of a facial image more difficult. Wright et al. [77] propose a robust face recognition method that assumes the testing image can be represented by the sparse linear combinations of the images in the training dataset. The noise or occlusion is also sparse. A review paper [81] compared the ℓ_1-minimization benchmarks for the robust face recognition framework in terms of recognition accuracy and speed. Unfortunately, the original Approximate Message Passing (AMP) algorithm has pessimistic results. Moreover, if the faces in the dataset are not well aligned, it will affect the recognition rate. The image registration

is beyond the scope of this thesis. We assume the test facial images are well aligned with the training images in face recognition.

1.4 Organization of the Thesis

The remainder of this thesis is organised as follows. In Chapter 2, we focus on the low rank matrix completion problem. Chapter 3 describes the BSS problem and presents a dictionary learning approach to address this problem. A detailed discussion about robust face recognition is present in Chapter 4. Finally, we concludes this thesis and identifies future works in the last Chapter.

Chapter 2

Low Rank Matrix Completion

2.1 Introduction

This Chapter focuses on an ℓ_0-search for low-rank matrix completion. Here, the ℓ_0-search is referred to the search process that the variable is constrained in the set of low-rank matrices. As discussed in detail in [19], the ℓ_0-search for low-rank matrix completion is significantly different from heuristic algorithms that are used for compressive sensing. Methods that are particularly designed for matrix completion have to be developed. Early examples include the PowerFactorization (PF) [39] and the OptSpace [47] algorithms. More recently, in [21], the authors discovered that the major technical difficulty of the ℓ_0-search comes from the fact that the objective function is not continuous. The singular points create the so called barriers to stop an optimization process from converging to the global optimum. To address this issue, a method involving singularity detection and jumping was developed in [21], a geometric objective function that is used to replace the original objective function was proposed in [19], and a regularized objective function was studied in [11]. A more detailed

description and analysis of these methods is presented in Section 2.2.

However, this problem is NP-hard [40]. In order to solve this problem effectively, one can use its convex relaxation, e.g., nuclear norm minimization. Scores of methods have been proposed for low-rank matrix completion. Many of them are based on the similarities between compressive sensing reconstruction and low-rank matrix completion, both of which involve solving under-deterministic linear inverse problems with sparsity constraints. Following the popular ℓ_1-minimization approach for compressive sensing, a convex relaxation of the low-rank matrix completion problem, nuclear norm minimization [16, 14, 71] has been successfully applied. The nuclear norm minimization is finding the minimum nuclear norm of a complete matrix via given observed entries. Under certain conditions, the nuclear norm minimization recovers the same unique solution as the rank minimization. At the same time, greedy algorithms for low-rank matrix completion [53, 56], as counterparts of those for compressive sensing, have also been developed.

In this Chapter, we proposed a new method, termed as smoothed subspace evolution (SSE), to solve the so called singular point issue. In SSE, a new objective function is proposed and it is continuous everywhere. In particular, a multiplication term is introduced to smooth the discontinuous objective function in [21]. In contrast, an addition (regularized) term is proposed in [11]. The advantage of the proposed approach is presented in Section 2.2 and 2.3. The new approach is implemented based on quasi-Newton method, which has super linear convergence and is easy to compute. Numerical results demonstrate that the proposed method outperforms all other benchmark algorithms.

The main contributions of this Chapter include:

- We study different mechanisms to handle the singularity issue in ℓ_0-

search, and rigorously show how the regularization technique may fail. The regularization term solves the discontinuous problem of the objective function. It will force the searching process away from the singular points, while it will generate a local minimum at the neighbor of the singular points as a side effect.

- To address the singularity issue, we propose a continuous objective function to replace the original objective function. We prove that the proposed objective function is a good approximation of the original one. In the limit, it differs from the original one only at the singular points, and its lower level sets are the closure of those of the original one.

- A quasi-Newton method is implemented to solve the related optimization problem. In order to reduce the computational complexity of the optimization process, a local coordinate based quasi-Newton approach is introduced and discussed.

- Simulations demonstrate that the proposed method achieves excellent numerical performance. The SSE also has the best performance among all tested algorithms even when the number of observations is close to the oracle rate[1].

The remainder of this Chapter is organized as follows. In Section 2.2.1, we formally introduce the ℓ_0-search for low-rank matrix completion. Section 2.2.2 is devoted to describing the singularity issue and analyzing several techniques developed to cope with this issue. Our solution to the singularity issue is presented in Section 2.3. The considerations for implementation of the proposed

[1]The oracle rate refers to the minimum number of observations that is needed to determine a matrix. For a $m \times n$ rank r matrix, its oracle rate is $r(m+n-r)$.

method are discussed in Section 2.4. The empirical performance improvement is demonstrated in Section 2.5.

2.2 Backgrounds

2.2.1 An Optimization Framework for ℓ_0-Search

The ℓ_0-search of low-rank matrix completion can be formulated as follows. Following the common approach [53, 39, 47, 21], assume that the rank r is given.[2] Let $\Omega \subset [m] \times [n]$ be the set of indices of the observed entries, where $[K] = \{1, 2, \cdots, K\}$. Define the projection operator $\mathcal{P}_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ by

$$\mathcal{P}_\Omega(\boldsymbol{X}) \mapsto \boldsymbol{X}_\Omega, \text{ where } (\boldsymbol{X}_\Omega)_{i,j} = \begin{cases} \boldsymbol{X}_{i,j} & \text{if } (i,j) \in \Omega \\ 0 & \text{if } (i,j) \notin \Omega. \end{cases}$$

The task is to find a rank-r matrix $\boldsymbol{X}'$ that is consistent with the observation matrix $\boldsymbol{X}_\Omega$, i.e., given Ω and $\boldsymbol{X}_\Omega$,

$$\begin{aligned} &\text{Find a } \boldsymbol{X}' \text{ s.t.} \\ &\operatorname{rank}(\boldsymbol{X}') \le r \text{ and } \mathcal{P}_\Omega(\boldsymbol{X}') = \mathcal{P}_\Omega(\boldsymbol{X}) = \boldsymbol{X}_\Omega. \end{aligned} \tag{2.2.1}$$

The problem 2.2.1 can be formulated as an optimization problem. Define $\mathcal{U}_{m,r} = \left\{\boldsymbol{U} \in \mathbb{R}^{m\times r} : \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_r\right\}$ and the function $f : \mathcal{U}_{m,r} \to \mathbb{R}$ as

$$f(\boldsymbol{U}) = \min_{\boldsymbol{W}\in\mathbb{R}^{r\times n}} \|\boldsymbol{X}_\Omega - \mathcal{P}_\Omega(\boldsymbol{U}\boldsymbol{W})\|_F^2. \tag{2.2.2}$$

[2] In practice, one may try to sequentially guess a rank bound until a satisfactory solution has been found.

This function measures the consistency between the matrix $\boldsymbol{U}$ and the observation $\boldsymbol{X}_\Omega$. In particular, if $f\left(\hat{\boldsymbol{U}}\right) = 0$ for some $\hat{\boldsymbol{U}} \in \mathcal{U}_{m,r}$ and let $\hat{\boldsymbol{W}} \in \mathbb{R}^{r\times n}$ be the matrix solving the least squares problem in evaluating $f\left(\hat{\boldsymbol{U}}\right)$, then the rank-$r$ matrix $\hat{\boldsymbol{X}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{W}}$ is a solution of the problem 2.2.1. Hence, the ℓ_0-search problem can be formulated as

$$\min_{\boldsymbol{U}\in\mathcal{U}_{m,r}} \; f\left(\boldsymbol{U}\right) = 0. \tag{2.2.3}$$

Remark 2.2.1. A detailed discussion on the necessity to have the constraint $\boldsymbol{U} \in \mathcal{U}_{m,r}$ is given in Section 2.4.2.

The objective function (2.2.2) can be decomposed as a summation of atomic functions [21]. Let Ω_i be the index set of the observed entries in the i^{th} column, i.e., $\Omega_i = \{k : \; (k,i) \in \Omega\}$. Let $\boldsymbol{x}_{\Omega_i}$ be the i^{th} column of $\boldsymbol{X}_\Omega$, and $\boldsymbol{w}_i$ be the i^{th} column of $\boldsymbol{W}$. Define $\boldsymbol{U}_{\Omega_i} \in \mathbb{R}^{m\times r}$ as $\left(\boldsymbol{U}_{\Omega_i}\right)_{k,\ell} = \boldsymbol{U}_{k,\ell}$ if $k \in \Omega_j$ and $\left(\boldsymbol{U}_{\Omega_i}\right)_{k,\ell} = 0$ if $k \notin \Omega_i$. Then it can be verified that

$$f\left(\boldsymbol{U}\right) = \sum_{i=1}^{n} \underbrace{\min_{\boldsymbol{w}_i} \; \left\|\boldsymbol{x}_{\Omega_i} - \boldsymbol{U}_{\Omega_i}\boldsymbol{w}_i\right\|_2^2}_{f_i(\boldsymbol{U})} = \sum_{i=1}^{n} f_i\left(\boldsymbol{U}\right). \tag{2.2.4}$$

Each atomic function f_i involves a least squares problem. The optimal $\boldsymbol{w}_i$ for a given $\boldsymbol{U}$ has a closed form

$$\boldsymbol{w}_i\left(\boldsymbol{U}\right) = \boldsymbol{U}_{\Omega_i}^{\dagger}\boldsymbol{x}_{\Omega_i}, \tag{2.2.5}$$

where the superscript † denotes the pseudo-inverse. It is noteworthy that when $\boldsymbol{U}_{\Omega_i}$ is column-rank deficient, the optimal $\boldsymbol{w}_i\left(\boldsymbol{U}\right)$ is not unique[3] but the atomic

[3]Equation (2.2.5) gives the minimum ℓ_2-norm solution when $\boldsymbol{U}_{\Omega_i}$ is column-rank deficient.

function $f_i(\boldsymbol{U})$ takes a unique value. As a result, each atomic function can be written as

$$f_i(\boldsymbol{U}) = \left\| \boldsymbol{x}_{\Omega_i} - \boldsymbol{U}_{\Omega_i} \boldsymbol{U}_{\Omega_i}^{\dagger} \boldsymbol{x}_{\Omega_i} \right\|_2^2. \tag{2.2.6}$$

We then make the following assumption.

Assumption 2.2.2. *Let $q_i = |\Omega_i|$. We assume that $q_i > r$ for all $i \in [n]$.*[4]

The assumption is motivated by the fact that the observed column does not provide much information when $q_i \leq r$. Suppose that $q_i \leq r$ for some $i \in [n]$. A randomly generated $\boldsymbol{U}$ from the uniform distribution on $\mathcal{U}_{m,r}$ will have full column rank and give $f_i(\boldsymbol{U}_{\Omega_i}) = 0$ with probability one [22].

With this assumption, we formally define singular points. The motivation is that $\boldsymbol{U}_{\Omega_i}^{\dagger}$, as a function of $\boldsymbol{U}_{\Omega_i}$, is differentiable if and only if $\boldsymbol{U}_{\Omega_i}$ has full column rank.

Definition 2.2.3. A matrix $\boldsymbol{U} \in \mathcal{U}_{m,r}$ is a singular point of an atomic function f_i in 2.2.6 if $\boldsymbol{U}_{\Omega_i}$ is column-rank deficient. It is a singular point of the overall function f if there exists an $i \in [n]$ such that $\boldsymbol{U}_{\Omega_i}$ is column-rank deficient.

An illustration of the singular points is given in the next Subsection.

2.2.2 Singularity Issue

Optimization methods, for example, the gradient descent method, can be applied to solve the optimization problem (2.2.3). However, the optimization procedure may not converge to a global minimizer satisfying $f = 0$. Numerical experiments in [21] show that the optimization procedure may be trapped

[4] If the observation matrix $\boldsymbol{X}_{\Omega}$ contains columns with $q_i \leq r$, then we simply delete the columns corresponding to $q_i \leq r$ and use the resulting matrix for completion.

to singular points. The following example is designed to demonstrate this phenomenon.

Example 1. With slight abuse of notations, consider an incomplete rank-one matrix

$$\boldsymbol{X}_{\Omega} = \begin{bmatrix} ? & 1 \\ 1 & ? \\ 1 & 1 \end{bmatrix},$$

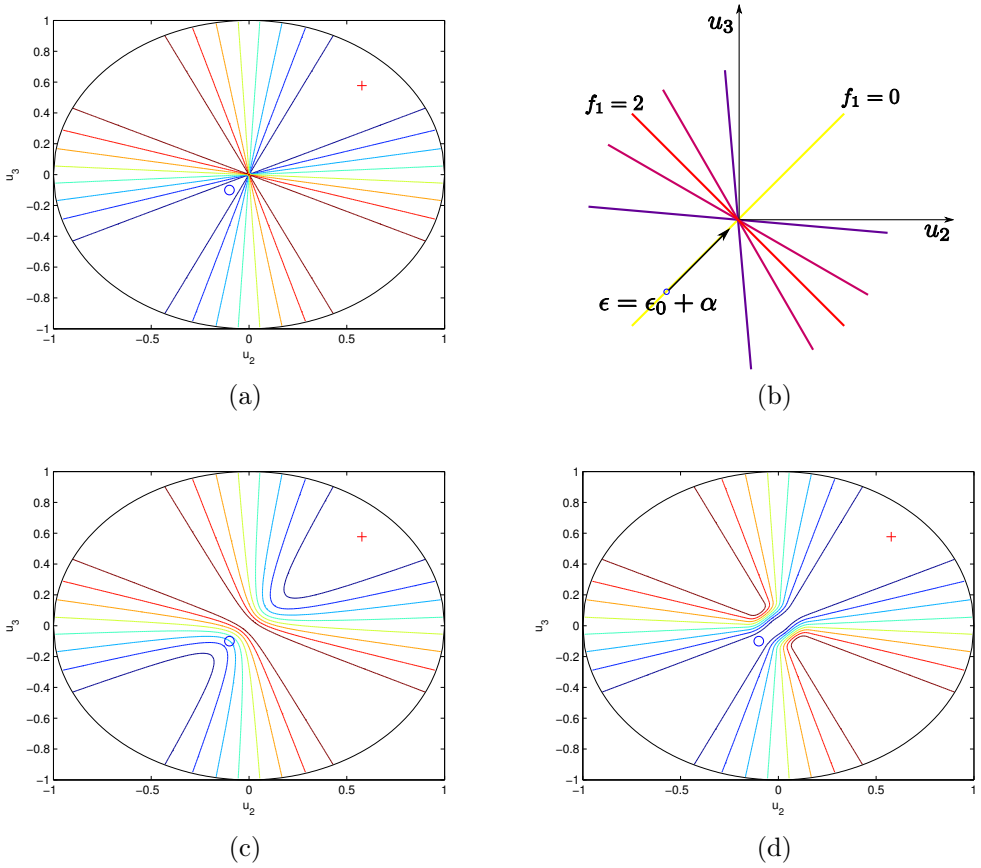
where the question marks denote the missing entries. Based on the information that $r = 1$, the solution of the corresponding matrix completion problem is the all-one matrix, i.e., $\boldsymbol{X}_{i,j} = 1$, $\forall i, j$. Formulate the completion problem as the optimization problem in (2.2.3). There are two solutions, which is given by $\boldsymbol{U}^* = \pm\frac{1}{\sqrt{3}}[1,1,1]^T$. In this case, we only focus on one of the solutions $\boldsymbol{U}^* = \frac{1}{\sqrt{3}}[1,1,1]^T$.

We shall show that the objective function (2.2.2) is discontinuous. Recall the decomposition in (2.2.4). Consider a vector $\boldsymbol{U}$ in $\mathcal{U}_{3,1}$ parametrized by ϵ: $\boldsymbol{U}(\epsilon) = \left[\sqrt{1-2\epsilon^2}, \epsilon, \epsilon\right]^T$ where $\epsilon \in \left[-1/\sqrt{2}, 1/\sqrt{2}\right]$. The optimal ϵ is given by $\epsilon^* = 1/\sqrt{3}$. For a given $\boldsymbol{U}(\epsilon)$, let $w_1(\epsilon)$ be the solution of the least squares problem in evaluating $f_1(\epsilon)$. Then,

$$w_1(\epsilon) = \begin{cases} \frac{1}{\epsilon},\ f_1(\boldsymbol{U}(\epsilon)) = 0 & \text{if } \epsilon \neq 0, \\ 0,\ f_1(\boldsymbol{U}(\epsilon)) = 2 & \text{if } \epsilon = 0. \end{cases}$$

Hence, the atomic function f_1, as well as the overall objective function f, is discontinuous at the point $\boldsymbol{U}(0)$.

It has been observed [21] that a gradient descent algorithm that minimizes f in Eq. 2.2.4 may be trapped in the neighborhood of singular points.



(a) (b)

(c) (d)

Figure 2.2.1: Illustration of the singularity issue and possible solutions. Consider Example 1. The contours of the original f_1, the regularized $f_{\mu,1}$ ($\mu = 0.1$), and the smoothed $\tilde{f}_1$ ($\rho = \frac{1}{6}$) are depicted in sub-figures (a), (c), and (d) respectively, where the blue circle and the red cross denote the initial point and the global optimum respectively. Sub-figure (b) illustrates how an infinitesimal gradient descent procedure gets trapped to the singular point when minimizing f. Sub-figures (c) and (d) give the intuitions on why regularization does not solve the singularity issue but the proposed smoothing technique does.

In particular, it can be verified that $\forall \epsilon \in \left(-\frac{1}{\sqrt{3}}, 0\right)$, $\frac{d}{d\epsilon} f\left(\boldsymbol{U}\left(\epsilon\right)\right) < 0$ (see Subsection 2.6.1 for the proof). Consider an *infinitesimal* gradient-descent search process (the real gradient-descent search process) starting from a $\boldsymbol{U}\left(\epsilon_0\right)$ with $\epsilon_0 \in \left(-\frac{1}{\sqrt{3}}, 0\right)$. That $\frac{d}{d\epsilon} f\left(\boldsymbol{U}\left(\epsilon\right)\right) < 0$ suggests that the search path, parametrized by ϵ, is given by $\epsilon = \epsilon_0 + \alpha$ where $\alpha > 0$. However, note that $f\left(\boldsymbol{U}\left(0\right)\right) > \lim_{\epsilon \uparrow 0^-} f\left(\boldsymbol{U}\left(\epsilon\right)\right)$. The infinitesimal search can never pass the singular point $\epsilon = 0$. A detailed illustration is given in Fig. 2.2.1b. Refer next Section for more detailed explanation.

There are several approaches to address the singularity issue. In [21], Dai et al. proposed the Subspace Evolution and Transfer (SET) method to detect the barriers that are created by singular points and jump across the barriers whenever they are detected. This method achieves good empirical performance. However, the definition of barriers and the design of the jump step are somewhat heuristic. Furthermore, the detection and jump steps are complicated, suggesting that the computational cost per iteration is very high. With the goal of obtaining certain performance guarantees, the same set of authors proposed to replace the original objective function (2.2.2) with a geometric objective function [19]. With this replacement, strong performance guarantees, better than those for ℓ_1-minimization, were obtained for certain special cases. However, the geometric objective function is quite different from the original objective function.

More recently, a regularization technique was proposed in [11], where the

new objective function is given by

$$\begin{aligned} f_\mu(\boldsymbol{U}) &= \min_{\boldsymbol{W}\in\mathbb{R}^{r\times n}} \|\boldsymbol{X}_\Omega - \mathcal{P}_\Omega(\boldsymbol{UW})\|_F^2 + \mu\|\boldsymbol{UW}\|_F^2 \\ &= \min_{\boldsymbol{W}\in\mathbb{R}^{r\times n}} \|\boldsymbol{X}_\Omega - \mathcal{P}_\Omega(\boldsymbol{UW})\|_F^2 + \mu\|\boldsymbol{W}\|_F^2 . \end{aligned} \tag{2.2.7}$$

The regularized objective function in [11] is slightly different from the one presented in Eq. (2.2.7): the regularization term in [11] only involves the unobserved entries $(\boldsymbol{UW})_{i,j}$ where $(i,j) \notin \Omega$ while we include all entries of $\hat{\boldsymbol{X}} = \boldsymbol{UW}$ in the regularization term. This small change doesn't affect the essence of this method but makes the rigorous analysis relatively easier. It is straightforward to verify that the objective function f_μ is continuous whenever the regularization constant $\mu > 0$ [11]. In fact, the function f_μ is a continuous approximation of f: the smaller $\mu > 0$, the better the approximation. In practice, one may choose a small constant $\mu > 0$ or keep decreasing the value of μ along the optimization process.

However, a careful study reveals that the continuous approximation f_μ creates local minimum in the neighborhood of singular points of f. This is undesirable as the optimization process may converge to the created local minimum. To demonstrate this phenomenon, we rigorously analyze how $f_\mu(\boldsymbol{U}(\epsilon))$ behaves in Example 1. The details are presented in Subsection 2.6.1. The analysis shows the following. Let $\epsilon_0 < 0$ denote the starting point. For all $\epsilon_0 \in \left(-\frac{1}{4}, 0\right)$, let μ be sufficiently small such that $\mu < |\epsilon_0|^3/3$. One can get that $\epsilon_0 < -\sqrt{\mu/2} < 0$. We prove that $f'_\mu(\epsilon_0) < 0$, $f'_\mu\left(-\sqrt{\mu/2}\right) > 0$ and $f'_\mu(0) < 0$. This implies that there exist a minimum $\epsilon_{\min,\mu} \in \left(\epsilon_0, -\sqrt{\mu/2}\right)$ and a maximizer $\epsilon_{\max,\mu} \in \left(-\sqrt{\mu/2}, 0\right)$. As a result, the gradient search will stop at the point $\epsilon_{\min,\mu} \in \left(\epsilon_0, -\sqrt{\mu/2}\right)$. The gradient search will not converge

to the global minimizer $\epsilon^* = 1/\sqrt{3}$.

Power Factorization (PF) [39] is another low rank matrix completion approach, which optimizes $\boldsymbol{U}$ and $\boldsymbol{W}$ alternatively to minimize $\|\boldsymbol{X}_\Omega - \mathcal{P}_\Omega(\boldsymbol{UW})\|_F^2$ using linear-least squares procedure. In paper [39], the objective function is $\|vec(\boldsymbol{X}_\Omega) - \mathcal{A}(\boldsymbol{UW})\|_2$, where $\mathcal{A}$ is a linear operator that $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^q$ and q is the number of observations. For the sake of consistency, we also change the objective function in this case. It has been proved that the objective function is discontinuous when optimizes $\boldsymbol{W}$ while keeps $\boldsymbol{U}$ fixed in Example 1. In PF, one also has to optimize $\boldsymbol{U}$ by fixing the optimized $\boldsymbol{W}$. Consider $\boldsymbol{U}(\epsilon_1) = \left[\sqrt{1-2\epsilon_1^2}, \epsilon_1, \epsilon_1\right]^T$ and the objective function

$$f(\boldsymbol{W}) = \min_{\boldsymbol{U}(\epsilon_1)} \left\|\boldsymbol{X}_\Omega^T - \mathcal{P}_{\Omega^T}\left(\boldsymbol{W}^T\boldsymbol{U}^T\right)\right\|_F^2.$$

The objective function is $f(\boldsymbol{W}) = f_3(\boldsymbol{W}) = \min_{\epsilon_1} \left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \epsilon_1 \right\|_F^2$. Take the derivative with respect to ϵ_1 and let $\frac{df_3(\boldsymbol{W})}{d\epsilon_1} = 0$. It is easy to obtain that

$$\epsilon_1 = \begin{cases} \frac{w_1+w_2}{w_1^2+w_2^2},\ f_3(\epsilon_1(\epsilon)) = 0 & \text{if } \epsilon \neq 0, \\ 1 \quad ,\ f_3(\epsilon_1(\epsilon)) = 1 & \text{if } \epsilon = 0. \end{cases}$$

Then, it is obvious that the objective function is $f(\boldsymbol{W}(\epsilon))$ is also discontinuous at point $\boldsymbol{W}(\epsilon = 0)$.

In a summary, the singularity issue cannot be completely addressed by either the SET method in [21], or the geometric objective function in [19], or the regularization technique in [11] or the PF method in [39].

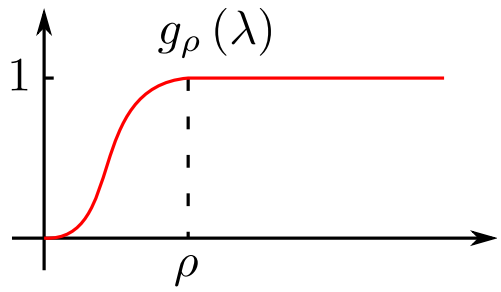


Figure 2.3.1: The modulation function $g_\rho(\lambda)$.

2.3 Smoothed Objective Function

The key idea behind the new approach is to replace the original objective function (2.2.2) with a continuous objective function that is similar to the original one. As discussed in Section 2.2.2, adding a regularization term results in a continuous objective function but it cannot solve the singularity issue. Our approach is to introduce multiplicative terms rather than additive ones.

This function is designed so that it is second order differentiable, i.e., give $\rho > 0$,

$$g(0) = 0,\ g'(0) = 0,\ g''(0) = 0,$$
$$g(\rho) = 1,\ g'(\rho) = 0,\ g''(\rho) = 0.$$

Towards that end, we define the modulation function

$$g_\rho(\lambda) = \begin{cases} 0 & \text{if } \lambda \le 0, \\ 6\left(\frac{\lambda}{\rho}\right)^5 - 15\left(\frac{\lambda}{\rho}\right)^4 + 10\left(\frac{\lambda}{\rho}\right)^3 & \text{if } \lambda \in (0,\rho), \\ 1 & \text{if } \lambda \ge \rho. \end{cases} \tag{2.3.1}$$

An illustration of the modulation function is given in Fig. 2.3.1.

We propose to replace the original objective function (2.2.2) with the new objective function defined as

$$\tilde{f}(\boldsymbol{U}) = \sum_{i=1}^{n} f_i(\boldsymbol{U}_{\Omega_i}) \cdot g_{\rho_i}(\lambda_{\min}(\boldsymbol{U}_{\Omega_i})), \tag{2.3.2}$$

where $\lambda_{\min}(\boldsymbol{U}_{\Omega_i})$ gives the minimum singular value of the matrix $\boldsymbol{U}_{\Omega_i}$. With this replacement, the matrix completion problem is formulated as

$$\min_{\boldsymbol{U} \in \mathcal{U}_{m,r}} \tilde{f}(\boldsymbol{U}) = 0. \tag{2.3.3}$$

The smoothed objective function has nice properties described in the following Theorem. It suggests that the smoothed function $\tilde{f}$ is a good approximation of f.

Theorem 2.3.1. *1. For any $\rho_1 > 0, \cdots, \rho_n > 0$, then $\tilde{f}(\boldsymbol{U})$ is continuous everywhere.*

2. When $\rho_1 = \cdots = \rho_n = 0$, then $\tilde{f}$ and f differ only at the singular points. That is, $\tilde{f} = f$ for all $\boldsymbol{U} \in \mathcal{U}_{m,r} \backslash \mathcal{S}$, where

$$\mathcal{S} = \bigcup_{i=1}^{n} \mathcal{S}_i = \bigcup_{i=1}^{n} \{\boldsymbol{U} : \lambda_{\min}(\boldsymbol{U}_{\Omega_i}) = 0\}. \tag{2.3.4}$$

3. For $\forall c \in \mathbb{R}$, define the lower level set $\mathcal{U}_f(c) = \{\boldsymbol{U} : f(\boldsymbol{U}) \leq c\}$. When $\rho_1 = \cdots = \rho_n = 0$, $\mathcal{U}_{\tilde{f}}(c)$ is the closure of $\mathcal{U}_f(c)$.

The proof is presented in Subsection 2.6.2. Indeed, from the last part of Theorem 2.3.1, the proposed function $\tilde{f}$ is *the best possible lower semi-continuous approximation* of the original discontinuous function f when $\rho_1 = \cdots = \rho_n = 0$.

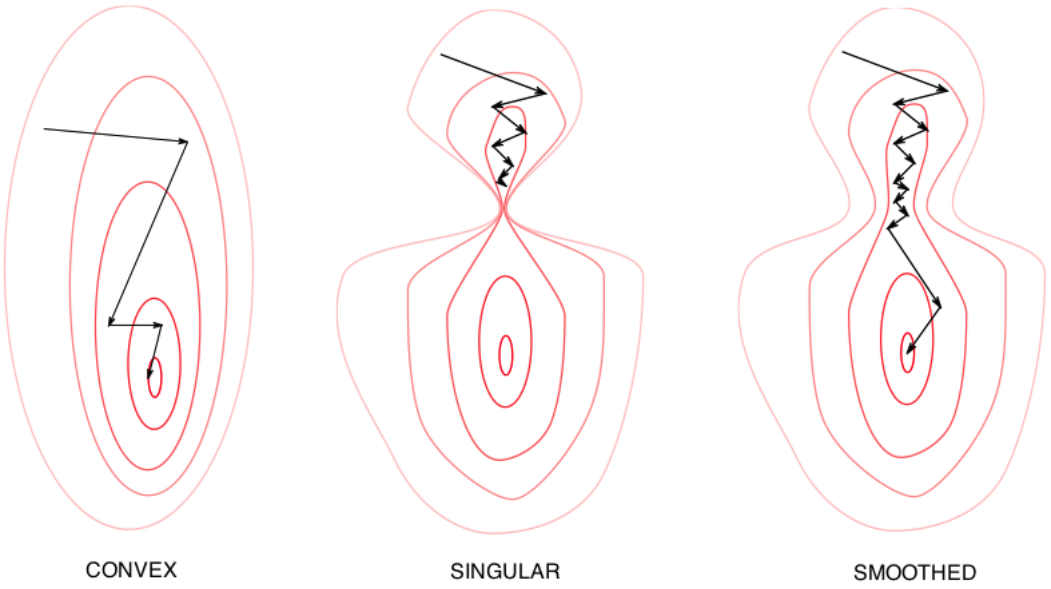


Figure 2.3.2: This is an alternative illustration of the singularity issue and the smoothed solution. Left: convex case. It is easy to find the optimal. Middle: singular case. The singular point prevents the searching path to optimal. Right: Our smooth solution. It lets the searching process pass through.

The effect of adding the modulation functions, intuitively speaking, is to open "tunnels" for the optimization process to pass through. See Fig. 2.2.1d for an illustration. The smaller ρ_is are, the better the function $\tilde{f}$ approximates the function f, but the narrower the tunnels are, and the slower the convergence rate is. The next Subsection discusses a particular way to choose the parameters ρ_is.

2.3.1 A Choice of Parameter ρ_is

As discussed above, the positive parameters ρ_is should be chosen as small as possible for the purpose of approximation, but as large as possible to speed up the convergence rate. In this Chapter, the tool for choosing ρ_is is random matrix theory.

We use an example to intuitively demonstrate that the parameters ρ_is may be different for different columns. Consider a matrix completion problem with $m = 100$ and $r = 2$. Then the variable $\boldsymbol{U} \in \mathcal{U}_{100,2}$. Suppose that all the entries in the first column of $\boldsymbol{X}_\Omega$ are known but only the first two entries in the second column of $\boldsymbol{X}_\Omega$ are revealed, i.e., $\Omega_1 = [m]$ and $\Omega_2 = \{1, 2\}$. According to Eq. (2.3.1) it makes no difference for all $\rho_1 \in (0, 1)$ while $\lambda_{\min}(\boldsymbol{U}_{\Omega_1}) = 1$. However, $\lambda_{\min}(\boldsymbol{U}_{\Omega_2}) < 1$ for most $\boldsymbol{U} \in \mathcal{U}_{100,2}$. The choice of ρ_2 decides how likely $g_{\rho_2}(\boldsymbol{U}_{\Omega_2}) < 1$: the larger ρ_2 is, the more likely $g_{\rho_2}(\boldsymbol{U}_{\Omega_2}) < 1$.

The rigorous interpretation of the above observation is as follows. The space $\mathcal{U}_{m,r}$ is compact (Closed and bounded). Hence, the uniform probability measure μ on $\mathcal{U}_{m,r}$ is well defined. Consider a random $\boldsymbol{U} \in \mathcal{U}_{m,r}$ generated from the uniform distribution μ. For a given $\Omega_i \subset [m]$, let $\lambda_1 \geq \cdots \geq \lambda_r \geq 0$ be the singular values of $\boldsymbol{U}_{\Omega_i}$. The empirical distribution of λ_js is given by

$$F_i(t) = \frac{1}{r} \left| \{ j : \ \lambda_j \leq t \} \right|.$$

The quantity $\mathrm{E}_\mu [F_i(t)]$ gives the probability that $\lambda_{\min}(\boldsymbol{U}_{\Omega_i}) \leq t$, where E_μ is the expectation with respect to the probability distribution μ. The parameter ρ_i should be chosen so that $\mathrm{E}_\mu [F_i(t)]$ is small.

Generally speaking, it is difficult to exactly quantify the probability $\mathrm{E}_\mu [F_i(t)]$. Nevertheless, two facts are useful in approximating it. Firstly, the empirical distribution $F_i(t)$ relies only on the size of Ω_i, i.e., $|\Omega_i|$, rather than the specific choice of Ω_i. Secondly, when $m, r, q_i \to \infty$ simultaneously with fixed ratios, $F_i(t)$ converges to a proper probability distribution. The following theorem states the asymptotic behavior of $\lambda_{\min}(\boldsymbol{U}_{\Omega_i})$.

Theorem 2.3.2. *Let $\boldsymbol{U} \in \mathcal{U}_{m,r}$ be randomly generated from the uniform dis-*

tribution on $\mathcal{U}_{m,r}$. Suppose that $r \leq \frac{m}{2}$.[5] *Let $\Omega_i \subseteq [m]$ and $q_i = |\Omega_i|$. Define*

$$\begin{cases} \alpha_i = \left(1 - \frac{r}{m} - \frac{q_i}{m}\right)^2, \ \beta_i = \left(\frac{q_i}{m} - \frac{r}{m}\right)^2, & \text{if } r \leq q_i \leq \frac{m}{2}, \\ \alpha_i = \left(\frac{q_i}{m} - \frac{r}{m}\right)^2, \ \beta_i = \left(1 - \frac{r}{m} - \frac{q_i}{m}\right)^2, & \text{if } r \leq \frac{m}{2} < q_i, \end{cases}$$

$$\begin{cases} a_i = \frac{1+\alpha_i-\beta_i-\sqrt{(1+\alpha_i-\beta_i)^2-4\alpha_i}}{2}, \\ b_i = \frac{1+\alpha_i-\beta_i+\sqrt{(1+\alpha_i-\beta_i)^2-4\alpha_i}}{2}, \end{cases}$$

and

$$\tau_i = \begin{cases} \sqrt{1-b_i} & \text{if } r \leq q_i \leq \frac{m}{2}, \\ \sqrt{a_i} & \text{if } r \leq \frac{m}{2} < q_i. \end{cases} \tag{2.3.5}$$

Then for any $\epsilon > 0$, as $m, r, q_i \to \infty$ with constant ratios,

$$\Pr\left(\lambda_{\min}\left(\boldsymbol{U}_{\Omega_i}\right) \leq \tau_i - \epsilon\right) \to 0,$$

and

$$\Pr\left(\lambda_{\min}\left(\boldsymbol{U}_{\Omega_i}\right) \leq \tau_i + \epsilon\right) \to 1.$$

Note that although the above results are asymptotic, they provide a good approximation for finite m, r, q_i. See [22] for the detailed discussion of the convergence rate and a numerical comparison between the empirical distribution $F_i(t)$ and the asymptotic distribution.

Our choice of the parameters ρ_is are based on the above asymptotic results. Specifically, we set $\rho_i = \eta\tau_i$, where τ_i is defined in (2.3.5) and η is a constant independent of individual columns.

[5] In this Chapter, we only consider the case where $r \leq \frac{m}{2}$ because this is the most useful case in practice.

2.3.2 Gradient of the Smoothed Objective Function

In this Subsection, we compute $\nabla \tilde{f} \in \mathbb{R}^{m\times r}$ where the (k,ℓ)th entry of $\nabla \tilde{f}$ is given by $\partial \tilde{f}/\partial \boldsymbol{U}_{k,\ell}$.

To start, note that $\nabla \tilde{f} = \sum_i \nabla f_i \cdot g_{\rho_i} + f_i \cdot \nabla g_{\rho_i}$. The following proposition computes ∇f_i and ∇g_{ρ_i}. To simplify the notations, we omit the subscript i.

Proposition 2.3.3. *Suppose that $\lambda_{\min}(\boldsymbol{U}_\Omega) > 0$ and that it is not a repetitive singular value, i.e., all other singular values of $\boldsymbol{U}_\Omega$ are strictly larger than $\lambda_{\min}$. Let $\boldsymbol{u}_{\min}$ and $\boldsymbol{v}_{\min}$ be the left and right singular vectors corresponding to $\lambda_{\min}$ of $\boldsymbol{U}_\Omega$. It holds that*

$$\nabla f = -2\left(\boldsymbol{x}_\Omega - \boldsymbol{U}_\Omega \boldsymbol{w}\right)\boldsymbol{w}^T, \tag{2.3.6}$$

where $\boldsymbol{w} = \boldsymbol{U}_\Omega^\dagger \boldsymbol{x}_\Omega$, and $\nabla g_\rho = \frac{dg_\rho}{d\lambda_{\min}} \cdot \nabla \lambda_{\min}$ where

$$\frac{dg_\rho}{d\lambda_{\min}} = \begin{cases} \frac{30}{\rho}\left(\frac{\lambda_{\min}}{\rho}\right)^4 - \frac{60}{\rho}\left(\frac{\lambda_{\min}}{\rho}\right)^3 + \frac{30}{\rho}\left(\frac{\lambda_{\min}}{\rho}\right)^2 & \text{if } \lambda \in (0,\rho), \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\nabla \lambda_{\min} = \boldsymbol{u}_{\min}\boldsymbol{v}_{\min}^T.$$

The proof of this Proposition is detailed in Subsection 2.6.3.

Remark 2.3.4. When $\lambda_{\min}(\boldsymbol{U}_\Omega) = 0$, f and $\lambda_{\min}$ are not differentiable at $\boldsymbol{U}_\Omega$. However, since $g_\rho = 0$ and $dg_\rho/d\lambda_{\min} = 0$, one may set $\nabla \tilde{f} = \boldsymbol{0}$ in practice.

Remark 2.3.5. If $\lambda_{\min}$ is a repetitive singular value, i.e., there exists more than one singular values equaling to $\lambda_{\min}$, then $\nabla \lambda_{\min}$ is not well defined. However, this happens with probability zero when $\boldsymbol{U}$ is randomly generated from the

uniform distribution on $\mathcal{U}_{m,r}$. Furthermore, even this happens during the optimization process, directly applying $\nabla\lambda_{\min} = \boldsymbol{u}_{\min}\boldsymbol{v}_{\min}^T$ does not introduce any practical issue in our simulations.

2.3.3 An Illustration of the Smoothed Function

In this Subsection, we discuss how our proposed approach addresses the singularity issue with an example. For our proposed algorithm, one has to compute the smoothed objective function. Recall the Example 1, the smoothed object function is $\tilde{f} = \tilde{f}_1 + \tilde{f}_2 = f_1 g_1 + f_2 g_2$. According to equation (2.3.1), the singular value of $\boldsymbol{U}_{\Omega_i}$ is needed. The singular value of $\boldsymbol{U}_{\Omega_1} = [\ \epsilon \quad \epsilon\ ]^T$ is $\lambda_1 = \sqrt{2\epsilon^2}$ and the singular value of $\boldsymbol{U}_{\Omega_2} = [\ \sqrt{1-2\epsilon^2} \quad \epsilon]^T$ is $\lambda_2 = \sqrt{1-\epsilon^2}$. The proof of the singular values is shown in Subsection 2.6.4. Let $\eta = 1$, it is easy to compute the value of $\rho = \frac{1}{3}$. For $\epsilon \in [-0.1, 0]$, we have $0 \leq \lambda_1 < \rho$ and $\lambda_2 > \rho$. Then the smooth functions are

$$\begin{cases} g_1 & = 6(\lambda_1/\rho)^5 - 15(\lambda_1/\rho)^4 + 10(\lambda_1/\rho)^3 \\ g_2 & = 1 \end{cases}.$$

Consider the gradient of the first atomic function, as $\epsilon \to 0$, then g_1 approaches to zero. Hence, at position $[1, 0, 0]^T$ the value of the first atomic function of the new objective function is 0. Therefore, the smoothed objective function is continuous along $[\sqrt{1-2\epsilon^2}, \epsilon, \epsilon]^T$ and its gradient is always negative. Then a gradient decent method will find the global minimum. An intuition on how the proposed smoothing technique addresses the singular issue is given in Fig. 2.2.1d. It opens a tunnel to let the line search process pass through the singular point. Hence, the singularity issue is addressed.

2.4 Algorithm Implementation

In the last Section, a new objective function $\tilde{f}$ was defined in (2.3.2). This Section is devoted to developing an efficient algorithm to solve the optimization problem (2.3.3). We first briefly describe the generic optimization methods and then explain how to modify a generic method so that it fits the optimization problem at hand.

2.4.1 Optimization Methods in Euclidean Space

A line search strategy is essential for optimization methods. The main steps are summarized as follows. With slight abuse of notations, consider the generic scenario of minimizing an objective function $f(\boldsymbol{u})$ where $\boldsymbol{u} \in \mathbb{R}^n$. A line search method updates the variable $\boldsymbol{u}$ iteratively via

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \alpha_k \boldsymbol{p}_k, \tag{2.4.1}$$

where α_k is the step size and $\boldsymbol{p}_k$ is the search direction. In the gradient descent method, $\boldsymbol{p}_k$ is taken as $-\nabla f_k$, while in a Newton method,

$$\boldsymbol{p}_k = -\boldsymbol{B}_k^{-1} \nabla f_k, \tag{2.4.2}$$

where $\boldsymbol{B}_k$ is a symmetric and non-singular matrix that approximates the Hessian. When the Hessian approximation $\boldsymbol{B}_k$ is appropriately chosen, the convergence rate of a Newton method is much faster than that of the gradient descent [58].

We are particularly interested in a quasi-Newton method. The advantage is that it requires only the gradient of the objective function to be computed: the

Hessian is estimated using the gradients from multiple iterations. In particular, let

$$\boldsymbol{s}_k = \boldsymbol{u}_{k+1} - \boldsymbol{u}_k, \tag{2.4.3}$$

$$\boldsymbol{y}_k = \nabla f\left(\boldsymbol{u}_{k+1}\right) - \nabla f\left(\boldsymbol{u}_k\right), \tag{2.4.4}$$

and $\boldsymbol{B}_k$ be an approximation of the Hessian at the point $\boldsymbol{u}_k$. In the quasi-newton method [58], the most common Hessian approximation is given by $\boldsymbol{B}_{k+1}\boldsymbol{s}_k = \boldsymbol{y}_k$, which is also known as *secant equation*. With knowledge of $\boldsymbol{u}_k$ and $\nabla f\left(\boldsymbol{u}_k\right)$ from multiple iterations, one can estimate $\boldsymbol{B}_{k+1}$. Let $\boldsymbol{H}_{k+1}$ denote the inverse of $\boldsymbol{B}_{k+1}$, i.e., $\boldsymbol{H}_{k+1} = \boldsymbol{B}_{k+1}^{-1}$. The search direction of $f(\boldsymbol{u}_k)$ can be evaluated from

$$\boldsymbol{p}_k = -\boldsymbol{H}_{k+1}\nabla f\left(\boldsymbol{u}_k\right). \tag{2.4.5}$$

To avoid computation of matrix inverse, the most popular quasi-Newton method, Broyden–Fletcher–Goldfarb–Shanno (BFGS) method [58], chooses to update $\boldsymbol{H}_k$ via

$$\boldsymbol{H}_{k+1} = (\boldsymbol{I} - \frac{\boldsymbol{s}_k\boldsymbol{y}_k^T}{\boldsymbol{y}_k^T\boldsymbol{s}_k})\boldsymbol{H}_k(\boldsymbol{I} - \frac{\boldsymbol{y}_k\boldsymbol{s}_k^T}{\boldsymbol{y}_k^T\boldsymbol{s}_k}) - \frac{\boldsymbol{s}_k\boldsymbol{s}_k^T}{\boldsymbol{y}_k^T\boldsymbol{s}_k}. \tag{2.4.6}$$

A flowchart of the BFGS method is detailed in Algorithm 2.1.

As we will explain in the next Subsection, the BFGS method in Algorithm 2.1 needs to be modified to fit the optimization problem in (2.3.3).

2.4.2 Why Optimize on Grassmann Manifold

In this Subsection, we show why we do not directly apply standard optimization techniques. Then we present a global coordinates based BFGS algorithm on Grassmann manifold in Section. 2.4.3.

Algorithm 2.1 The BFGS Method [58]

Given a starting point $\boldsymbol{u}_0$, convergence tolerance $\delta > 0$, and the initial approximation $\boldsymbol{H}_0 = \boldsymbol{I}$, perform the following.
$k \leftarrow 0$;
while $\|\nabla f(\boldsymbol{u}_k)\| > \delta$

Compute the search direction of $f(\boldsymbol{u}_k)$ using Eq. (2.4.5).

Set Eq. (2.4.1) where α_k is computed from a line search backtracking procedure to satisfy the *Armijo condition,*

$$f(\boldsymbol{u}_{k+1}) \leq f(\boldsymbol{u}_k) + \alpha_k \nabla f_k^T \boldsymbol{p}_k. \tag{2.4.7}$$

Compute Eq. (2.4.3) and Eq. (2.4.4). If $\boldsymbol{s}_k^T \boldsymbol{y}_k > 0$, compute $\boldsymbol{H}_{k+1}$by (2.4.6). Otherwise, $\boldsymbol{H}_{k+1} = \boldsymbol{I}$;

$k \leftarrow k + 1$;
end(while)

We first discuss the necessity of the constraint $\boldsymbol{U} \in \mathcal{U}_{m,r}$. It is clear that for any $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{W} \in \mathbb{R}^{r\times n}$, the matrix $\boldsymbol{UW}$ has rank at most r. It looks that restricting the search space to $\mathcal{U}_{m,r}$ is unnecessary. We shall argue that the constraint $\boldsymbol{U} \in \mathcal{U}_{m,r}$ helps in understanding how good and how singular an estimate $\hat{\boldsymbol{U}}$ is. Recall Example 1. Clearly $\boldsymbol{U}^* = \frac{1}{\sqrt{3}}[1,1,1]^T$ is a global optimum. If we drop the constraint $\boldsymbol{U} \in \mathcal{U}_{m,r}$ and consider a sequence $\boldsymbol{U}^{(k)} = [k+1, k, k]^T$, then it is clear that the ℓ_2 distance between $\boldsymbol{U}^{(k-1)}$ and $\boldsymbol{U}^{(k)}$ is always three for all k. The sequence $\boldsymbol{U}^{(k)}$ does not converge. If we enforce the constraint $\boldsymbol{U} \in \mathcal{U}_{m,r}$, then the equivalent sequence is given by $[k+1, k, k]^T / \sqrt{2k^2 + (k+1)^2}$ which converges to the global optimum $\boldsymbol{U}^*$. The constraint $\boldsymbol{U} \in \mathcal{U}_{m,r}$ helps in defining the goodness of an estimate $\hat{\boldsymbol{U}}$. In terms of singularity detection, the constraint $\boldsymbol{U} \in \mathcal{U}_{m,r}$ is also necessary. Otherwise, let $\boldsymbol{U}' = \alpha \boldsymbol{U}$ where $\alpha > 0$ is a real number. No matter how well-conditioned $\boldsymbol{U}_\Omega$ is, the matrix $\boldsymbol{U}'_\Omega = (\alpha \boldsymbol{U})_\Omega$ can be made arbitrarily close to

being singular by decreasing α. From the above discussions, the advantage of the constraint $\boldsymbol{U} \in \mathcal{U}_{m,r}$ becomes clear.

We then show the exact space on which the function f and $\tilde{f}$ are defined. Let $\boldsymbol{V} \in \mathcal{U}_{r,r}$. For any given $\boldsymbol{U} \in \mathcal{U}_{m,r}$, $\Omega_i \subseteq [m]$ and $\boldsymbol{w} \in \mathbb{R}^r$, it holds that $(\boldsymbol{U}\boldsymbol{w})_{\Omega_i} = \left((\boldsymbol{U}\boldsymbol{V})\left(\boldsymbol{V}^T\boldsymbol{w}\right)\right)_{\Omega_i}$ and $\lambda_{\min}\left(\boldsymbol{U}_{\Omega_1}\right) = \lambda_{\min}\left((\boldsymbol{U}\boldsymbol{V})_{\Omega_1}\right)$. As a result, $f\left(\boldsymbol{U}\right) = f\left(\boldsymbol{U}\boldsymbol{V}\right)$ and $\tilde{f}\left(\boldsymbol{U}\right) = \tilde{f}\left(\boldsymbol{U}\boldsymbol{V}\right)$. Let $\mathrm{span}\left(\boldsymbol{U}\right)$ denote the subspace spanned by the matrix $\boldsymbol{U}$. It is clear that $\mathrm{span}\left(\boldsymbol{U}\right) = \mathrm{span}\left(\boldsymbol{U}\boldsymbol{V}\right)$.

More formally, the variables of f and $\tilde{f}$ are in the so called Grassmann manifold. The Grassmann manifold $\mathcal{G}_{m,r}$ is the set of all r-dimensional linear subspaces in $\mathbb{R}^m$, i.e., $\mathcal{G}_{m,r} = \{\mathrm{span}\left(\boldsymbol{U}\right) : \boldsymbol{U} \in \mathcal{U}_{m,r}\}$. Given a subspace $\mathcal{U} \in \mathcal{G}_{m,r}$, one can always find a matrix $\boldsymbol{U} \in \mathcal{U}_{m,r}$ such that $\mathcal{U} = \mathrm{span}\left(\boldsymbol{U}\right)$. The matrix $\boldsymbol{U}$ is referred to as a generator matrix of $\mathcal{U}$ and the columns of $\boldsymbol{U}$ are often referred to as an orthonormal basis of $\mathcal{U}$. Since $\mathrm{span}\left(\boldsymbol{U}\right) = \mathrm{span}\left(\boldsymbol{U}\boldsymbol{V}\right)$ for all $\boldsymbol{V} \in \mathcal{U}_{r,r}$, it is clear that the generator matrix for a given subspace is not unique. Nevertheless, a given matrix $\boldsymbol{U} \in \mathcal{U}_{m,r}$ uniquely defines a subspace. It is therefore common in practice to use $\boldsymbol{U}$ to represent both the matrix and its spanned subspace.

For the problem at hand, the manifold structure needs to be considered in implementing an optimization method. In the standard quasi-Newton method which is designed for the Euclidean space, the difference between two points $\boldsymbol{U}^{(k+1)} - \boldsymbol{U}^{(k)}$ is used for Hessian estimation. Suppose that $\boldsymbol{U}^{(k)} = -\boldsymbol{U}^{(k+1)}$. Then $\boldsymbol{U}^{(k+1)} - \boldsymbol{U}^{(k)} = 2\boldsymbol{U}^{(k+1)}$ in the Euclidean space. However, if we take the Grassmann manifold into consideration, then $\mathrm{span}\left(\boldsymbol{U}^{(k+1)}\right) = \mathrm{span}\left(\boldsymbol{U}^{(k)}\right)$ and the proper definition of $\boldsymbol{U}^{(k+1)} - \boldsymbol{U}^{(k)}$ (in the Grassmann manifold) should yield $\boldsymbol{0}$.

2.4.3 From Euclidean Space to Grassmann Manifold

In this Subsection, we will discuss the modifications needed for BFGS method on Grassmann manifold since the equations of BFGS are not well-defined on Grassmann manifold. Then a global coordinates based BFGS algorithm on Grassmann manifold is presented.

To re-define (2.4.1), note that its meaning is to move $\boldsymbol{u}_k$ along the direction $\boldsymbol{p}_k$ with step size α_k. The same operation can be defined on Grassmann manifold as follows. For a given $\boldsymbol{U} \in \mathcal{U}_{m,r}$ represent by $\mathcal{U} = span(\boldsymbol{U}) \in \mathcal{G}_{m,r}$, define $\mathcal{T}$ as the tangent space at $\boldsymbol{U}$, which contains all possible valid 'directions' such that $\mathcal{T}_{\boldsymbol{U}} = \left\{ \triangle \in \mathbb{R}^{m\times r} : \triangle^T \boldsymbol{U} = 0 \right\}$. For any given step size t, the obtained new point $\boldsymbol{U}(t)$ on Grassmann manifold is given by

$$\boldsymbol{U}_1 = \boldsymbol{U}(t) = (\boldsymbol{U}_0 \boldsymbol{V}_\triangle, \boldsymbol{U}_\triangle) \begin{pmatrix} \cos \Sigma t \\ \sin \Sigma t \end{pmatrix} \boldsymbol{V}_\triangle^T, \tag{2.4.8}$$

where $\triangle = \boldsymbol{U}_\triangle \Sigma \boldsymbol{V}_\triangle$ is the compact singular value decomposition of the tangent vector. This actually generates the so called geodesic path, which will play an essential role in re-defining (2.4.3) and (2.4.4). The details and proof of Eq. (2.4.8) have been given in [28]. It is worth to note that the gradient of the objective function $\tilde{f}(\boldsymbol{U})$ on Grassmann manifold is $\mathcal{T}_{\boldsymbol{U}} \ni \nabla \tilde{f}(\boldsymbol{U}) = (\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^T)\frac{\partial \tilde{f}(\boldsymbol{U})}{\partial \boldsymbol{U}}$, where $(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^T)$ is the projection on the tangent space $\mathcal{T}_{\boldsymbol{U}}$. In this Section 2.4.3, for the sake of simplicity, we use subscription $(\cdot)_1$ and $(\cdot)_0$ to denote the current step and the previous step, respectively.

Similarly, the re-definition of (2.4.3) is based on its geometric interpretation: moving $\boldsymbol{u}_k$ along $\boldsymbol{s}_k$ (with unit step size) gives $\boldsymbol{u}_{k+1}$. To proceed, it is noteworthy two subtleties. First, the output $\boldsymbol{s}_k$ should be in the tangent space

of $\boldsymbol{u}_k$. Second, extra steps will be needed to compensate the effects when the matrix $\boldsymbol{u}$ is used to represent the subspace $\mathcal{U} = span(\boldsymbol{u})$. In particular, $\boldsymbol{u}$ and $\boldsymbol{uv}$ $(\forall \boldsymbol{v} \in \mathcal{U}_{r,r})$ represent the same subspace $span(\boldsymbol{u}) = span(\boldsymbol{uv})$, but moving $\boldsymbol{u}$ along $\triangle \in \mathcal{T}_{\boldsymbol{u}}$ will not give the same point as moving $\boldsymbol{uv}$ $(\boldsymbol{v} \neq \boldsymbol{I}_r)$ along $\triangle \in \mathcal{T}_{\boldsymbol{u}}$. This claim can be proved by the following proposition.

Proposition 2.4.1. *Fix $\boldsymbol{U}_0, \boldsymbol{U}_1 \in \mathcal{U}_{m,r}$ and let $\triangle_0 \in \mathbb{R}^{m\times r}$ be the tangent vector in $\mathcal{T}_{\boldsymbol{U}_0}$ such that the geodesic path $\boldsymbol{U}(t)$ given by (2.4.8) satisfies $\boldsymbol{U}(0) = \boldsymbol{U}_0$ and $span(\boldsymbol{U}(1)) = span(\boldsymbol{U}_1)$. Let $\boldsymbol{V}_1\boldsymbol{\Lambda}\boldsymbol{V}_2^T$ be the singular decomposition of $\boldsymbol{U}_0^T\boldsymbol{U}_1$. Denote $\boldsymbol{U}_0\boldsymbol{V}_1$ and $\boldsymbol{U}_1\boldsymbol{V}_2$ by $\bar{\boldsymbol{U}}_0$ and $\bar{\boldsymbol{U}}_1$, respectively. Then, the tangent vector $\triangle$ is given by*

$$\mathcal{T}_{\boldsymbol{U}_0} \ni \triangle_0 = \boldsymbol{G}\mathrm{diag}([\cdots, a_i, \cdots])\boldsymbol{V}_1^T, \tag{2.4.9}$$

where $a_i = \arccos\lambda_i$, where λ_i is the i^{th} singular value of $\boldsymbol{\Lambda}$. Matrix $\boldsymbol{G} = diag[\cdots, g_i, \cdots]$ and

$$g_i = \begin{cases} \frac{\bar{\boldsymbol{U}}_{1,:i} - \lambda_i \bar{\boldsymbol{U}}_{0,:i}}{\left\|\bar{\boldsymbol{U}}_{1,:i} - \lambda_i \bar{\boldsymbol{U}}_{0,:i}\right\|} & \text{if } \lambda_i \neq 1, \\ 0 & \text{if } \lambda_i = 1. \end{cases}$$

By using (2.4.8), it is easy to show that

$$\boldsymbol{U}(1) = \boldsymbol{U}_1\boldsymbol{V}_2\boldsymbol{V}_1^T. \tag{2.4.10}$$

Hence, the above claim is proved. The details of the proof of this proposition refer to Dai et al's further study Lemma 1 in [19]. Then, on Grass-

mann manifold, moving $\boldsymbol{U}_0$ along $\triangle_0$ in Eq. (2.4.9) gives $\boldsymbol{U}(1)$ in Eq. (2.4.10). Furthermore, $\triangle_0$ lies in the tangent space of $\boldsymbol{U}_0$. The matrix expression of transporting an arbitrary tangent vector $\triangle$ from $\boldsymbol{U}_0$ to $\boldsymbol{U}(t)$ is given in [28, 4] and [67],

$$\bar{\boldsymbol{T}}_0 = (\boldsymbol{U}_0\boldsymbol{V}_\triangle, \boldsymbol{U}_\triangle) \begin{pmatrix} -\sin \Sigma t \\ \cos \Sigma t \end{pmatrix} \boldsymbol{U}_\triangle^T + (I - \boldsymbol{U}_\triangle \boldsymbol{U}_\triangle^T). \tag{2.4.11}$$

The transfer operator between two arbitrary tangent spaces is investigated in the following Lemma.

Lemma 2.4.2. *Given our objective function* $f(\boldsymbol{U}) = \|\boldsymbol{X}_\Omega - \mathcal{P}(\boldsymbol{U}\boldsymbol{W})\|_F^2$. *Assume we have tangent vectors* $\Delta_0 \in \mathcal{T}_0$ *and* $\bar{\triangle}_1 \in \mathcal{T}_1$. *Suppose that* $\bar{\triangle}_1 = \bar{\boldsymbol{T}}_0 \triangle_0$ *and* $\bar{\boldsymbol{U}}_1 = \boldsymbol{U}_1\boldsymbol{V}_2\boldsymbol{V}_1^T$. *Hence, for tangent vector* $\triangle_0 \in \mathcal{T}_0$, *its parallel vector* $\triangle_1$ *in tangent space* $\mathcal{T}_1$ *is then*

$$\mathcal{T}_1 \ni \triangle_1 = \bar{\boldsymbol{T}}_0 \triangle_0 \boldsymbol{V}_1 \boldsymbol{V}_2^T. \tag{2.4.12}$$

The proof is detailed in Subsection 2.6.5. Eq. (2.4.12) can be reformulated as in vectorization form and parallel transport gives,

$$\vec{\triangle}_1 = \boldsymbol{T}_0 \vec{\triangle} = (\boldsymbol{V}_2\boldsymbol{V}_1^T \otimes \bar{\boldsymbol{T}}_0)\vec{\triangle}_0, \tag{2.4.13}$$

where we use $\vec{(\cdot)}$ denote the vectorization of the matrix $(\cdot)$. Hence, the parallel transport matrix between two arbitrary tangent spaces of $\boldsymbol{U}_0$ and $\boldsymbol{U}_1$ is obtained,

$$\boldsymbol{T}_0 = \boldsymbol{V}_2 \boldsymbol{V}_1^T \otimes \bar{\boldsymbol{T}}_0, \tag{2.4.14}$$

where the symbol $\otimes$ denotes the Kronecker product of operators. Then, according to Eq. (2.4.9) and Eq. (2.4.14), on Grassmann manifold, the Eq. (2.4.3) can be reformulated as a vectorized form

$$\mathcal{T}_1 \ni \boldsymbol{S}_1 : \vec{S}_1 = \boldsymbol{T} \vec{\triangle}_0, \tag{2.4.15}$$

where in this case $\mathbb{R}^{mr \times mr} \ni \boldsymbol{T} = \boldsymbol{I}_r \otimes \boldsymbol{T}_0$.

In order to re-define Eq. (2.4.4), it is worth to note that on manifolds one can not apply subtraction between two vectors as they are in different tangent spaces (Fig. 2.4.1). Based on Eq. (2.4.15), it is straight forward to obtain

$$\mathcal{T}_1 \ni \boldsymbol{Y}_1 : \vec{Y}_1 = \nabla \vec{f}(\boldsymbol{U}_1) - \boldsymbol{T} \cdot \nabla \vec{f}(\boldsymbol{U}_0). \tag{2.4.16}$$

As one needs to calculate Hessian to solve (2.4.5), we have to calculate the inverse of the approximated Hessian, which is computational expensive. Both Qi et al. [63] and Savas and Lim [67] use an approximation of the inverse of the Hessian algorithm. The recursive update formula is

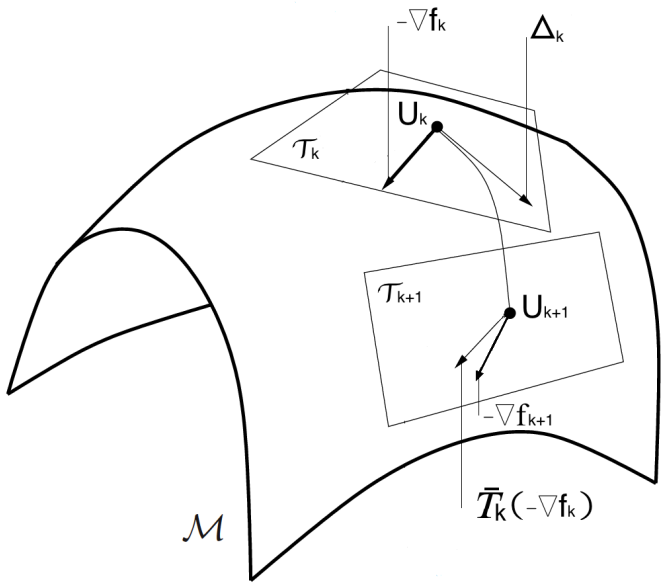


Figure 2.4.1: An illustration of Parallel transport $\bar{\boldsymbol{T}}(-\nabla f_k)$ of tangent vector $-\nabla f_k$ from tangent space $\mathcal{T}_k$ to $\mathcal{T}_{k+1}$ on Grassmann manifold $\mathcal{M}$, where $\triangle_k$ is the decent direction. (After [28])

$$\begin{aligned}\boldsymbol{H}_1 =& \tilde{\boldsymbol{H}}_0 - \frac{\tilde{\boldsymbol{H}}_0 \vec{Y}_1 \vec{S}_1^T}{\left\langle \vec{Y}_1, \vec{S}_1 \right\rangle} - \frac{\vec{S}_1 (\tilde{\boldsymbol{H}}_0 \vec{Y}_1)^T}{\left\langle \vec{Y}_1, \vec{S}_1 \right\rangle} \\ &+ \left[\frac{\left\langle \vec{Y}_1, \tilde{\boldsymbol{H}}_0 \vec{Y}_1 \right\rangle}{\left\langle \vec{Y}_1, \vec{S}_1 \right\rangle^2} - \frac{1}{\left\langle \vec{Y}_1, \vec{S}_1 \right\rangle} \right] \vec{S}_1 \vec{S}_1^T, \end{aligned} \tag{2.4.17}$$

where $\boldsymbol{H} \in \mathbb{R}^{mr \times mr}$ is the inverse of the approximated Hessian and

$$\tilde{\boldsymbol{H}}_0 = \boldsymbol{T} \boldsymbol{H}_0 \boldsymbol{T}^{-1}, \tag{2.4.18}$$

where $\boldsymbol{T}^{-1} : \mathcal{T}_1 \mapsto \mathcal{T}_0$ is the inverse step of $\boldsymbol{T}$. Then, one can obtain the

quasi-Newton searching direction by

$$\mathcal{T}_1 \ni \triangle : \vec{\triangle} = -\boldsymbol{H}_1 \nabla \vec{f}(\boldsymbol{U}_1) \tag{2.4.19}$$

2.4.4 BFGS Algorithm in Global Coordinates

In previous Subsection, we focus on finding the column space, which all the columns of the matrix $\boldsymbol{UW}$ lie in, is spanned by the columns of $\boldsymbol{U}$. One can also use the row space instead. Any details on searching in column/row space refer to [21]. Here, we will discuss a complicated scenario searching in column and row space alternatively. In this case, the searching points are on the column/row space every other iteration. Furthermore, e.g., given $\forall \boldsymbol{U}_{k-1}, \boldsymbol{U}_{k+1} \in \mathcal{U}_{m\times r}$, we do not know either the update direction $\triangle_{k-1} \in \mathcal{T}_{k-1}$ or the geodesic path from $\boldsymbol{U}_{k-1}$ to $\boldsymbol{U}_{k+1}$. We gave an explicit solution of over come these two problems in Section 2.4.3. The pseudo code of the alternate BFGS algorithm on Grassmann manifold, which searches the column and row space alternatively, is given in Algorithm 2.2. It is worth to note that one has to change all the subscript $(\cdot)_0$ and $(\cdot)_1$ in Eq. (2.4.15)-(2.4.19) to $(\cdot)_{k-1}$ and $(\cdot)_{k+1}$ respectively, as we search the column/row space alternatively.

2.4.5 BFGS Algorithm in Local Coordinates

In this Subsection, we present an efficient BFGS algorithm on Grassmann manifold. It is obvious that one has to store and transfer the tangents and Hessian every step in previous BFGS algorithms on Grassmann manifold. Both of the

Algorithm 2.2 Alternate BFGS Algorithm on Grassmann Manifold

Input: starting point $\boldsymbol{U}_0 \in \mathcal{U}_{m\times r}$, the sparse matrix $\boldsymbol{X}_{\boldsymbol{\Omega}}$, its sparse pattern $\boldsymbol{\Omega}$, the inverse of Hessian matrix on column space $\boldsymbol{H}^c = \boldsymbol{I}_{mr}$ and the inverse of Hessian matrix on row space $\boldsymbol{H}^r = \boldsymbol{I}_{mr}$. Let $\boldsymbol{H}_{k+1}$ be the inverse Hessian matrix at current iteration $k+1$. Let $k \leftarrow 0$.

1. If $k+1 < 3$, let $\boldsymbol{H}_1 = \boldsymbol{H}^c$ and $\boldsymbol{H}_2 = \boldsymbol{H}^r$.
2. Else, solve Eq. (2.4.15) and Eq. (2.4.16).
3. If $k+1$ is odd, $\boldsymbol{H}_{k-1} = \boldsymbol{H}^c$, else $\boldsymbol{H}_{k-1} = \boldsymbol{H}^r$.
4. If $\langle S_{k+1}, Y_{k+1}\rangle > 0$, calculate the $\boldsymbol{H}_{k+1}$ with Eq. (2.4.17).
5. End if in step 1.
6. If k+1=odd, $\boldsymbol{H}^c = \boldsymbol{H}_{k+1}$, else $\boldsymbol{H}^r = \boldsymbol{H}_{k+1}$. End if of step 6.
7. Then, solve Eq. (2.4.19).
8. With a step size a, find a good $\boldsymbol{U_{k+1}}$ and $\boldsymbol{V}_{k+1}$ that satisfy the armijo condition, which is $f(\boldsymbol{U}_{k+1}) < f(\boldsymbol{U}_{k-1}) + a \cdot c \cdot \langle \nabla f(\boldsymbol{U}_k), \triangle_k\rangle$,where c is an small positive value.
9. Let $\boldsymbol{U}_{k+1} = orth(\boldsymbol{V}_{k+1}^T)$, $\boldsymbol{X}_\Omega = \boldsymbol{X}_\Omega^T$ and $\boldsymbol{\Omega} = \boldsymbol{\Omega}^T$.
10. Let $k \leftarrow k+1$, return to step 1 until convergence.

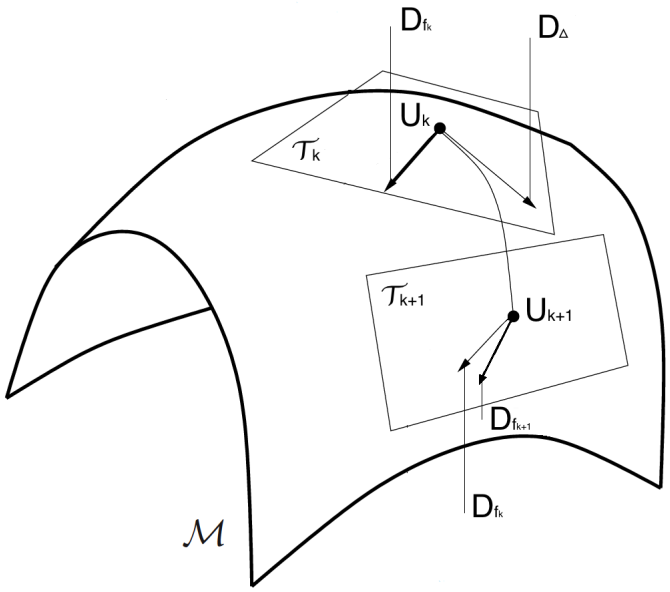


Figure 2.4.2: An illustration of tangent vectors in local coordinates on Grassmann manifold. In local coordinates, the tangent vector $\boldsymbol{D}_{f_k}$ is constant in both tangent spaces $\mathcal{T}_k$ and $\mathcal{T}_{k+1}$. (After [28])

algorithms lack of computational efficiency, especially when the dimension of the searching subspace is high.

Savas and Lim [67] proposed a local coordinates based BFGS approach to over come this high computational cost problem. It has been proved [67] that the local presentation of a tangent $\boldsymbol{D}$ is always constant in the transported basis $\boldsymbol{U}^{\perp}(t)$, which is $\mathcal{T}_0 \ni \triangle = \boldsymbol{U}^{\perp}\boldsymbol{D}$ and $\mathcal{T}_t \ni \triangle(t) = \boldsymbol{U}^{\perp}(t) \cdot \boldsymbol{D}$. The basis $\boldsymbol{U}^{\perp}$ is the orthogonal complement of $\boldsymbol{U}$, which spans the tangent space of $\boldsymbol{U}$. The transported basis is given by $\boldsymbol{U}^{\perp}(t) = \bar{\boldsymbol{T}}(t) \cdot \boldsymbol{U}^{\perp}$, where $\bar{\boldsymbol{T}}(t)$ is the parallel transport operator in Eq. (2.4.11). Considering optimization only in column space, an illustration of the local presentation of tangents is given in Fig. 2.4.2. The local presentation of the $-\nabla f_k$ is $\boldsymbol{D}_{f_k}$, and it is constant in the transported basis $\boldsymbol{U}_k^{\perp}(t)$ in the tangent space $\mathcal{T}_{k+1}$. Hence, in the basis $\boldsymbol{U}_k^{\perp}(t)$, the local presentation of the two quantities $\boldsymbol{S}_{k+1}$ and $\boldsymbol{Y}_{k+1}$ are the same with

the expressions in Euclidean space. The local presentation of the approximated Hessian is also proven [67] to be constant in local coordinate. Consequently, despite one has to parallel transfer the basis $\boldsymbol{U}^{\perp}$ at each iteration, the BFGS algorithm on Grassmann manifold in local coordinates is the same with the BFGS approach in Euclidean space.

Considering optimization in column and row space alternatively, the global parallel transport operator $\bar{\boldsymbol{T}}(t)$ is not valid in local coordinates. In other words, the local presentation of the decent direction $\boldsymbol{D}_{\triangle}$ shown in Fig. 2.4.2 is unknown. Here, we focus on the parallel transport between two arbitrary tangent spaces in local coordinates.

Lemma 2.4.3. *Recall the settings in Section 2.4.3, let* $\boldsymbol{U}_k \in \mathcal{U}_{m,r}$ *and* $\bar{\boldsymbol{U}}_k \in \mathcal{U}_{m,r}$ *span the same subspace, where* $\bar{\boldsymbol{U}}_k = \boldsymbol{U}_k\boldsymbol{V}_{k+1}$ *for* $\boldsymbol{V}_{k+1} \in \mathcal{U}_{r,r}$ *and* $k \in \{1, 2\}$. *Let* $\bar{\boldsymbol{U}}_1 = \tau(\bar{\boldsymbol{U}}_0)$, *where* τ *denote the standard parallel transport operator* $\bar{\boldsymbol{T}}$. *Denote* $\left\{\boldsymbol{b}_1^{(0)}, \cdots, \boldsymbol{b}_{r(m-r)}^{(0)}\right\} \subset \mathbb{R}^{mr}$ *as a basis of* $\mathcal{N}(\boldsymbol{U}_0)$, *then the basis of* $\mathcal{N}(\boldsymbol{U}_1)$ *is* $\left\{\boldsymbol{b}_1^{(1)}, \cdots, \boldsymbol{b}_{r(m-r)}^{(1)}\right\} \subset \mathbb{R}^{mr}$, where $\boldsymbol{b}_i^{(1)} = (\boldsymbol{V}_2 \otimes \boldsymbol{I}_m)\tau((\boldsymbol{V}_1 \otimes \boldsymbol{I}_m)\boldsymbol{b}_i^{(0)})$ for $i \in [1, r(m-r)]$.

The proof of this lemma is given in Subsection 2.6.6.

We investigate the parallel transport of the tangent basis $\boldsymbol{U}^{\perp}$ between two arbitrary tangent spaces in local coordinates. Hence, it is easy to extend the Algorithm 2.2 to local coordinates and the computational cost is deduced.

2.5 Performance Study

In our numerical study, the low-rank matrix $\boldsymbol{X}$ and the index set Ω are uniformly randomly generated. In particular, if we decompose the low-rank matrix into $\boldsymbol{X} = \boldsymbol{U}_{\boldsymbol{X}}\boldsymbol{S}_{\boldsymbol{X}}\boldsymbol{V}_{\boldsymbol{X}}^T$, then $\boldsymbol{U}_{\boldsymbol{X}} \in \mathcal{U}_{m,r}$ and $\boldsymbol{V}_{\boldsymbol{X}} \in \mathcal{U}_{n,r}$ are uniformly dis-

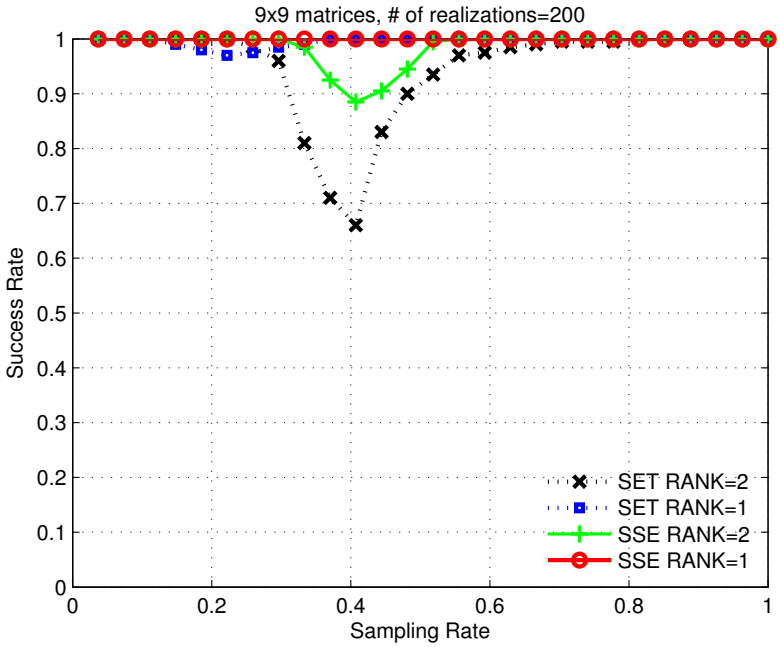


Figure 2.5.1: Performance improvement of SSE compared with SET.

tributed, and diag $(\boldsymbol{S_X}) \in \mathbb{R}^r$ is Wishart distributed. See [21] for more details on this random matrix model and its advantages. The index set Ω is randomly generated from the uniform distribution over $\{\Omega' \subset [m] \times [n] : |\Omega'| \leq K\}$ for some constant K. The sampling rate for a given observation matrix $\boldsymbol{X}_\Omega$ is defined as $|\Omega| / (m \times n)$.

Two scenarios are considered: the noiseless and noisy cases on $\boldsymbol{X}$. Let $\hat{\boldsymbol{X}}$ be the estimated matrix output by a particular algorithm. In the noiseless case, the criterion for a successful completion is that rank $\left(\hat{\boldsymbol{X}}_\Omega\right) \leq r$ and $\left\|\boldsymbol{X}_\Omega - \hat{\boldsymbol{X}}_\Omega\right\|_F^2 \leq \epsilon_e \left\|\boldsymbol{X}_\Omega\right\|_F^2$, where the error tolerance constant $\epsilon_e \geq 0$ is ideally zero but set to 10^{-6} in practice. In the noisy case, the observation matrix is given by $\boldsymbol{X}_\Omega = \mathcal{P}_\Omega(\boldsymbol{X}') + \mathcal{P}_\Omega(\boldsymbol{Z})$ where $\boldsymbol{X}' \in \mathbb{R}^{m \times n}$ is a random low-rank matrix and $\boldsymbol{Z} \in \mathbb{R}^{m \times n}$ is randomly generated from the standard Gaussian random matrix with proper variance. Let ϵ_n denote the ratio

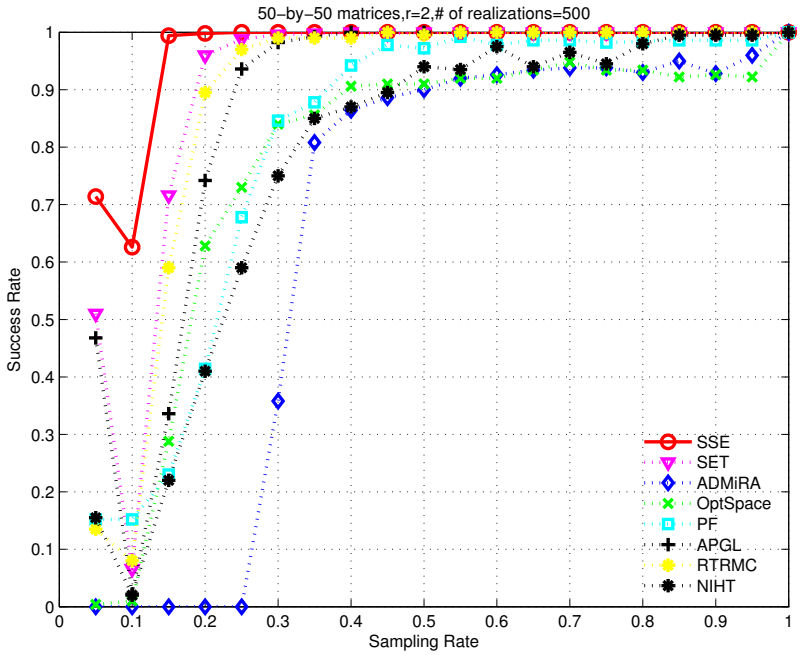


Figure 2.5.2: Performance comparison: noiseless case.

$\|\boldsymbol{Z}_\Omega\|_F^2 / \|\boldsymbol{X}'_\Omega\|_F^2$. Then the signal-to-noise ratio (SNR) of $\boldsymbol{X}_\Omega$ is defined as $10\log_{10}\frac{1}{\epsilon_n}$dB. The criterion for a successful completion is that rank $\left(\hat{\boldsymbol{X}}_\Omega\right) \leq r$ and $\left\|\boldsymbol{X}_\Omega - \hat{\boldsymbol{X}}_\Omega\right\|_F^2 \leq \epsilon_n \|\boldsymbol{X}_\Omega\|_F^2$. The success ratio is the rate between the total number of successful completion and the number of realizations.

In the first experiment, we compare numerical performance of the proposed algorithm SSE and SET in noiseless case using 9-by-9 matrices. The number of realizations is 200 under each sampling Rate. The results in both rank one and rank two cases are shown in Fig. 2.5.1. Higher success rate indicates better numerical performance. The improvement of the numerical performance is significant. For rank one case, SSE can consistently complete all the 9-by-9 matrices under any sampling rate.

In the second experiment, we compare the proposed SSE with 7 benchmark algorithms, i.e., SET [21], ADMiRA [53], OptSpace [47], PF [39], APGL [71],

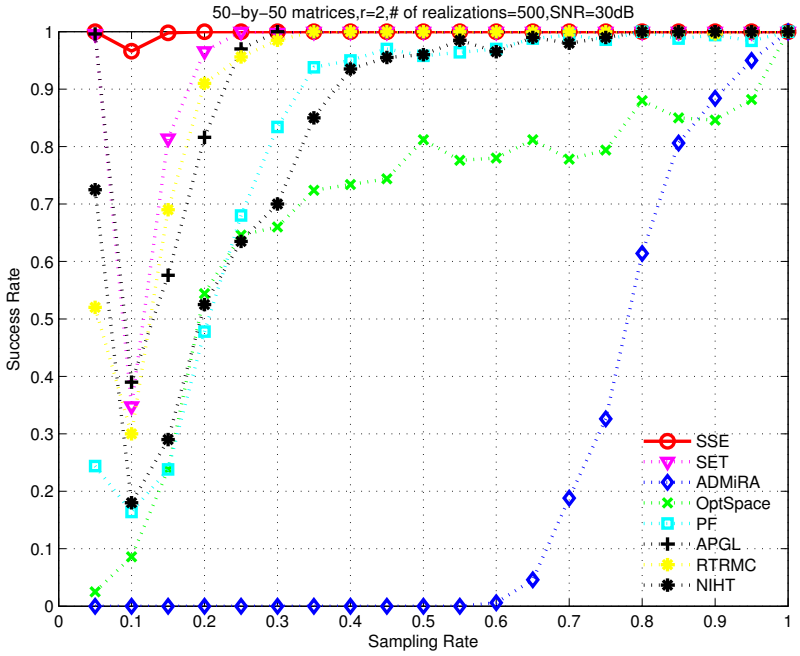


Figure 2.5.3: Performance comparison: noisy case.

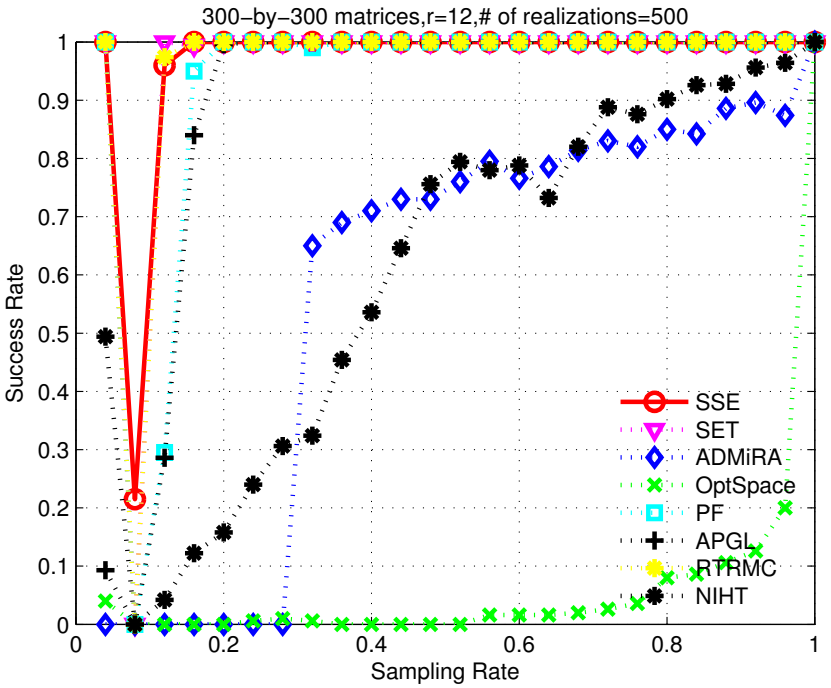


Figure 2.5.4: Performance comparison: large matrix case.

RTRMC [11] and recently proposed NIHT [70]. A set of 50-by-50 rank two matrices were used in both noiseless and noisy cases. The number of iterations is set to 5000 for all tested algorithms except the SET method [21]: as explained by the authors of [21], the computational complexity of each iteration in the SET algorithm is significantly larger than that in other methods; we follow the authors' suggestion and set the number of iterations to 500 for the SET method. We present results in Fig. 2.5.2 and 2.5.3. One interesting scenario is when the sampling number p of the matrix is close to its *degree of freedom*, all the tested algorithms exhibit bad performance (zero success rate). However, the proposed algorithm SSE ties the success bound of p to the degree of freedom. Simulation results clearly demonstrate the performance improvement of the proposed method for both noiseless and noisy cases. We also interested in testing the proposed method using large matrix with a higher rank, i.e., a set of 300-by-300 matrices with rank=12. The results are presented in Fig. 2.5.4. As one can see from the curves, the proposed method has good performance especially when the sampling number is close to the oracle rate. In particular, the oracle rate is 7056 and the SSE method has the best performance at sampling point 0.08 (sample number is 7200, which is very close to the oracle rate 7056) while all other methods have zero success rate.

2.6 Proofs

2.6.1 Analytical Results for Example 1: Minimizing f_u

To start, we first write the explicit forms for $f_{\mu,1}(\epsilon)$ and $f_{\mu,2}(\epsilon)$, respectively,

$$f_{\mu,1}(\epsilon) = \min_{w_1} \left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} \epsilon \\ \epsilon \end{bmatrix} w_1 \right\|_2^2 + \mu w_1^2,$$
$$f_{\mu,2}(\epsilon) = \min_{w_2} \left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} \sqrt{1-2\epsilon^2} \\ \epsilon \end{bmatrix} w_2 \right\|_2^2 + \mu w_2^2.$$

The optimal w_1 and w_2 can be evaluated by setting the derivative with respect to w_1 and w_2 to zero. It can be verified that

$$w_1^* = \frac{2\epsilon}{2\epsilon^2+\mu}, \text{ and } w_2^* = \frac{\sqrt{1-2\epsilon^2}+\epsilon}{1-\epsilon^2+\mu}.$$

With the explicit values of w_1^* and w_2^*, the values of $f_{\mu,1}(\epsilon)$ and $f_{\mu,2}(\epsilon)$ can be evaluated as follows:

$$f_{\mu,1}(\epsilon) = 2(1-\epsilon w_1^*) = 2 - \frac{4\epsilon^2}{2\epsilon^2+\mu} = \frac{2\mu}{2\epsilon^2+\mu},$$

and

$$\begin{aligned} f_{\mu,2}(\epsilon) &= \left(1-\sqrt{1-2\epsilon^2}w_2^*\right) + (1-\epsilon w_2^*) \\ &= 2 - \frac{1-\epsilon^2+2\epsilon\sqrt{1-2\epsilon^2}}{1-\epsilon^2+\mu} \\ &= 1 + \frac{\mu-2\epsilon\sqrt{1-2\epsilon^2}}{1-\epsilon^2+\mu}. \end{aligned}$$

The first derivatives are computed in the following:

$$f'_{\mu,1}(\epsilon) = -\frac{8\mu\epsilon}{(\mu + 2\epsilon^2)^2},$$

and

$$f'_{\mu,2}(\epsilon) = \frac{2\epsilon\left(\mu - 2\epsilon\sqrt{1-2\epsilon^2}\right)}{(1-\epsilon^2+\mu)^2} - \frac{\frac{2-8\epsilon^2}{\sqrt{1-2\epsilon^2}}}{1-\epsilon^2+\mu}.$$

We shall estimate $f'_\mu(\epsilon)$ for several choices of ϵ. Fix an $\epsilon_0 \in \left(-\frac{1}{4}, 0\right)$. Let $\mu > 0$ be sufficiently small such that $\mu < |\epsilon_0|^3/3 < 0.006$. We shall study the sign of $f'_\mu(\epsilon)$ at 0, $-\sqrt{\mu/2}$, and ϵ_0. At $\epsilon = 0$,

$$f'_\mu(0) = f'_{\mu,1}(0) + f'_{\mu,2}(0) = -\frac{2}{1+\mu} < 0. \tag{2.6.1}$$

At $\epsilon = -\sqrt{\mu/2}$, $f'_{\mu,1} = \sqrt{2}\mu^{-\frac{1}{2}} > 15$ as $\mu < 0.006$. To obtain a lower bound on $f'_{\mu,2}$, note that

$$\begin{aligned}
&\frac{-2\sqrt{\frac{\mu}{2}}\left(\mu + 2\sqrt{\frac{\mu}{2}}\sqrt{1-\mu}\right)}{\left(1-\frac{\mu}{2}+\mu\right)^2} \\
&\overset{(a)}{>} -2\sqrt{\frac{\mu}{2}}\left(\mu + 2\sqrt{\frac{\mu}{2}}\sqrt{1-\mu}\right) \\
&\overset{(b)}{>} -2\sqrt{\mu}\left(\mu + 2\sqrt{\mu}\right) \\
&\overset{(c)}{>} -2\sqrt{\mu}\cdot 3\sqrt{\mu} = -6\mu > -0.05,
\end{aligned}$$

where (a) holds as $\left(1+\frac{\mu}{2}\right) > 1$, (b) follows from that $\frac{\mu}{2} < \mu$ and $1-\mu < \mu$, and (c) holds because $\sqrt{\mu} > \mu$ for all $\mu \in (0,1)$. Furthermore, note that

$$-\frac{\frac{2-4\mu}{\sqrt{1-\mu}}}{1+\frac{\mu}{2}} \overset{(a)}{>} -\frac{2-4\mu}{0.9} > -2.45,$$

where (a) follows from that $\sqrt{1-\mu}\left(1+\frac{\mu}{2}\right) > 0.9$ for all $\mu < 0.006$. As a result,

$$f'_\mu\left(-\sqrt{\mu/2}\right) > 15 - 0.05 - 2.45 > 0. \tag{2.6.2}$$

To estimate $f'_\mu(\epsilon_0)$, we first construct an upper bound on $f'_{\mu,1}(\epsilon_0)$:

$$f'_{\mu,1}(\epsilon_0) = \frac{8\mu(-\epsilon_0)}{(\mu+2\epsilon_0^2)^2} \overset{(a)}{<} \frac{8\mu|\epsilon_0|}{4\epsilon_0^4} = \frac{2\mu}{|\epsilon_0|^3} \overset{(b)}{<} \frac{2}{3},$$

where (a) follows from that $\mu + 2\epsilon_0^2 > 2\epsilon_0^2$ and (b) is due to the assumption $\mu < |\epsilon_0|^3/3$. To obtain an upper bound on $f'_{\mu,2}(\epsilon_0)$, we note that for the assumed ranges of ϵ_0 and μ, one has $-2\epsilon_0\left(\mu - 2\epsilon_0\sqrt{1-2\epsilon_0^2}\right)/\left(1-\epsilon_0^2+\mu\right)^2 < 0$, $\sqrt{1-2\epsilon_0^2} < 1$, $1-\epsilon_0^2+\mu < 1$, and

$$-\frac{2-8\epsilon_0^2}{\sqrt{1-2\epsilon_0^2}\left(1-\epsilon_0^2+\mu\right)} < -2+8\epsilon_0^2 < -\frac{3}{2}.$$

Hence, $f'_{\mu,2}(\epsilon) < -\frac{3}{2}$, and

$$f'_\mu(\epsilon_0) < \frac{2}{3} - \frac{3}{2} < 0. \tag{2.6.3}$$

In summary, we have proved that when $\epsilon_0 \in \left(-\frac{1}{4}, 0\right)$ and $0 < \mu < |\epsilon_0|^3/3$, one has $f'_\mu(0) < 0$, $f'_\mu\left(-\sqrt{\mu/2}\right) > 0$, and $f'_\mu(\epsilon_0) < 0$.

2.6.2 Proof of Theorem 2.3.1

Note that $\tilde{f}$ can be written as a summation of finite many atomic functions, i.e., $\tilde{f} = \sum_i \tilde{f}_i = \sum_i f_i g_{\rho_i}$. The statements in the theorem hold if these statements hold for each atomic function $\tilde{f}_i = f_i g_{\rho_i}$. For notational simplicity, we drop the subscript i and ρ_i in this proof.

We first show that $\tilde{f}$ is continuous for positive ρ, that is, for any convergent sequence $\boldsymbol{U}_k \to \boldsymbol{U}$, one has $\tilde{f}(\boldsymbol{U}_k) \to \tilde{f}(\boldsymbol{U})$. Towards this end, note that for any $\epsilon > 0$, when k is sufficiently large,

$$\begin{aligned}
&\left|\tilde{f}(\boldsymbol{U}_k) - \tilde{f}(\boldsymbol{U})\right| \\
&= |f(\boldsymbol{U}_k)\, g(\lambda_{\min}(\boldsymbol{U}_k)) - f(\boldsymbol{U}_k)\, g(\lambda_{\min}(\boldsymbol{U})) \\
&\quad + f(\boldsymbol{U}_k)\, g(\lambda_{\min}(\boldsymbol{U})) - f(\boldsymbol{U})\, g(\lambda_{\min}(\boldsymbol{U}))| \\
&\leq |f(\boldsymbol{U}_k)| \cdot |g(\lambda_{\min}(\boldsymbol{U}_k)) - g(\lambda_{\min}(\boldsymbol{U}))| \\
&\quad + |f(\boldsymbol{U}_k) - f(\boldsymbol{U})|\, g(\lambda_{\min}(\boldsymbol{U})) \\
&\leq \|\boldsymbol{x}_\Omega\|_2^2\, \epsilon + |f(\boldsymbol{U}_k) - f(\boldsymbol{U})| \cdot |g(\lambda_{\min}(\boldsymbol{U}))|\,.
\end{aligned}$$

We discuss this upper bound for two different cases. Define the set of singular points of f by $\mathcal{S} = \{\boldsymbol{U} \in \mathcal{U}_{m,r} :\ \lambda_{\min}(\boldsymbol{U}_\Omega) = 0\}$. When $\boldsymbol{U} \in \mathcal{U}_{m,r} \backslash \mathcal{S}$, then f is continuous at a neighborhood of $\boldsymbol{U}$. For sufficiently large k, $|f(\boldsymbol{U}_k) - f(\boldsymbol{U})| \cdot |g(\boldsymbol{U})| \leq \epsilon \cdot 1 = \epsilon$. When $\boldsymbol{U} \in \mathcal{S}$, then $g(\boldsymbol{U}) = 0$ and $|f(\boldsymbol{U}_k) - f(\boldsymbol{U})| \cdot |g(\boldsymbol{U})| = 0 < \epsilon$. Combining the above results, one has $\left|\tilde{f}(\boldsymbol{U}_k) - \tilde{f}(\boldsymbol{U})\right| \leq c \cdot \epsilon$ for some fixed constant c when k is sufficiently large. It follows that the function $\tilde{f}$ is continuous.

The second part of this theorem is clear from the fact that $g(\boldsymbol{U}) = 1$ for all $\boldsymbol{U} \in \mathcal{U}_{m,r} \backslash \mathcal{S}$.

The proof of the third part is sketched as follows. We first prove that $\overline{\mathcal{U}_f(c)} \subseteq \mathcal{U}_{\tilde{f}}(c)$ when $\rho_1 = \cdots = \rho_n = 0$. From the continuity of $\tilde{f}$, it is clear that for all positive $\rho > 0$, the level set $\left\{\boldsymbol{U} :\ \tilde{f}_\rho(\boldsymbol{U}) \leq 0\right\}$ is closed. Furthermore, for any given $\rho^{(1)} > \rho^{(2)}$, it holds that $g_{\rho^{(1)}}(\boldsymbol{U}) \leq g_{\rho^{(2)}}(\boldsymbol{U})$ and $\tilde{f}_{\rho^{(1)}}(\boldsymbol{U}) \leq \tilde{f}_{\rho^{(2)}}(\boldsymbol{U})$. It then follows that $\mathcal{U}_{\tilde{f}_{\rho^{(1)}}} \supseteq \mathcal{U}_{\tilde{f}_{\rho^{(2)}}}$. As a result,

$\mathcal{U}_{\tilde{f}}(c) = \lim_{\rho^{(k)}\downarrow 0} \bigcap_{k=1}^{K} \mathcal{U}_{\tilde{f}_{\rho^{(k)}}}(c)$ is a closed set. Combining the fact that $\mathcal{U}_f(c) \subseteq \mathcal{U}_{\tilde{f}}(c)$, it follows that $\overline{\mathcal{U}_f(c)} \subset \mathcal{U}_{\tilde{f}}(c)$.

Then we prove the converse, i.e., $\mathcal{U}_{\tilde{f}}(c) \subseteq \overline{\mathcal{U}_f(c)}$. This part is trivial when $c < 0$. We shall focus on the case that $c \geq 0$. Consider any point $\boldsymbol{U} \in \mathcal{U}_{\tilde{f}}(c) \setminus \overline{\mathcal{U}_f(c)}$. We shall show that there is a sequence $\boldsymbol{U}^{(k)}$, $k = 1, 2, \cdots$, such that $\boldsymbol{U}^{(k)} \in \mathcal{U}_f$ and $\boldsymbol{U}^{(k)} \to \boldsymbol{U}$. By assumption $\boldsymbol{U} \in \mathcal{U}_{\tilde{f}}(c) \setminus \overline{\mathcal{U}_f(c)}$, it is clear that $\tilde{f}(\boldsymbol{U}) \neq f(\boldsymbol{U})$ and $\boldsymbol{U}$ is a singular point. By the construction of $\tilde{f} = f \cdot g$, it is clear that $f(\boldsymbol{U}) > c$ and $g(\boldsymbol{U}) = 0$. Since the columns of $\boldsymbol{U}_\Omega \in \mathbb{R}^{m\times r}$ are linearly dependent, by singular value decomposition, the matrix $\boldsymbol{U}_\Omega$ can be written as $\boldsymbol{U}_\Omega = \sum_{\ell=1}^{z} \lambda_\ell \boldsymbol{u}_\ell \boldsymbol{v}_\ell^T$ where $z < r$ is the number of nonzero singular values, λ_ℓ is the ℓ^{th} nonzero singular value, $\boldsymbol{u}_\ell$ and $\boldsymbol{v}_\ell$ are the corresponding left and right singular vectors respectively. We fix a vector $\boldsymbol{v}_{z+1} \in \mathbb{R}^r$ such that $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_z, \boldsymbol{v}_{z+1}$ are orthonormal. Now define the project residue vector $\boldsymbol{x}_{\Omega,r}$ produced by projecting $\boldsymbol{x}_\Omega$ onto the subspace spanned by $\boldsymbol{U}_\Omega$, i.e., $\boldsymbol{x}_{\Omega,r} = \boldsymbol{x}_\Omega - \boldsymbol{U}_\Omega \boldsymbol{U}_\Omega^\dagger \boldsymbol{x}_\Omega$. By the assumption that $f(\boldsymbol{U}) \neq 0$, $\boldsymbol{x}_{\Omega,r} \neq \boldsymbol{0}$. Normalize it to $\boldsymbol{x}'_{\Omega,r}$ so that $\left\|\boldsymbol{x}'_{\Omega,r}\right\|_2 = 1$. Then it is clear that $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_z, \boldsymbol{x}'_{\Omega,r}$ are orthonormal. Define $\boldsymbol{U}^{(k)} = \boldsymbol{U} + \frac{1}{k}\boldsymbol{x}'_{\Omega,r}\boldsymbol{v}_{z+1}^T$ so that $\boldsymbol{U}_\Omega^{(k)} = \sum_{\ell=1}^{z} \lambda_\ell \boldsymbol{u}_\ell \boldsymbol{v}_\ell^T + \frac{1}{k}\boldsymbol{x}'_{\Omega,r}\boldsymbol{v}_{z+1}^T$. It is straightforward to verify that $f\left(\boldsymbol{U}^{(k)}\right) = 0 \leq c$ and $\boldsymbol{U}^{(k)} \to \boldsymbol{U}$. As a result, $\boldsymbol{U} \in \overline{\mathcal{U}_f}$. The third part of this theorem is therefore proved.

2.6.3 Proof of Proposition 2.3.3

The explicit proof of the claim Eq. (2.3.6) is given in the Appendix A of [21]. The smooth function (2.3.2) is about the minimum singular value of the matrix $\boldsymbol{U}_{\Omega_i} \in \mathbb{R}^{q_i\times r}$. For simplicity, we use a new notation in the following formulas,

where $\boldsymbol{A}_i = \boldsymbol{U}_{\Omega_i}$. We have to calculate the Singular SVD of the matrix $\boldsymbol{A}_i$,

$$\boldsymbol{A}_i = \boldsymbol{U}\Lambda\boldsymbol{V}^T \tag{2.6.4}$$

After T. Papadopoulo and M.I.A Louralis's work [59], we can compute the Jacobian of the SVD,

$$\boldsymbol{U}^T \frac{\partial \boldsymbol{A}_i}{\partial a_{jk}} \boldsymbol{V} = \Omega_{\boldsymbol{U}}^{jk}\Lambda + \frac{\partial \Lambda}{\partial a_{jk}} + \Lambda\Omega_{\boldsymbol{V}}^{jk}, \tag{2.6.5}$$

where a_{jk} refers to the (j,k)-th element of the matrix $\boldsymbol{A}_i$, $\Omega_{\boldsymbol{U}}^{jk} = U^T \frac{\partial U}{\partial a_{jk}}$ and $\Omega_{\boldsymbol{V}}^{jk} = \frac{\partial V^T}{\partial a_{jk}}\boldsymbol{V}$. Since $\Omega_{\boldsymbol{U}}^{jk}$ and $\Omega_{\boldsymbol{V}}^{jk}$ are skew symmetric, all the diagonal elements are zeros. Hence it is easy to obtain the first derivatives with respect to a_{jk} of the r-th singular value,

$$\frac{\partial \lambda_r}{\partial a_{jk}} = u_{jr} v_{kr}. \tag{2.6.6}$$

Then, the first order derivative of the minimum singular value is

$$\nabla\lambda_{min} = \nabla\lambda_r = \frac{\partial \lambda_r}{\partial \boldsymbol{A}_i} = \boldsymbol{U}_{:,r}\boldsymbol{V}_{:,r}^T. \tag{2.6.7}$$

It is easy to proof that

$$\frac{dg_\rho}{d\lambda_{\min}} = \begin{cases} \frac{30}{\rho}\left(\frac{\lambda_{\min}}{\rho}\right)^4 - \frac{60}{\rho}\left(\frac{\lambda_{\min}}{\rho}\right)^3 + \frac{30}{\rho}\left(\frac{\lambda_{\min}}{\rho}\right)^2 & \text{if } \lambda \in (0,\rho)\,, \\ \\ 0 & \text{otherwise.} \end{cases}$$

Hence, the claim of $\nabla g_\rho = \frac{dg_\rho}{d\lambda_{\min}} \cdot \nabla\lambda_{\min}$ is proved.

2.6.4 Proof of Singular Values in Example 2.2.2

For any $\boldsymbol{U} = [\ u_1 \quad u_2\ ]^T$, its singular value is given by using eigenvalue decomposition,

$$\boldsymbol{U}^T\boldsymbol{U} - \lambda^2\boldsymbol{I} = 0$$

where $\boldsymbol{I}$ denotes a identity matrix. Then, $\lambda = \sqrt{u_1^2 + u_2^2}$. In this example, $U_{\Omega_1} = [\ \epsilon \quad \epsilon\ ]^T$, hence $\lambda_1 = \sqrt{\epsilon^2 + \epsilon^2} = \sqrt{2\epsilon^2}$. For $U_{\Omega_2} = [\ \sqrt{1-2\epsilon^2} \quad \epsilon]^T$, hence $\lambda_1 = \sqrt{(1-2\epsilon^2) + \epsilon^2} = \sqrt{1-\epsilon^2}$.

2.6.5 Proof of Lemma 2.4.2

Consider our objective function $f(\boldsymbol{U}) = \|\boldsymbol{X}_\Omega - \mathcal{P}(\boldsymbol{U}\boldsymbol{W})\|_F^2$ and its first derivative $\nabla f(\boldsymbol{U}) = \boldsymbol{X}_r\boldsymbol{U}^T$, we have

$$\nabla f(\boldsymbol{U}\boldsymbol{V}) = \boldsymbol{X}_r(\boldsymbol{V}^T\boldsymbol{W})^T = \boldsymbol{X}r\boldsymbol{W}^T\boldsymbol{V} = \nabla f(\boldsymbol{U}) \cdot \boldsymbol{V}. \tag{2.6.8}$$

Now consider $\bar{\boldsymbol{U}}_{k+1} = \boldsymbol{U}_{k+1}\boldsymbol{V}_2\boldsymbol{V}_1^T$ and Eq. (2.6.8), it is easy to obtain

$$\nabla f(\boldsymbol{U}_{k+1}) = \nabla f(\bar{\boldsymbol{U}}_{k+1})\boldsymbol{V}_1\boldsymbol{V}_2^T$$

This equation still holds when we subtract $\triangle_1$ from $\nabla f(\boldsymbol{U}_{k+1})$ and $\bar{\triangle}_1$ from $\nabla f(\bar{\boldsymbol{U}}_{k+1})$, which gives

$$\begin{aligned}
\nabla f(\boldsymbol{U}_{k+1}) - \triangle_1 &= (\nabla f(\bar{\boldsymbol{U}}_{k+1}) - \bar{\triangle}_1)\boldsymbol{V}_1\boldsymbol{V}_2^T \\
\nabla f(\boldsymbol{U}_{k+1}) - \triangle_1 &= (\nabla f(\bar{\boldsymbol{U}}_{k+1})\boldsymbol{V}_1\boldsymbol{V}_2^T - \bar{\triangle}_1\boldsymbol{V}_1\boldsymbol{V}_2^T \\
\triangle_1 &= \bar{\triangle}_1\boldsymbol{V}_1\boldsymbol{V}_2^T
\end{aligned}$$

Hence, given an arbitrary tangent vector $\triangle_0 \in \mathcal{T}_{k-1}$ and consider $\bar{\triangle}_1 =$

$\bar{\boldsymbol{T}}_{k-1}\triangle_0$, its parallel vector in tangent space $\mathcal{T}_{k+1}$ is then

$$\mathcal{T}_{k+1} \ni \triangle_1 = \bar{\boldsymbol{T}}_{k-1} \cdot \triangle_0 \cdot \boldsymbol{V}_1 \boldsymbol{V}_2^T.$$

2.6.6 Proof of Lemma 2.4.3

Given the objective function at $\boldsymbol{U}_0$, $f(\boldsymbol{U}_0) = \min_{\boldsymbol{W}} \|\boldsymbol{X}_\Omega - \mathcal{P}_\Omega(\boldsymbol{U}_0\boldsymbol{W})\|_F^2$. One can obtain its gradient $G_0 = \nabla f(\boldsymbol{U}_0) = -2\boldsymbol{X}_r\boldsymbol{W}_{U_0}^T$, where $\boldsymbol{X}\boldsymbol{r} = \boldsymbol{X}_\Omega - \mathcal{P}_\Omega(\boldsymbol{U}\boldsymbol{W}_{U_0})$. Consequently, it is easy to find

$$f(\bar{\boldsymbol{U}}_0) = f(\boldsymbol{U}_0\boldsymbol{V}_1) = \min_{\boldsymbol{W}} \left\|\boldsymbol{X}_\Omega - \mathcal{P}_\Omega(\boldsymbol{U}_0\boldsymbol{V}_1\boldsymbol{V}_1^T\boldsymbol{W})\right\|_F^2$$

and the gradient $\bar{\boldsymbol{G}}_0$ is

$$\begin{aligned}
\bar{\boldsymbol{G}}_0 &= \nabla f(\bar{\boldsymbol{U}}_0) \\
&= \nabla f(\boldsymbol{U}_0\boldsymbol{V}_1) \\
&= -2\boldsymbol{X}_r(\boldsymbol{V}_1^T\boldsymbol{W}_{U_0})^T \\
&= -2\boldsymbol{X}_r\boldsymbol{W}_{U_0}^T\boldsymbol{V}_1 \\
&= \boldsymbol{G}_0\boldsymbol{V}_1
\end{aligned}$$

The gradient $\boldsymbol{G}_0$ can be decomposed as a sum of its basis with different

weights $vec(\boldsymbol{G}_0) = \sum_{i=1}^{r(m-r)} \boldsymbol{b}_i^{(0)} c_i$, then

$$\begin{aligned} \bar{\boldsymbol{G}}_0 &= \boldsymbol{G}_0 \boldsymbol{V}_1 \\ &= (\sum_{i=1}^{r(m-r)} \boldsymbol{B}_i^{(0)} c_i) \boldsymbol{V}_1 \\ &= \sum_{i=1}^{r(m-r)} \boldsymbol{B}_i^{(0)} \boldsymbol{V}_1 c_i \\ &= \sum_{i=1}^{r(m-r)} \bar{\boldsymbol{B}}_i^{(0)} c_i, \end{aligned} \tag{2.6.9}$$

where $\boldsymbol{B}_i^{(0)} \in \mathbb{R}^{m \times r}$ is the matrix form of $\boldsymbol{b}_i^{(0)} \in \mathbb{R}^{mr}$. Hence,

$$\bar{\boldsymbol{b}}_i^{(0)} = (\boldsymbol{V}_1^T \otimes \boldsymbol{I}_m) \boldsymbol{b}_i^{(0)} \; i \in [1, ..., r(m-r)]. \tag{2.6.10}$$

Applying the above results to $\boldsymbol{U}_1 \in \mathcal{U}_{m,r}$ and $\bar{\boldsymbol{U}}_1 \in \mathcal{U}_{m,r}$, we can drive

$$\boldsymbol{b}_i^{(1)} = (\boldsymbol{V}_2 \otimes \boldsymbol{I}_m) \bar{\boldsymbol{b}}_i^{(1)} \; i \in [1, ..., r(m-r)]. \tag{2.6.11}$$

According to Eq. (2.6.10), Eq. (2.6.11) and $\bar{\boldsymbol{U}}_1 = \tau(\bar{\boldsymbol{U}}_0)$, the relation between $\boldsymbol{b}_i^{(1)}$ and $\boldsymbol{b}_i^{(0)}$ is

$$\begin{aligned} \boldsymbol{b}_i^{(1)} &= (\boldsymbol{V}_2 \otimes \boldsymbol{I}_m) \bar{\boldsymbol{b}}_i^{(1)} \\ &= (\boldsymbol{V}_2 \otimes \boldsymbol{I}_m) \tau \left(\bar{\boldsymbol{b}}_i^{(0)} \right) \\ &= (\boldsymbol{V}_2 \otimes \boldsymbol{I}_m) \tau ((\boldsymbol{V}_1^T \otimes \boldsymbol{I}_m) \boldsymbol{b}_i^{(0)}). \end{aligned}$$

Chapter 3

Blind Source Separation

3.1 Introduction

Blind source separation has been investigated during the last two decades. Early studies focus on the instantaneous and (over-)determined BSS problem, and address the problem under the framework of independent component analysis (ICA) [43], assuming that the sources are statistically independent. This has led to some well-known approaches, such as Infomax [7], maximum likelihood estimation [33], the maximum a posterior (MAP) [8], and FastICA [43]. Convolutive and/or underdetermined BSS problems have also been extensively studied especially in the speech processing applications, where the sensor measurements are usually modeled as convolutive (often underdetermined) mixtures of the original sources due to the presence of room reverberations (and often more sources than sensors). Effort in this direction has led to algorithms such as degenerate unmixed estimation technique (DUET) [45], non-negative matrix factorization (NMF) [52], and sparse representation technique [85].

In this Chapter, we focus on blind image separation application, in which

the instantaneous model is usually adopted. To address this problem, several approaches have been proposed in the literature, including, for example, the Bayesian approaches based on Markov random field model (MRF) [46], sparse component analysis (SCA) [37] and morphological component analysis (MCA) [69] based on sparse representations. In MCA, source separation is addressed by decomposing the images into different morphological components in terms of sparsity of each component in a signal dictionary. The MCA has also been extended to multichannel case as multichannel MCA (MMCA) [9] and generalized MCA (GMCA) [10]. In MMCA, each source is assumed to be sparse in a specific transform domain. However, in GMCA, each source can be represented by the linear combination of morphological components and each component has a sparse representation by a specific dictionary. Recently, MMCA is further adapted to Blind MMCA (BMMCA) [3] based on learned dictionary for separating mixed images. This method is motivated by the idea of image denoising using a learned dictionary from corrupted image in [30], which in principle extends the denoising problem to BSS. The BMMCA method is interesting in that the dictionary is directly trained from the mixtures, alleviating the issue of requiring training data, and as a result the algorithm can still perform in a blind manner. However, the BMMCA method trains multiple dictionaries for different sources, and in each iteration only updates one atom, rendering a potentially ineffective sparse representation of the image sources and a computationally inefficient procedure.

In this Chapter, we propose a new method, termed *SparseBSS*, which not only addresses the above limitations but also has some interesting new properties (discussed below). The implementation is based on simultaneous codeword optimization (SimCO) [23] framework on Grassmann manifolds, which ensures

that the constraints on the column norms of the mixing matrix and dictionaries are satisfied. Numerical experiments for blind image separation show the advantages of SparseBSS over the ICA, GMCA, and BMMCA methods.

The major differences of our proposed algorithm from the existing methods include:

- Different from most dictionary based BSS algorithms where multiple dictionaries are used, we use only one dictionary to sparsely represent different sources. On one hand, this reduces the computational cost. On the other hand, there is no noticeable performance difference between the two approaches when the single dictionary used contains sufficient many codewords (the number of codewords is still less than that of multiple dictionaries combined).

- Formulating the overall separation problem into two sub-problems, we adapt the recently proposed SimCO optimization method [23] to solve both. The advantage of unifying the two stages is that, in practice, the same algorithm framework and codes can be used for both stages, thus significantly reducing the implementation effort.

- Another important reason to adapt the SimCO framework is to alleviate the possible ill-convergence problem existing in the traditional dictionary learning methods, e.g., K-SVD [5] and MOD [31]. In [23], it was observed that singular points, rather than the local minima, tend to be the major obstacle preventing algorithm from converging to a global minimizer. By adopting regularized SimCO, we are able to force the search path away from singular points and improve the performance.

- Also, we investigated the smoothed technique in Chapter 2 to solve the

singular issue in SimCO, termed *smoothed SimCO*. Similarly, a continuous objective function is proposed to replace the original one.

The remainder of this Chapter is organised as follows. Section 3.2 introduces the background of the BSS problem. Section 3.3 describes the framework of the BSS problem based on dictionary learning. A proposed algorithm SparseBSS is introduced and compared in details with the related benchmark algorithm BMMCA. In Section 3.4, we briefly introduce the background of dictionary learning algorithms and then discuss the important observation of the singularity issue, which is a major reason for the failure of dictionary learning algorithms and hence dictionary learning based BSS algorithms. Afterwards, two available approaches are presented to address this problem. We conclude our work in the last Subsection.

3.2 Background

Typically a linear mixture model is assumed where the mixtures $\boldsymbol{Z} \in \mathbb{R}^{r\times N}$ are described as $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{V}$. Each row of $\boldsymbol{S} \in \mathbb{R}^{s\times N}$ is a source and $\boldsymbol{A} \in \mathbb{R}^{r\times s}$ models the linear combinations of the sources. The matrix $\boldsymbol{V} \in \mathbb{R}^{r\times N}$ represents additive noise or interference introduced during mixture acquisition and transmission.

Usually, in the BSS problem, the only known information is the mixtures $\boldsymbol{Z}$ and the number of sources. One needs to determine both the mixing matrix $\boldsymbol{A}$ and the sources $\boldsymbol{S}$, i.e., mathematically, one needs to solve

$$\min_{\boldsymbol{A},\boldsymbol{S}} \|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\|_F^2 .$$

Such a problem has an infinite number of solutions, i.e., the problem is ill-posed. In order to find the true sources and the mixing matrix (subject to scale and permutation ambiguities), it is often required to add extra constraints to the problem formulation.

Sparsity prior is another property that can be used for BSS. Many natural signals are sparse under some dictionaries [17]. The mixtures, viewed as a superposition of sources, are in general less sparse compared to the original sources. Based on this fact, the sparse prior has been used in solving the BSS problem from various perspectives since 2001, e.g., sparse ICA (SPICA) [12] and SCA [37]. In this approach, there is typically no requirement that the original sources have to be independent. As a result, these algorithms are capable of dealing with highly correlated sources, for example, in separating two superposed identical speeches, with one being a few samples delayed version of the other. Jourjine et al., proposed an SCA based algorithm in [45]. SCA algorithms look for a sparse representation under predefined bases such as DCT, wavelet, curvelet, etc. MCA [69] and its extended algorithms for multichannel cases, MMCA [9] and GMCA [10], are also based on the assumption that the original sources are sparse in different bases instead of explicitly constructed dictionaries. However, these algorithms do not exhibit an outstanding performance since in most cases the predefined dictionaries are too general to offer sufficient details of sources when used in sparse representation.

A method to address this problem is to learn data-specific dictionaries. In [29], the author advised to train a dictionary from the mixtures/corrupted-images and then decompose it into a few dictionaries according to the prior knowledge about the main components in different sources. This algorithm is used for separating images with different main frequency components (e.g.,

Cartoon and Texture images) and obtained satisfactory results in image denoising. Peyré et al. proposed in [61] to learn dictionary from a set of exemplar images for each source. Xu et al., [79] proposed an algorithm which allows the dictionaries to be learned from the sources or the mixtures. In most BSS problems, however, dictionaries learned from the mixtures or from similar exemplar images rarely well represent the original sources.

To get more accurate separation results, the dictionaries should be adapted to the unknown sources. The motivation is clear from the assumption that the sources are sparsely represented by some dictionaries. The initial idea of learning dictionaries while separating the sources was suggested by Abolghasemi et al. [3]. They proposed a two-stage iterative process. In this process, each source is equipped with a dictionary, which is learned in each iteration, right after the previous mixture learning stage. Considering the size of dictionaries being much larger than the mixing matrix, the main computational cost is on the dictionary learning stage. This two-stage procedure was further developed in Zhao et al. [82]. The method was termed as SparseBSS, which employs a joint optimization framework based on the idea of SimCO dictionary update algorithm [23]. Furthermore, from the viewpoint of the dictionary redundancy, SparseBSS uses only one dictionary to represent all the sources, and is therefore computationally much more efficient than using multiple dictionaries as in [3]. This joint dictionary learning and source separation framework is the focus of this Chapter.

3.3 Framework of Dictionary Learning Based BSS Problem

We consider the following linear and instantaneous mixing model. Suppose that there are s source signals of the same length, denoted by $\boldsymbol{s}_1,\ \boldsymbol{s}_2,\ \cdots,\ \boldsymbol{s}_s$ respectively, where $\boldsymbol{s}_i \in \mathbb{R}^{1\times N}$ is a row vector to denote the i^{th} source. Assume that these sources are linearly mixed into l observation signals, denoted by $\boldsymbol{z}_1,\ \boldsymbol{z}_2,\ \cdots,\ \boldsymbol{z}_l$ respectively where $\boldsymbol{z}_j \in \mathbb{R}^{1\times N}$. In the matrix format, denote $\boldsymbol{S} = \left[\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, \cdots, \boldsymbol{s}_s^T\right]^T \in \mathbb{R}^{s\times N}$ and $\boldsymbol{Z} = \left[\boldsymbol{z}_1^T, \boldsymbol{z}_2^T, \cdots, \boldsymbol{z}_l^T\right]^T \in \mathbb{R}^{l\times N}$. Then the mixing model is given by

$$\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{V}, \tag{3.3.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{l\times s}$ is the mixing matrix and $\boldsymbol{V} \in \mathbb{R}^{l\times N}$ is denoted as zero mean additive Gaussian noise. We also assume that $l \geq s$, i.e., the under-determined case will not be discussed here.

3.3.1 Separation with Dictionaries Known in Advance

For some BSS algorithms, such as MMCA [9], orthogonal dictionaries $\boldsymbol{D}_i$'s are required to be known a priori. Each source $\boldsymbol{s}_i$ is assumed to be sparsely represented by a different $\boldsymbol{D}_i$. Hence, we have $\boldsymbol{s}_i = \boldsymbol{D}_i\boldsymbol{x}_i$ with $\boldsymbol{x}_i$'s being sparse. Given the observation $\boldsymbol{Z}$ and the dictionaries $\boldsymbol{D}_i$'s, MMCA [9] aims to estimate the mixing matrix and sources, based on the following form:

$$\min_{\boldsymbol{A},\boldsymbol{S}} \|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\|_F^2 + \sum_{i=1}^{n} \lambda_i \left\|\boldsymbol{s}_i \boldsymbol{D}_i^{\dagger}\right\|_1. \tag{3.3.2}$$

Here $\lambda_i > 0$ is the weighting parameter determined by the noise deviation σ, $\|\cdot\|_F$ represents the Frobenius norm, $\|\cdot\|_1$ is the ℓ_1 norm and $\boldsymbol{D}_i^\dagger$ denotes the pseudo-inverse of $\boldsymbol{D}_i$. Predefined dictionaries generated from typical mathematical transforms, e.g., DCT, wavelets and curvelets, do not target to particular sources, and thus do not always provide sufficiently accurate reconstruction and separation results. Elad et al. [29] designed a method first to train a redundant dictionary by K-SVD algorithm in advance, and then decompose it into a few dictionaries, one for each source. This method works well when the original sources have components that are largely different from each other under some unknown mathematical transformations (e.g. Cartoon and Texture images under the DCT transformation). Otherwise the dictionaries found may not be appropriate in the sense that they may fit better to the mixtures rather than the sources.

3.3.2 Separation with Unknown Dictionaries

3.3.2.1 SparseBSS Algorithm Framework

According to my best knowledge, BMMCA and SparseBSS are the two most recently BSS algorithms which implement the idea of performing source separation and dictionary learning simultaneously. We focus on Sparse BSS in this Chapter. In SparseBSS, one assumes that all the sources can be sparsely represented under the same dictionary. In order to obtain enough training samples for dictionary learning, multiple overlapped segments (patches) of the sources are taken. To extract small overlapped patches from the source image $\boldsymbol{s}_i$, a binary matrix $\boldsymbol{P}_k \in \mathbb{R}^{n\times N}$ is defined as a patching operator[1] [82]. The product

[1]Note that in this Chapter $\boldsymbol{P}_k$ is defined as a patching operator for image sources. The patching operator for audio sources can be similarly defined as well.

$\boldsymbol{P}_k \cdot \boldsymbol{s}_i^T \in \mathbb{R}^{n\times 1}$ is needed to obtain and vectorize the kth patch of size $\sqrt{n}\times\sqrt{n}$ taken from image $\mathcal{S}_i$. Denote $\boldsymbol{P} = [\boldsymbol{P}_1, ..., \boldsymbol{P}_K] \in \mathbb{R}^{n\times KN}$, where K is the number of patches taken from each image. Then the extraction of multiple sources $\boldsymbol{S}$ is defined as $\mathcal{P}\boldsymbol{S} = ([\boldsymbol{P}_1, ..., \boldsymbol{P}_K])(\left[\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, ..., \boldsymbol{s}_s^T\right] \otimes \boldsymbol{I}_K) = \boldsymbol{P} \cdot (\boldsymbol{S}^T \otimes \boldsymbol{I}_K) \in \mathbb{R}^{n\times Ks}$, where symbol $\otimes$ denotes the Kronecker product and $\boldsymbol{I}_K$ indicates the identity matrix. The computational cost associated with converting from images to patches is low. Each column of $\mathcal{P}\boldsymbol{S}$ represents one vectorized patch. We sparsely represent $\mathcal{P}\boldsymbol{S}$ by using only one dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times d}$ and a sparse coefficient matrix $\boldsymbol{X} \in \mathbb{R}^{d\times Ks}$, which suggests $\mathcal{P}\boldsymbol{S} \approx \boldsymbol{D}\boldsymbol{X}$. This is different from BMMCA, where multiple dictionaries are used for multiple sources.

With these notations, the BSS problem is formulated as the following joint optimization problem

$$\min_{\boldsymbol{A},\boldsymbol{S},\boldsymbol{D},\boldsymbol{X}} \lambda \left\|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\right\|_F^2 + \left\|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S}\right\|_F^2 . \tag{3.3.3}$$

The parameter λ is introduced to balance the measurement error and the sparse approximation error, and $\boldsymbol{X}$ is assumed to be sparse.

To find the solution of the above problem, we propose a joint optimization algorithm to iteratively update the following two pairs of variables $\{\boldsymbol{D}, \boldsymbol{X}\}$ and $\{\boldsymbol{A}, \boldsymbol{S}\}$ over two stages until a (local) minimizer is found. Note that in each stage there is only one pair of variables to be updated simultaneously by keeping the other pair fixed.

- Dictionary learning stage

$$\min_{\boldsymbol{D},\boldsymbol{X}} \left\|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S}\right\|_F^2 , \tag{3.3.4}$$

- Mixture learning stage

$$\min_{\boldsymbol{A},\boldsymbol{S}} \lambda \left\| \boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S} \right\|_F^2 + \left\| D\boldsymbol{X} - \mathcal{P}\boldsymbol{S} \right\|_F^2 . \tag{3.3.5}$$

Without being explicit in (3.3.3), a sparse coding process is involved where greedy algorithms such as orthogonal matching pursuit (OMP) [60] and subspace pursuit (SP) [20] are used to solve

$$\min_{\boldsymbol{X}} \left\| \boldsymbol{X} \right\|_0 , \text{ s.t. } \left\| \boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S} \right\|_F^2 \leq \epsilon,$$

where $\left\| \boldsymbol{X} \right\|_0$ counts the number of nonzero elements in $\boldsymbol{X}$, the dictionary $\boldsymbol{D}$ is assumed fixed, and $\epsilon > 0$ is an upper bound on the sparse approximation error.

During the optimization, further constraints are made on the matrices $\boldsymbol{A}$ and $\boldsymbol{D}$. Consider the dictionary learning stage. Since the performance is invariant to scaling and permutations of the dictionary codewords (columns of $\boldsymbol{D}$), we follow the convention in the literature, e.g., [23], and enforce the dictionary to be updated on the set

$$\mathcal{D} = \left\{ \boldsymbol{D} \in \mathbb{R}^{n \times d} : \ \left\| \boldsymbol{D}_{:,i} \right\|_2 = 1, \ 1 \leq i \leq d \right\}, \tag{3.3.6}$$

where $\boldsymbol{D}_{:,i}$ stands for the i^{th} column of $\boldsymbol{D}$. A detailed description of the advantage by adding this constraint can be found in [23]. Sparse coding, once performed, provides the information about which elements of $\boldsymbol{X}$ are zeros and which are non-zeros. Define the sparsity pattern by $\Omega = \{(i,j) : \ \boldsymbol{X}_{i,j} \neq 0\}$, which is the index set of the nonzero elements of $\boldsymbol{X}$. Define $\mathcal{X}_\Omega$ as the set of all matrices conforming to the sparsity pattern Ω. This is the feasible set of

the matrix $\boldsymbol{X}$. The optimization problem for the dictionary learning stage can be written as

$$\begin{aligned}\min_{\boldsymbol{D}\in\mathcal{D}} f_\mu(\boldsymbol{D}) = & \min_{\boldsymbol{D}\in\mathcal{D}} \min_{\boldsymbol{X}\in\mathcal{X}_\Omega} \|\boldsymbol{D}\boldsymbol{X}-\mathcal{P}\boldsymbol{S}\|_F^2+\mu\|\boldsymbol{X}\|_F^2, \\ = & \min_{\boldsymbol{D}\in\mathcal{D}} \min_{\boldsymbol{X}\in\mathcal{X}_\Omega} \left\| \begin{bmatrix} \mathcal{P}\boldsymbol{S} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{D} \\ \sqrt{\mu}\boldsymbol{I} \end{bmatrix} \boldsymbol{X} \right\|_F^2 . \end{aligned} \tag{3.3.7}$$

The term $\mu\|\boldsymbol{X}\|_F^2$ introduces a penalty to alleviate the singularity issue. See more details in Section 3.4.3.

In the mixture learning stage, similar to the dictionary learning stage, we constrain the mixing matrix $\boldsymbol{A}$ in the set

$$\mathcal{A}=\left\{\boldsymbol{A}\in\mathbb{R}^{l\times s}:\ \|\boldsymbol{A}_{:,i}\|_2=1,\ 1\le i\le s\right\}. \tag{3.3.8}$$

Otherwise if the mixing matrix $\boldsymbol{A}$ is scaled by a constant c and the source $\boldsymbol{S}$ is inversely scaled by c^{-1}, then for any $\{\boldsymbol{A},\boldsymbol{S}\}$ we can always find a solution $\{c\boldsymbol{A},c^{-1}\boldsymbol{S}|c>1\}$ which further decreases the objective function (3.3.3) from $\lambda\|\boldsymbol{Z}-\boldsymbol{A}\boldsymbol{S}\|_F^2+\|D\boldsymbol{X}-\mathcal{P}\boldsymbol{S}\|_F^2$ to $\lambda\|\boldsymbol{Z}-\boldsymbol{A}\boldsymbol{S}\|_F^2+c^{-2}\|D\boldsymbol{X}-\mathcal{P}\boldsymbol{S}\|_F^2$. Now if we view the sources $\boldsymbol{S}\in\mathbb{R}^{s\times n}$ as a "sparse" matrix with the sparsity pattern $\Omega'=\{(i,j):\ 1\le i\le s,\ 1\le j\le N\}$, then the optimization problem for the mixture learning stage is exactly the same as that for the dictionary learning stage:

$$\begin{aligned}\min_{\boldsymbol{A}\in\mathcal{A}} f_\lambda(\boldsymbol{A}) = & \min_{\boldsymbol{A}\in\mathcal{A}} \min_{\boldsymbol{S}\in\mathbb{R}^{s\times n}} \lambda\|\boldsymbol{Z}-\boldsymbol{A}\boldsymbol{S}\|_F^2+\left\|\mathcal{P}^\dagger(\boldsymbol{D}\boldsymbol{X})-\boldsymbol{S}\right\|_F^2 \\ = & \min_{\boldsymbol{A}\in\mathcal{A}} \min_{\boldsymbol{S}\in\mathcal{X}_{\Omega'}} \left\| \begin{bmatrix} \sqrt{\lambda}\boldsymbol{Z} \\ \mathcal{P}^\dagger(\boldsymbol{D}\boldsymbol{X}) \end{bmatrix} - \begin{bmatrix} \sqrt{\lambda}\boldsymbol{A} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{S} \right\|_F^2 , \end{aligned} \tag{3.3.9}$$

where the fact that $\mathbb{R}^{s\times n} = \mathcal{X}_{\Omega'}$ has been used and $\mathcal{P}^{\dagger}(\cdot)$ is the unpatching operator. Here, we do not require the prior knowledge about the scaling matrix in front of the true mixing matrix [10], as otherwise required in MMCA and GMCA algorithms.

To conclude this Subsection, we emphasize the following treatment of the optimization problems (3.3.7) and (3.3.9). Both of them involve a joint optimization over two variables, i.e., $\boldsymbol{D}$ and $\boldsymbol{X}$ for (3.3.7) and $\boldsymbol{A}$ and $\boldsymbol{S}$ for (3.3.9). Note that if $\boldsymbol{D}$ and $\boldsymbol{A}$ are fixed, then the optimal $\boldsymbol{X}$ and $\boldsymbol{S}$ can be easily computed by solving the corresponding least squares problems. Motivated by this fact, we write (3.3.7) and (3.3.9) as $\min_{\boldsymbol{D}\in\mathcal{D}} f_{\mu}(\boldsymbol{D})$ and $\min_{\boldsymbol{A}\in\mathcal{A}} f_{\lambda}(\boldsymbol{A})$ respectively, when $f_{\mu}(\boldsymbol{D})$ and $f_{\lambda}(\boldsymbol{A})$ are properly defined in (3.3.7) and (3.3.9). In this way, the optimization problems, at least from the surface, only involve one variable. This helps the discovery of the singularity issue and the developments of handling singularity. See Section 3.4 for details.

3.3.2.2 Implementation Details of SparseBSS

The dictionaries at the beginning and the end of the k^{th} iteration, denoted by $\boldsymbol{D}^{(k)}$ and $\boldsymbol{D}^{(k+1)}$ respectively, can be related by $\boldsymbol{D}^{(k+1)} = \boldsymbol{D}^{(k)} + \alpha^{(k)}\boldsymbol{\eta}^{(k)}$ where $\alpha^{(k)}$ is an appropriately chosen step size and $\boldsymbol{\eta}^{(k)}$ is the search direction. The step size $\alpha^{(k)}$ can be determined by *Armijo condition* or *Golden selection* presented in [58]. The search direction $\boldsymbol{\eta}^{(k)}$ can be determined by a variety of gradient methods [28, 58]. The decision of $\boldsymbol{\eta}^{(k)}$ plays the key role which directly affects the convergence rate of the whole algorithm. Generally speaking, a Newton direction is a preferred choice (compared with the gradient descent direction) [58]. In many cases, direct computation of the Newton direction is computationally prohibitive. Iterative methods can be used to

search the Newton direction. Take the Newton Conjugate Gradient (Newton CG) method as an example. It starts with the gradient descent direction $\boldsymbol{\eta}_0$ and iteratively refines it towards the Newton direction. Denote the gradient of $f_\mu(\boldsymbol{D})$ as $\nabla f_\mu(\boldsymbol{D})$. Denote $\nabla_{\boldsymbol{\eta}}(\nabla f_\mu(\boldsymbol{D}))$ as the directional derivative of $\nabla f_\mu(\boldsymbol{D})$ along $\boldsymbol{\eta}$ [41]. In each line search step of the Newton CG method, instead of computing the Hessian $\nabla^2 f_\mu(\boldsymbol{D}) \in \mathbb{R}^{md\times md}$ explicitly, one only needs to compute $\nabla_{\boldsymbol{\eta}}(\nabla f_\mu(\boldsymbol{D})) \in \mathbb{R}^{m\times d}$. The required computational and storage resources are therefore much reduced.

When applying the Newton CG to minimize $f_\mu(\boldsymbol{D})$ in (3.3.7), the key computations are summarized below. Denote $\tilde{\boldsymbol{D}} = [\ \boldsymbol{D}^T \quad \mu\boldsymbol{I}\ ]^T$ and let $\Omega(:,j)$ be the index set of nonzero elements in $\boldsymbol{X}_{:,j}$. We consider $\tilde{\boldsymbol{D}}_i = \tilde{\boldsymbol{D}}_{:,\Omega(:,i)} \in \mathbb{R}^{(m+l)\times l}$ with $m > l$. Matrix $\tilde{\boldsymbol{D}}_i$ is a full column rank tall matrix. We denote

$$f_i(\tilde{\boldsymbol{D}}_i) = \min_{\boldsymbol{x}_i} \|\boldsymbol{y}_i - \tilde{\boldsymbol{D}}\boldsymbol{x}_i\|_2^2$$

and the optimal

$$\boldsymbol{x}_i^* = \arg\min_{\boldsymbol{x}_i} \|\boldsymbol{y}_i - \tilde{\boldsymbol{D}}\boldsymbol{x}_i\|_2^2.$$

Denote $\tilde{\boldsymbol{D}}_i^\dagger$ as the pseudo-inverse of $\tilde{\boldsymbol{D}}_i$. Then we have $\frac{\partial f}{\partial \boldsymbol{x}_i}\ |_{\boldsymbol{x}_i^*} = \boldsymbol{0}$, where $\boldsymbol{x}_i^* = \tilde{\boldsymbol{D}}_i^\dagger \boldsymbol{y}_i$, and $\nabla f_i(\tilde{\boldsymbol{D}}_i)$ can be written as

$$\nabla f_i(\tilde{\boldsymbol{D}}_i) = \frac{\partial f}{\partial \tilde{\boldsymbol{D}}_i} + \frac{\partial f}{\partial \boldsymbol{x}_i}\frac{\partial \boldsymbol{x}_i}{\partial \tilde{\boldsymbol{D}}_i} = -2(\boldsymbol{y}_i - \tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^*)\boldsymbol{x}_i^{*T} + \boldsymbol{0} \tag{3.3.10}$$

To compute $\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\tilde{\boldsymbol{D}}_i)\right)$, we have

$$\begin{aligned}\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\tilde{\boldsymbol{D}}_i)\right) &= 2\nabla_{\boldsymbol{\eta}}\left(\tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^* - \boldsymbol{y}_i\right)\boldsymbol{x}_i^{*T} + 2\left(\tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^* - \boldsymbol{y}_i\right)\nabla_{\boldsymbol{\eta}}\boldsymbol{x}_i^{*T} \\ &= 2\nabla_{\boldsymbol{\eta}}\tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^*\boldsymbol{x}_i^{*T} + 2\tilde{\boldsymbol{D}}_i\nabla_{\boldsymbol{\eta}}\boldsymbol{x}_i^*\boldsymbol{x}_i^{*T} + 2\left(\tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^* - \boldsymbol{y}_i\right)\nabla_{\boldsymbol{\eta}}\boldsymbol{x}_i^{*T} \\ &= 2\boldsymbol{\eta}\boldsymbol{x}_i^*\boldsymbol{x}_i^{*T} + 2\tilde{\boldsymbol{D}}_i\nabla_{\boldsymbol{\eta}}\boldsymbol{x}_i^*\boldsymbol{x}_i^{*T} + 2\left(\tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^* - \boldsymbol{y}_i\right)\nabla_{\boldsymbol{\eta}}\boldsymbol{x}_i^{*T},\end{aligned} \tag{3.3.11}$$

where

$$\nabla_{\boldsymbol{\eta}}\boldsymbol{x}^* = -(\tilde{\boldsymbol{D}}^T\tilde{\boldsymbol{D}})^{-1}\left(\left(\tilde{\boldsymbol{D}}^T\boldsymbol{\eta} + \boldsymbol{\eta}^T\tilde{\boldsymbol{D}}\right)\tilde{\boldsymbol{D}}^\dagger - \boldsymbol{\eta}^T\right)\boldsymbol{y}. \tag{3.3.12}$$

From the definition of $\tilde{\boldsymbol{D}}_i$, $\boldsymbol{D}_i$ is a sub-matrix of $\tilde{\boldsymbol{D}}_i$, therefore $\nabla f_i(\boldsymbol{D}_i)$ and $\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\boldsymbol{D}_i)\right)$ are also respectively sub-matrices of $\nabla f_i(\tilde{\boldsymbol{D}}_i)$ and $\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\tilde{\boldsymbol{D}}_i)\right)$, i.e., $\nabla f_i(\boldsymbol{D}_i) = \left(\nabla f_i(\tilde{\boldsymbol{D}}_i)\right)_{1:m,:}$ and $\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\boldsymbol{D}_i)\right) = \left(\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\tilde{\boldsymbol{D}}_i)\right)\right)_{1:m,:}$.

In addition, it is also worth noting that the SparseBSS model, using one dictionary to sparsely represent all the sources will get almost the same performance as using multiple but same-sized dictionaries when the dictionary redundancy $\frac{d}{n}$ is large enough. As a result it is reasonable to train only one dictionary for all the sources. An obvious advantage for using one dictionary is that the computational cost does not increase when the number of sources increases.

3.3.3 Blind MMCA and Its Comparison to SparseBSS

BMMCA [3] is another recently proposed BSS algorithm based on adaptive dictionary learning. Without knowing dictionaries in advance, BMMCA algorithm also trains dictionaries from the observed mixture $\boldsymbol{Z}$. Inspired by the

hierarchical scheme used in MMCA and the update method in K-SVD, the separation model in BMMCA is made up of a few rank-1 approximation problems, where each problem targets on the estimation of one particular source

$$\min_{\boldsymbol{A}_{:,i},\boldsymbol{s}_i,\boldsymbol{D}_i,\boldsymbol{X}_i} \lambda \left\|\boldsymbol{E}_i - \boldsymbol{A}_{:,i}\boldsymbol{s}_i\right\|_F^2 + \left\|\boldsymbol{D}_i\boldsymbol{X}_i - \mathcal{R}\boldsymbol{s}_i\right\|_2^2 + \mu \left\|\boldsymbol{X}_i\right\|_0 . \tag{3.3.13}$$

Different from the operator $\mathcal{P}$ defined earlier in SparseBSS algorithm, the operator $\mathcal{R}$ in BMMCA is used to take patches from only one estimated image $\boldsymbol{s}_i$. $\boldsymbol{D}_i$ is the trained dictionary for representing source $\boldsymbol{s}_i$. $\boldsymbol{E}_i$ is the residual which can be written as

$$\boldsymbol{E}_i = \boldsymbol{Z} - \sum_{j \neq i} \boldsymbol{A}_{:,j}\boldsymbol{s}_j. \tag{3.3.14}$$

Despite being similar in problem formulation, BMMCA and SparseBSS differ in terms of whether the sources share a single dictionary in dictionary learning. In the SparseBSS algorithm, only one dictionary is used to provide sparse representations for all sources. BMMCA requires multiple dictionaries, one for each source. In the mixing matrix update, BMMCA imitates the K-SVD algorithm by splitting the steps of update and normalization. Such two-step based approach does not bring the expected optimality of $\boldsymbol{A} \in \mathcal{A}$, thereby giving inaccurate estimation, while SparseBSS keeps $\boldsymbol{A} \in \mathcal{A}$ during the optimization process. In BMMCA, the authors claim that the ratio between the parameter λ and the noise standard deviation σ is fixed to 30, which will not guarantee good estimation results at various noise levels.

3.4 Dictionary Learning and the Singularity Issue

As becoming clear from previous discussions, dictionary learning plays an essential role in solving the BSS problem when the sparse prior is used, and hence is the focus of this Section. We firstly briefly introduce the relevant background, then discuss an interesting phenomenon, the singularity issue in the dictionary update stage, and finally present two approaches to handle the singularity issue. For the readers who are more interested in the SparseBSS algorithm itself may consider this Section as optional and skip to Section 3.5.

3.4.1 Brief Introduction of Dictionary Learning Algorithms

One of the earliest dictionary learning algorithms is the method of optimal directions (MOD) [31] proposed by Engan et al. The main idea is as follows: in each iteration, one first fixes the dictionary and uses OMP [60] or FOCUSS [36] to update the sparse coefficients, then fixes the obtained sparse coefficients and updates the dictionary in the next stage. MOD was later modified to iterative least squares algorithm (ILS-DLA) [32] and recursive least squares algorithm (RLS-DLA) [68]. Aharon et al. developed the K-SVD algorithm [5]. In each iteration, the first step is to update the sparse coefficients in the same way as in MOD. Then in the second step, one fixes the sparse pattern, and updates the dictionary and the nonzero coefficients simultaneously. In particular, the codewords in the dictionary are sequentially selected: the selected codeword and the corresponding row of the sparse coefficients are updated simultaneously by using SVD. More recently, Dai et al. [23] considered the dictionary learn-

ing problem from a new perspective. They formulated dictionary learning as an optimization problem on manifolds and developed simultaneous codeword optimization (SimCO) algorithm. In each iteration, SimCO allows multiple codewords of the dictionary to be updated with corresponding rows of the sparse coefficients jointly. This new algorithm can be viewed as a generalization of both MOD and K-SVD. Some other dictionary learning algorithms are also developed in the past decade targeting on various circumstances. For example, based on stochastic approximations, Mairal et al. [54] proposed an online algorithm to address the problem with large data sets.

Theoretical or in-depth analysis of the dictionary learning problem was meantime in progress as well. Gribonval et al. [38], Geng et al. [34] and Jenatton et al. [44] studied the stability and robustness of the objective function under different probabilistic modeling assumptions, respectively. In addition, Dai et al. observed in [23] that the dictionary update procedure may fail to converge to a minimizer. This is a common phenomenon happening in MOD, K-SVD and SimCO. Dai et al. further observed that ill-conditioned dictionaries, rather than stationary dictionaries, are the major reason that has led to the failure of the convergence. To alleviate this problem, regularized SimCO was proposed in [23]. The empirical performance improvement was observed. The same approach was also considered in [80] however without detailed discussion on the singularity issue.

3.4.2 Singularity Issue and Its Impacts

In dictionary update stage of existing mainstream algorithms, singularity is observed as the major reason leading to failures [23, 84]. Simulations in [23] suggests that the mainstream algorithms fail mainly because of singular points

in the objective function rather than non-optimal stationary points. As dictionary learning is an essential part of the aforementioned SparseBSS, the singularity issue also has a negative impact on the overall performance of BSS. To explain the singularity issue in dictionary update, we first formally define the singular dictionaries.

Definition 3.4.1. *A dictionary $\boldsymbol{D} \in \mathbb{R}^{m\times d}$ is singular under a given sparsity pattern Ω if there exists an $i \in [n]$ such that the corresponding sub-dictionary $\boldsymbol{D}_i \triangleq \boldsymbol{D}_{:,\Omega(:,i)}$ is column rank deficient. Or equivalently, the minimum singular value of $\boldsymbol{D}_i$, denoted as $\lambda_{\min}(\boldsymbol{D}_i)$, is zero.*

A dictionary $\boldsymbol{D} \in \mathbb{R}^{m\times d}$ is said to be ill-conditioned under a given sparsity pattern Ω if there exists an $i \in [n]$ such that the condition number of the sub-dictionary $\boldsymbol{D}_i$ is large, or equivalently $\lambda_{\min}(\boldsymbol{D}_i)$ is close to zero.

Definition 3.4.2. (*[23]*) *Define the condition number of a dictionary $\boldsymbol{D}$* as:

$$\kappa(\boldsymbol{D}) = \max_{i\in[n]} \frac{\lambda_{\max}(\boldsymbol{D}_i)}{\lambda_{\min}(\boldsymbol{D}_i)},$$

where $\lambda_{\max}(\boldsymbol{D}_i)$ and $\lambda_{\min}(\boldsymbol{D}_i)$ represent the maximum and the minimum singular value of the sub-dictionary $\boldsymbol{D}_i$ respectively.

The word "singular" comes from the fact that $f(\boldsymbol{D}) = \min_{\boldsymbol{X}\in\mathcal{X}_\Omega} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2$

is not continuous at a singular dictionary[2] and the corresponding

$$\boldsymbol{X}\left(\boldsymbol{D}\right) \triangleq \arg\min_{\boldsymbol{X}\in\mathcal{X}_{\Omega}} \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2$$

is not unique. The singularity of $f\left(\boldsymbol{D}\right)$ leads to convergence problems. Benchmark dictionary update procedures may fail to find a globally optimal solution. Instead they converge to a singular point of $f\left(\boldsymbol{D}\right)$, i.e., a singular dictionary.

Ill-conditioned dictionaries are in the neighborhood of singular ones. Algorithmically when one of the $\lambda_{\min}\left(\boldsymbol{D}_i\right)$s is ill-conditioned, the curvature of $f\left(\boldsymbol{D}\right)$ is quite large and the value of the gradient fluctuates dramatically. This seriously affects the convergence rate of the dictionary update process.

Furthermore, ill-conditioned dictionaries also bring a negative effect on the sparse coding stage. Denote $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ as the i^{th} column of $\boldsymbol{Y}$ and $\boldsymbol{X}$ respectively. Consider a summand of the formulation in sparse coding stage [5, 23], i.e.,

$$\min_{\boldsymbol{x}_i} \|\boldsymbol{y}_i-\boldsymbol{D}\boldsymbol{x}_i\|_F^2+\|\boldsymbol{x}_i\|_0 .$$

An ill-conditioned $\boldsymbol{D}$ corresponds to a very large condition number, which breaks the RIP [15], and results in the unstable solutions: with small perturbations added on the training sample $\boldsymbol{Y}$, the solutions of $\boldsymbol{X}$ deviate significantly.

[2] An illustration: take $\boldsymbol{Y}$, $\boldsymbol{D}$, $\boldsymbol{X}$ as scalars. If $\boldsymbol{Y}\neq 0$, there exists a singular point at $\boldsymbol{D}=0$ on $f\left(\boldsymbol{D}\right)=\min_{\boldsymbol{X}} \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2$, where $\boldsymbol{X}$ can be assigned as any real number.

3.4.3 Regularized SimCO

The main idea of regularized SimCO lies in the use of an additive penalty term to avoid singularity. Consider the objective function $f_\mu\left(\tilde{\boldsymbol{D}}\right)$ in (3.3.7),

$$\begin{aligned} f_\mu\left(\tilde{\boldsymbol{D}}\right) = & \min_{\boldsymbol{X}\in\mathcal{X}_\Omega} \|\boldsymbol{D}\boldsymbol{X}-\mathcal{P}(\boldsymbol{S})\|_F^2+\mu\|\boldsymbol{X}\|_F^2, \\ = & \min_{\boldsymbol{X}\in\mathcal{X}_\Omega} \left\| \begin{bmatrix} \mathcal{P}(\boldsymbol{S}) \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{D} \\ \sqrt{\mu}\boldsymbol{I} \end{bmatrix} \boldsymbol{X} \right\|_F^2. \end{aligned} \tag{3.4.1}$$

As long as $\mu \neq 0$ ($\mu > 0$ in our case), the block $\mu\boldsymbol{I}$ guarantees the full column rank of $\tilde{\boldsymbol{D}} = [\ \boldsymbol{D}^T \quad \mu\boldsymbol{I}\ ]^T$. Therefore, with the modified objective function $f_\mu\left(\tilde{\boldsymbol{D}}\right)$, there is no singular point so that gradient descent methods will only converge to stationary points.

This regularization technique is also applicable to MOD [23]. It is verified that this technique effectively mitigates the occurrence of ill-conditioned dictionary although at the same time some stationary points might be generated. To alleviate this problem, one can decrease gradually the regularization parameter μ during the optimization process [23]. In the end μ will decrease to zero. Nevertheless, it is still not guaranteed to converge to a global minimum. Similar to the regularized method in Chapter 2, the regularized SimCO fails at the singular point. As a result, another method to address the singularity issue is introduced below.

3.4.4 Smoothed SimCO

Also aiming at handling the singularity issue, smoothed SimCO [84] is to remove the singularity effect by multiplicative functions. The intuition is ex-

plained as follows. Write $f(\boldsymbol{D})$ into a summation of atomic functions

$$\begin{aligned} f(\boldsymbol{D}) &= \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 \\ &= \sum_i \|\boldsymbol{Y}_{:,i} - \boldsymbol{D}_i \boldsymbol{X}_{\boldsymbol{\Omega}(:,i),i}\|_2^2 \\ &= \sum_i f_i(\boldsymbol{D}_i), \end{aligned} \tag{3.4.2}$$

where each $f_i(\boldsymbol{D}_i)$ is termed as an atomic function and $\boldsymbol{D}_i$ is defined in Definition *3.4.1*. Let $\mathcal{I}$ be the index set corresponding to the $\boldsymbol{D}_i$'s of full column rank. Define an indicator function $\mathcal{X}_{\mathcal{I}}$ s.t. $\mathcal{X}_{\mathcal{I}}(i) = 1$ if $i \in \mathcal{I}$ and $\mathcal{X}_{\mathcal{I}}(i) = 0$ if $i \in \mathcal{I}^c$. Use $\mathcal{X}_{\mathcal{I}}(i)$ as a multiplicative modulation function and apply it to each $f_i(\boldsymbol{D}_i)$. Then one obtains

$$\bar{f}(\boldsymbol{D}) = \sum_i f_i(\boldsymbol{D}_i)\mathcal{X}_{\mathcal{I}}(i) = \sum_{i \in \mathcal{I}} f_i(\boldsymbol{D}_i). \tag{3.4.3}$$

This new function $\bar{f}$ is actually the best possible lower semi-continuous approximation of f and there is no new stationary point created.

Motivated from the above, we define

$$\tilde{f}(\boldsymbol{D}) = \sum_i f_i(\boldsymbol{D}_i) g_\rho\left(\lambda_{\min}(\boldsymbol{D}_i)\right), \tag{3.4.4}$$

where the shape of $g_\rho(\cdot)$ is given in Fig. 2.3.1. The function $g_\rho(\cdot)$ has the following properties: 1) $g_\rho(\lambda_{\min}) = 0$ for all $\lambda_{\min} \leq 0$; 2) $g_\rho(\lambda_{\min}) = 1$ for all $\lambda_{\min}(\boldsymbol{D}_i) > \delta > 0$, where δ is a threshold; 3) $g_\rho(\cdot)$ is monotonically increasing; 4) $g_\rho(\cdot)$ is second order differentiable. When using $\lambda_{\min}(\boldsymbol{D}_i)$ as the input variable for $g_\rho(\cdot)$ and the positive threshold $\delta \to 0$, $\lambda_{\min}(\boldsymbol{D}_i)$ becomes an

indicator function indicating whether $\boldsymbol{D}_i$ has a full column rank, i.e.,

$$g_\rho\left(\lambda_{\min}\left(\boldsymbol{D}_i\right)\right) = \begin{cases} 1 & \text{if } \boldsymbol{D}_i \text{ has full column rank;} \\ 0 & \text{otherwise.} \end{cases}$$

The modulated objective function $\tilde{f}$ has several good properties which do not exhibit in the regularized objective function (3.4.1). In particular, we have the following theorems.

Theorem 3.4.3. Consider the smoothed objective function $\tilde{f}$ and the original objective function f defined in (3.4.4) and (3.4.2) respectively.

1. *When $\delta > 0$, $\forall i$, $\tilde{f}(\boldsymbol{D})$ is continuous.*
2. *Consider the limit case where $\delta \to 0$ with $\delta > 0$, $\forall i$. The following statements hold:*
   (a) *$\tilde{f}(\boldsymbol{D})$ and $f(\boldsymbol{D})$ differ only at the singular points.*
   (b) *$\tilde{f}(\boldsymbol{D})$ is the best possible lower semi-continuous approximation of $f(\boldsymbol{D})$.*

Theorem 3.4.4. Consider the smoothed objective function $\tilde{f}$ and the original objective function f defined in (3.4.4) and (3.4.2) respectively. For any $a \in \mathbb{R}$, define the lower level set $\mathcal{D}_f(a) = \{\boldsymbol{D} : f(\boldsymbol{D}) \leq a\}$. It is provable that when $\delta \to 0$, $\mathcal{D}_{\tilde{f}}(a)$ is the closure of $\mathcal{D}_f(a)$.

In practice, we always choose a $\delta > 0$. The effect of a positive δ, roughly speaking, is to remove the barriers created by singular points, and replace them with "tunnels", whose widths are controlled by δ, to allow the optimization process to pass through. The smaller the δ is, the better $\tilde{f}$ approximates f, but the narrower the tunnels are, and the slower the convergence rate will be. As a result, the threshold δ should be properly chosen. A detailed discussion of choosing δ is presented in [83]. Compared with the choice of the parameter (μ) in the regularized SimCO [23], the choice of the smoothing threshold δ is easier: one can simply choose a small $\delta > 0$ without decreasing it during the process.

3.4.5 Implementation of Smoothed SimCO

In this Subsection, we present a Newton CG implementation to minimize the objective function $\tilde{f}(\boldsymbol{D})$. Most optimization methods are based on the so called line search strategy. The dictionaries at the beginning and the end of the k-th iteration, denoted by $\boldsymbol{D}^{(k)}$ and $\boldsymbol{D}^{(k+1)}$ respectively, can be related by $\boldsymbol{D}^{(k+1)} = \boldsymbol{D}^{(k)} + \alpha^{(k)}\boldsymbol{\eta}^{(k)}$ where $\alpha^{(k)}$ is the appropriately chosen step size and $\boldsymbol{\eta}_k$ is the search direction. The step size $\alpha^{(k)}$ can be determined by using criteria presented in [58]. The search direction $\boldsymbol{\eta}^{(k)}$ plays the key role in determining the convergence rate. Generally speaking, a Newton direction is preferred (compared with the gradient descent direction) [58]. In a standard Newton method, the computation of the Newton direction requires the Hessian of the objective function. Note that in the problem at hand, the variable $\boldsymbol{D}$ has size $m \times d$ and hence the corresponding Hessian has size $md \times md$. To compute the Hessian explicitly, it requires large computational resource as well as extra-ordinary storage resource. By contrast, Newton CG provides a means

to compute the Newton direction without explicitly computing the Hessian.

More specifically, the Newton CG method starts with the gradient descent direction $\boldsymbol{\eta}_0$ and iteratively refines it towards the Newton direction. The detailed steps in find a good search direction are given in Algorithm 3.1. In each iteration, instead of computing the Hessian $\nabla^2 \tilde{f}$ explicitly, one only needs to compute $\nabla_{\boldsymbol{\eta}} \left(\nabla \tilde{f} \right)$ where $\nabla_{\boldsymbol{\eta}} (\cdot)$ denotes the directional gradient. Note that $\nabla_{\boldsymbol{\eta}} \left(\nabla \tilde{f} \right) \in \mathbb{R}^{m \times d}$. The required computational and storage resources are much less than working with the Hessian directly.

Algorithm 3.1 The Newton CG algorithm: find the search direction.

Input: $\boldsymbol{D}$; **Output**: $\boldsymbol{\eta}$.
Define: $\mathcal{P}(\boldsymbol{\eta}_{:,i}) = (\boldsymbol{I} - \boldsymbol{D}_{:,i}\boldsymbol{D}_{:,i}^T)\boldsymbol{\eta}_{:,i}$.
For $k = 0, 1, 2, ...$
  Define tolerance $\epsilon_k = \min \left(0.5, \sqrt{\left\| \nabla \tilde{f} \right\|} \right) \left\| \nabla \tilde{f} \right\|$.
  Set $\boldsymbol{z}_0 = \boldsymbol{0}$, $\boldsymbol{r}_0 = \nabla \tilde{f}$, $\boldsymbol{d}_0 = -\boldsymbol{r}_0 = -\nabla \tilde{f}$.
  For $j = 0, 1, 2, ...$
    Set $\boldsymbol{H}_j = \nabla_{\boldsymbol{d}_j} \left(\nabla \tilde{f} \right)$.
    $\forall i$, let $(\boldsymbol{H}_j)_{:,i} = \mathcal{P} \left((\boldsymbol{H}_j)_{:,i} \right)$.
    If $tr\left(\boldsymbol{d}_j^T \boldsymbol{H}_j \right) \leqslant 0$
      If $j = 0$
        return $\boldsymbol{\eta} = -\nabla \tilde{f}$.
      else
        return $\boldsymbol{\eta} = \boldsymbol{z}_j$.
  Set $\alpha_j = tr\left(\boldsymbol{r}_j^T \boldsymbol{r}_j \right) / tr\left(\boldsymbol{d}_j^T \boldsymbol{H}_j \right)$.
  Set $\boldsymbol{r}_{j+1} = \boldsymbol{r}_j + \alpha_j \boldsymbol{H}_j$.
  If $\|\boldsymbol{r}_{j+1}\| < \epsilon_k$
    return $\boldsymbol{\eta} = \boldsymbol{z}_{j+1}$.
  Set $\beta_{j+1} = tr\left(\boldsymbol{r}_{j+1}^T \boldsymbol{r}_{j+1} \right) / tr\left(\boldsymbol{r}_j^T \boldsymbol{r}_j \right)$.
  Set $\boldsymbol{d}_{j+1} = -\boldsymbol{r}_{j+1} + \beta_{j+1} \boldsymbol{d}_j$.
end
$\forall i$, let $\boldsymbol{\eta}_{:,i} = \mathcal{P}(\boldsymbol{\eta}_{:,i})$.

Here, we focus on the computation of $\nabla \tilde{f}$. By the linearity of the differen-

tiation, it holds that

$$\nabla \tilde{f} = \sum_i \left(\nabla f_i\right) \cdot g_{\boldsymbol{\delta}_i} + f_i \cdot \left(\nabla g_{\boldsymbol{\delta}_i}\right).$$

Denote $\boldsymbol{D}_{:,\Omega(:,i)}$ by $\boldsymbol{D}_i$. Then it can be verified that

$$\nabla f_i\left(\boldsymbol{D}_i\right) = -2\left(\boldsymbol{y}_i - \boldsymbol{D}_i \boldsymbol{x}_i^*\right) \boldsymbol{x}_i^{*T}. \tag{3.4.5}$$

where $\boldsymbol{y}_i$ is the i^{th} column of $\boldsymbol{Y}$ and the corresponding optimal $\boldsymbol{x}_i^*$ is defined via $\boldsymbol{x}_i^* = \boldsymbol{D}_i^{\dagger} \boldsymbol{y}_i$. For the g function, it holds that when $\lambda_{\min}\left(\boldsymbol{D}_i\right) \neq 0$ and $\lambda_{\min}\left(\boldsymbol{D}_i\right)$ is not repetitive (all other singular values are different from $\lambda_{\min}$),

$$\nabla g_{\boldsymbol{\delta}_i}\left(\boldsymbol{D}_i\right) = \frac{dg_{\boldsymbol{\delta}_i}}{d\lambda_{\min}} \cdot \nabla \lambda_{\min}\left(\boldsymbol{D}_i\right) = \frac{dg_{\boldsymbol{\delta}_i}}{d\lambda_{\min}} \cdot \left(\boldsymbol{u}_{\min,i} \boldsymbol{v}_{\min,i}^T\right),$$

where $\boldsymbol{u}_{\min,i}$ and $\boldsymbol{v}_{\min,i}$ are the left and right singular vectors corresponding to the minimum singular value of $\boldsymbol{D}_i$ respectively.

Remark 3.4.5. If $\lambda_{\min}\left(\boldsymbol{D}_i\right) = 0$ or $\lambda_{\min}\left(\boldsymbol{D}_i\right)$ is repetitive, then $\nabla \lambda_{\min}\left(\boldsymbol{D}_i\right)$ is not well defined. However this happens with probability zero when the dictionary is randomly generated from the uniform distribution on $\mathcal{D}$. Furthermore, even this happens during the optimization procedure, directly applying $\nabla \lambda_{\min}\left(\boldsymbol{D}_i\right) = \boldsymbol{u}_{\min,i} \boldsymbol{v}_{\min,i}^T$ does not introduce any practical issue in our simulations.

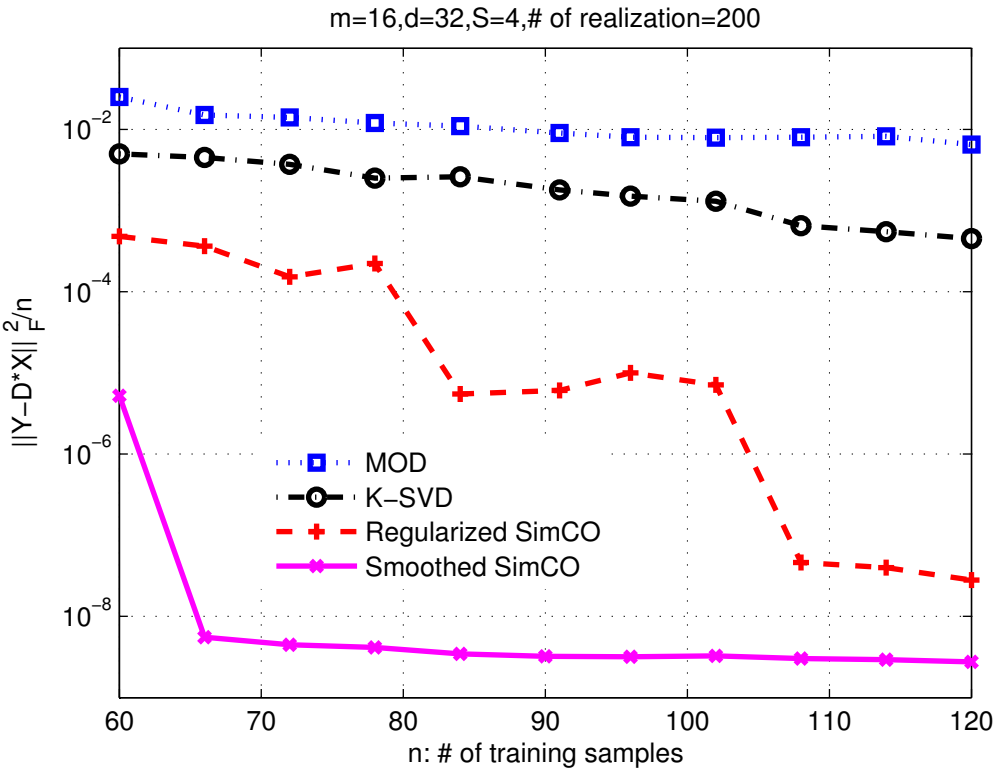
3.5 Algorithm Testing on Practical Applications

3.5.1 Empirical Tests for Smoothed SimCO

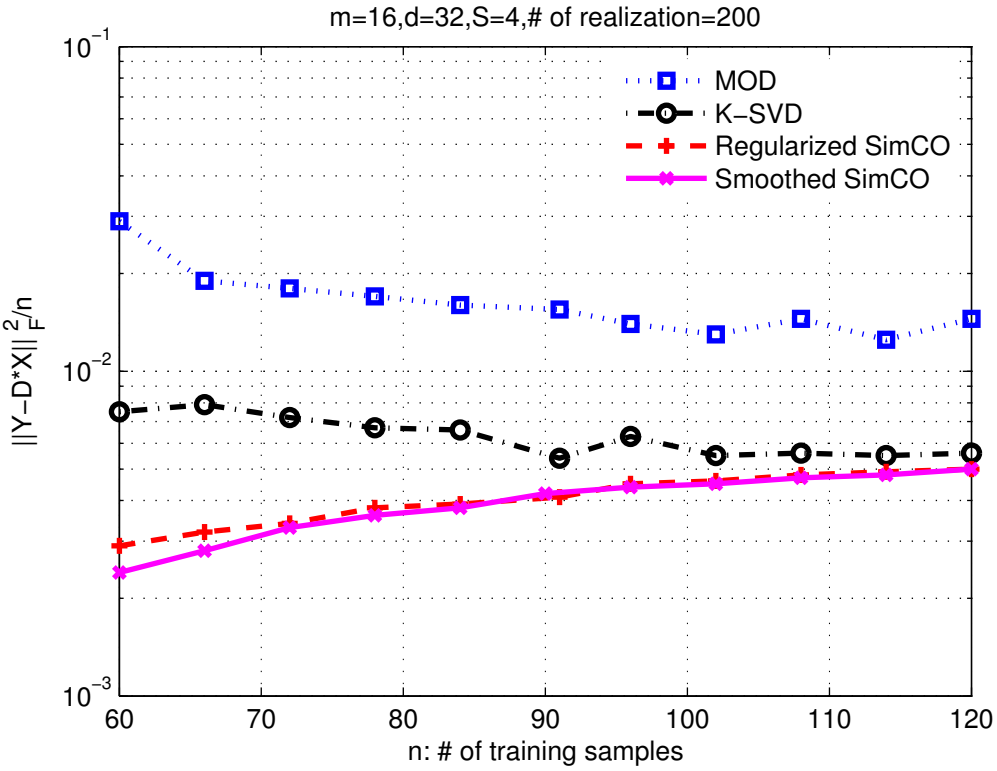
The settings for the numerical tests are as follows. The training samples are generated according to $\boldsymbol{Y} = \boldsymbol{D}_{\text{true}}\boldsymbol{X}_{\text{true}} + \boldsymbol{W}$ where $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ are Gaussian noise ($\boldsymbol{W} = \boldsymbol{0}$ for the noiseless case). The dictionary $\boldsymbol{D}_{\text{true}}$ is randomly generated from the uniform distribution on $\mathcal{D}$. Regarding the sparse coefficients, we assume that each column of $\boldsymbol{X}_{\text{true}}$ contains exactly s many non-zero elements of which the locations are randomly generated from the corresponding uniform distribution. The nonzero elements of $\boldsymbol{X}_{\text{true}}$ are randomly generated from the standard Gaussian distribution. To separate the effect of sparse coding, we also assume that the sparse coding stage is perfect, i.e., the true sparsity pattern Ω_{true} is available.

Both noiseless and noisy case are considered in the tests. Let $\hat{\boldsymbol{D}}$ and $\hat{\boldsymbol{X}}$ be the learned dictionary and the corresponding sparse coefficients, respectively. The normalized learning error is defined as $\left\|\boldsymbol{Y} - \hat{\boldsymbol{D}}\hat{\boldsymbol{X}}\right\|_F^2/n$. The criteria for success learning are designed for both cases using the normalized learning error: in the noiseless case, a success is claimed when $\left\|\boldsymbol{Y} - \hat{\boldsymbol{D}}\hat{\boldsymbol{X}}\right\|_F^2/n \leq \epsilon_e \left\|\boldsymbol{Y}\right\|_F^2$ where the constant ϵ_e is ideally zero but set to 10^{-6} in practice; for the noisy case, the criterion for a successful learning is given by $\left\|\boldsymbol{Y} - \hat{\boldsymbol{D}}\hat{\boldsymbol{X}}\right\|_F^2/n \leq \epsilon_n \left\|\boldsymbol{Y}\right\|_F^2$ where $\epsilon_n := \left\|\boldsymbol{W}\right\|_F^2/n/\left\|\boldsymbol{D}_{true}\boldsymbol{X}_{true}\right\|_F^2$.

In the tests, four algorithms, namely MOD, K-SVD, regularized SimCO, and smoothed SimCO, are compared. For each of these algorithms, the maximum number of iterations is set to 1000. For regularized SimCO, the regularization constant is initially set as $\mu = 0.1$ and then reduced to $\mu/10$ after every 100 iterations. In smoothed SimCO, the thresholds $\boldsymbol{\delta}_i$'s are set to $(0.001, 0.2)$



(a) Noiseless case: normalized distortion.



(b) Noisy case: normalized distortion.

Figure 3.5.1: The performance comparison.

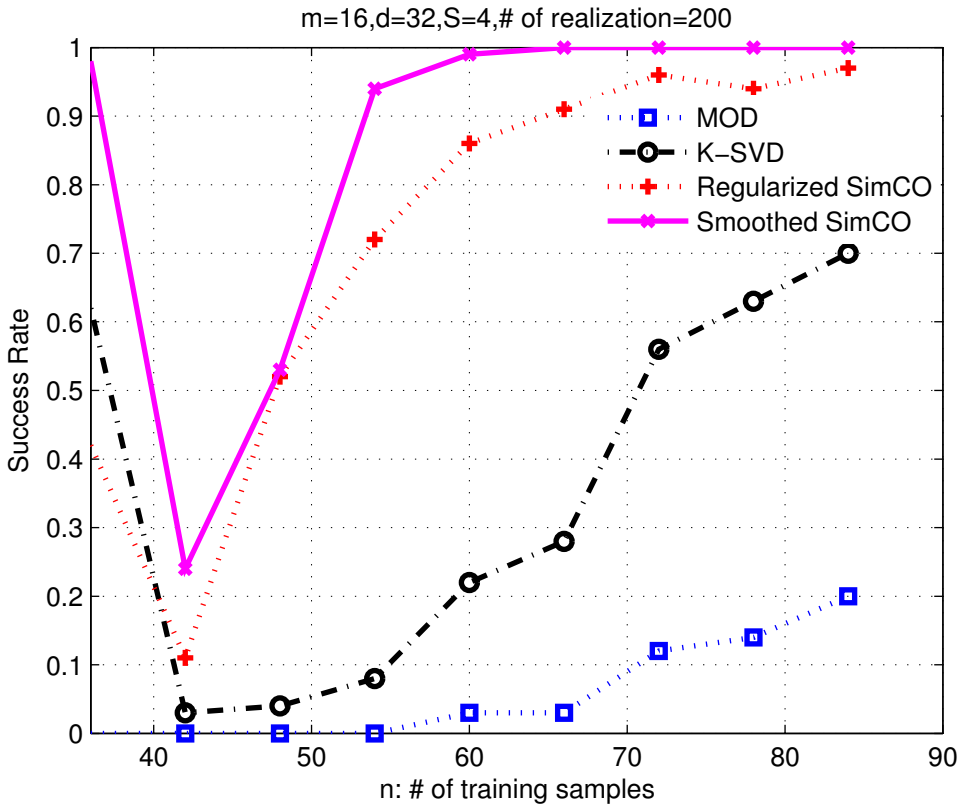


Figure 3.5.2: Noiseless case: success rate.

for the first 500 iterations and then to $(0, 0)$ for the rest 500 iterations. (Note that $\boldsymbol{\delta}_i = \boldsymbol{\delta}_j$ due to the simulation setting.)

The simulation results are presented in Fig. 3.5.1-3.5.2, where the first two sub-figures compare the normalized distortion and the last one focuses on the success rate. The advantage of the proposed smoothed SimCO is clear for both noiseless and noisy cases. In terms of success rate, smoothed SimCO reaches 100% success rate when the number of training samples $n > 60$ while MOD and K-SVD could not achieve 100% success rate even when $n \geq 84$. It is also interesting to observe the dip in the success rate when n is in the middle-range (Fig. 3.5.2). This is expected. On one hand, the success rate should increase when the number of training samples becomes larger. On the other hand,

when the number of training samples is extremely low, for example, $n = 1$, the learning problem becomes trivial. Hence, the most difficult case is when n is in the middle-range.

3.5.2 Algorithm Testing on BSS Problem

In this Section we present numerical results of the SparseBSS method compared with some other mainstream algorithms. We first focus on speech separation where an equal determined case will be considered. Then we show an example for blind image separation, where we will consider an overdetermined case. As final remarks, smoothed SimCO has several theoretic advantages over regularized SimCO. However, the computations of $(\lambda_{\min}(\boldsymbol{D}_i))$'s introduce extra cost. The choice between these two methods will depend on the size of the problem under consideration. Here, we use the regularized SimCO for SparseBSS.

In the speech separation case, two mixtures are used which are the mixtures of two audio sources. Two male utterances in different languages are selected as the sources. The sources are mixed by a 2×2 random matrix $\boldsymbol{A}$ (with normalized columns). For the noisy case, a 20 dB Gaussian noise was added to the mixtures. See Fig. 3.5.3 for the sources and mixtures.

We compare SparseBSS with two benchmark algorithms including FastICA and QJADE [18]. The BSSEVAL toolbox [75] is used for the performance measurement. In particular, an estimated source $\hat{\boldsymbol{s}}$ is decomposed as $\hat{\boldsymbol{s}} = \boldsymbol{s}_{\text{target}} + \boldsymbol{e}_{\text{interf}} + \boldsymbol{e}_{\text{noise}} + \boldsymbol{e}_{\text{artif}}$, where $\boldsymbol{s}_{\text{target}}$ is the true source signal, $\boldsymbol{e}_{\text{interf}}$ denotes the interferences from other sources, $\boldsymbol{e}_{\text{noise}}$ represents the deformation caused by the noise, and $\boldsymbol{e}_{\text{artif}}$ includes all other artifacts introduced by the separation algorithm. Based on the decomposition, three performance criteria can be

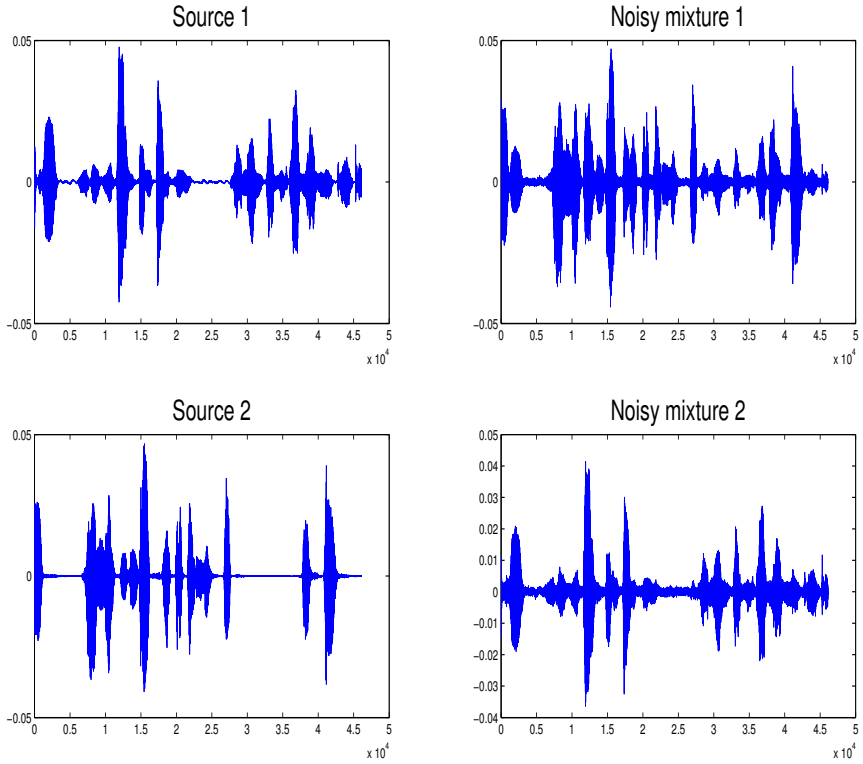


Figure 3.5.3: Two speech sources and the corresponding noisy mixtures (20 dB Gaussian noise).

defined: the source-to-distortion ratio SDR $= 10\log_{10}\frac{\|\boldsymbol{s}_{\text{target}}\|^2}{\|\boldsymbol{e}_{\text{interf}}+\boldsymbol{e}_{\text{noise}}+\boldsymbol{e}_{\text{artif}}\|^2}$, the source-to-artifact ratio SAR $= 10\log_{10}\frac{\|\boldsymbol{s}_{\text{target}}+\boldsymbol{e}_{\text{interf}}+\boldsymbol{e}_{\text{noise}}\|^2}{\|\boldsymbol{e}_{\text{artif}}\|^2}$, and the source-to-interference ratio SIR $= 10\log_{10}\frac{\|\boldsymbol{s}_{\text{target}}\|^2}{\|\boldsymbol{e}_{\text{interf}}\|^2}$. Among them, the SDR measures the overall performance (quality) of the algorithm, and the SIR focuses on the interference rejection. We investigate the gains of SDRs, SARs and SIRs from the mixtures to the estimated sources. For example, $\triangle SDR = SDR_{out} - SDR_{in}$, where SDR_{out} is calculated from its definition and SDR_{in} is obtained by letting $\hat{\boldsymbol{s}} = \boldsymbol{Z}$ with the same equation. The results (in dB) are summarized in Table 3.1.

	△SDR	△SIR	△SAR
QJADE	60.661	60.661	-1.560
FastICA	57.318	57.318	-0.272
SparseBSS	**69.835**	**69.835**	**1.379**

(a) The noiseless case.

	△SDR	△SIR	△SAR
QJADE	7.453	58.324	-1.245
FastICA	7.138	40.789	-1.552
SparseBSS	**9.039**	**62.450**	**0.341**

(b) The noisy case.

Table 3.1: Separation performance of the SparseBSS algorithm as compared to FastICA and QJADE. The proposed SparseBSS algorithm performs better than the benchmark algorithms. Table 3.1a. For the same algorithm, the $\triangle SDR$ and $\triangle SIR$ are the same in noiseless case. The $\triangle SDRs$ and $\triangle SIRs$ for all the tested algorithms are large and similar, suggesting that all the compared algorithms perform very well. The artifact introduced by SparseBSS is small as its $\triangle SAR$ is positive. Table 3.1b. In the presence of noise with SNR = 20 dB, SparseBSS excels the other algorithms in $\triangle SDR$, $\triangle SIR$ and $\triangle SAR$. One interesting phenomenon is that the $\triangle SDRs$ are much smaller than those in the noiseless case, implying that the distortion introduced by the noise is trivial. However, SparseBSS still has better performance.

The selection of λ is an important practical issue since it is related to the noise level and largely affects the algorithm performance. From the optimization formulation (3.3.3), it is clear that with a fixed SNR, different choices of λ may give different separation performance. To show this, we use the estimation error $\left\|\boldsymbol{A}_{\text{true}} - \hat{\boldsymbol{A}}\right\|_F^2$ of the mixing matrix to measure the separation performance, where $\boldsymbol{A}_{\text{true}}$ and $\hat{\boldsymbol{A}}$ are the true and estimated mixing matrices, respectively. The simulation results are presented in Fig. 3.5.4. Consistent with the intuition, simulations suggest that the smaller the noise level the larger the optimal value of λ. The results in Fig. 3.5.4 help in setting λ when the noise level is known a priori.

Next, we show an example for blind image separation, where we consider an overdetermined case. The mixed images are generated from two source images using a 4×2 full rank column normalized mixing matrix $\boldsymbol{A}$ with its elements generated randomly according to a Gaussian process. The mean squared errors (MSEs) are used to compare the reconstruction performance of the candidate algorithms when no noise is added. MSE is defined as $MSE = (1/N)\left\|\chi - \tilde{\chi}\right\|_F^2$, where χ is the source image and $\tilde{\chi}$ is the reconstructed image. The lower the MSE, the better the reconstruction performance. Table 3.2 illustrates the results of four tested algorithms. For the noisy case, a Gaussian white noise was added to the four mixtures with $\sigma = 10$. We use the Peak Signal-to-Noise Ratio (PSNR) to measure the reconstruction quality, which is defined as, $PSNR = 20\log_{10}(\frac{MAX}{\sqrt{MSE}})$, where MAX indicates the maximum possible pixel value of the image, (e.g., $MAX = 255$ for a uint-8 image). Higher PSNR indicates better quality. The noisy observations are illustrated in Fig. 3.5.5(b). For the BMMCA test, a better performance was demonstrated in [3].

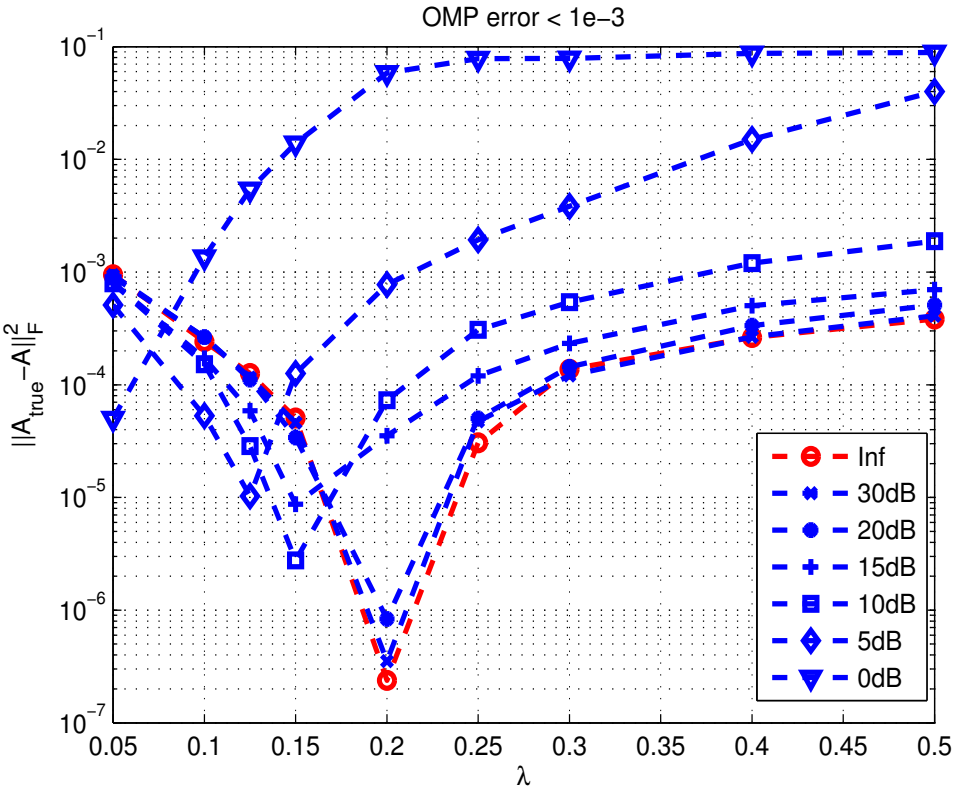


Figure 3.5.4: Relation of the parameter λ to the estimation error of the mixing matrix under different noise levels. The signal-to-noise ratio (SNR) is defined as $\rho = 10 \log_{10} \|\boldsymbol{AS}\|_F^2 / \|\boldsymbol{V}\|_F^2$ dB.

We point out that here a different true mixing matrix is used. And further more, in our tests the patches are taken with a 50% overlap (by shifting 4 pixels from the current patch to the next) while in [3] the patches are taken by shifting only one pixel from the current patch to the next.

	FastICA	GMCA	BMMCA	SparseBSS
Lena	8.7489	4.3780	3.2631	**3.1346**
Boat	18.9269	**6.3662**	12.5973	6.6555

Table 3.2: Achieved MSEs of the algorithms in a noiseless case.

Figure 3.5.5: Two classic images, *Lena* and *Boat* were selected as the source images, which are shown in (a). The mixtures are shown in (b). The separation results are shown in (c)-(f). We compared SparseBSS with other benchmark algorithms: FastICA [42], GMCA [10] and BMMCA [3]. We set the overlap percentage equal to 50% for both BMMCA and SparseBSS. The recovered source images by the SparseBSS tend to be less blurred as compared to the other three algorithms.

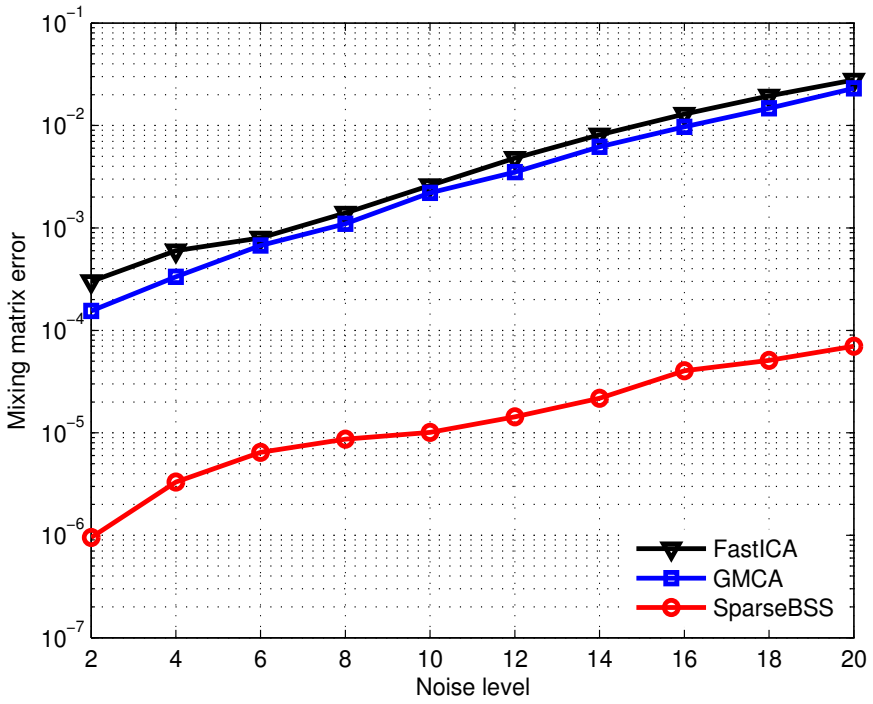


Figure 3.5.6: Compare the performance of estimating the mixing matrix for all the methods in different noise standard deviation σs. In this experiment, σ varies from 2 to 20. The performance of GMCA is better than that of FastICA. The curve for BMMCA is not available as the setting for the parameters is too sophisticated and inconsistent for different σ to obtain a good result. SparseBSS outperforms the compared algorithms.

At last, we show another example of blind image separation to demonstrate the importance of the singularity aware process. In this example, we use two classic images *Lena* and *Texture* as the source images (Fig. 3.5.7(a)). Four noiseless mixtures were generated from the sources. The separation results are shown in Fig. 3.5.7(b) and (c). Noting that images like *Texture* contain a lot of frequency components corresponding to a particular frequency. Hence an initial dictionary with more codewords corresponding to the particular frequency may perform better for the estimation of these images. Motivated by

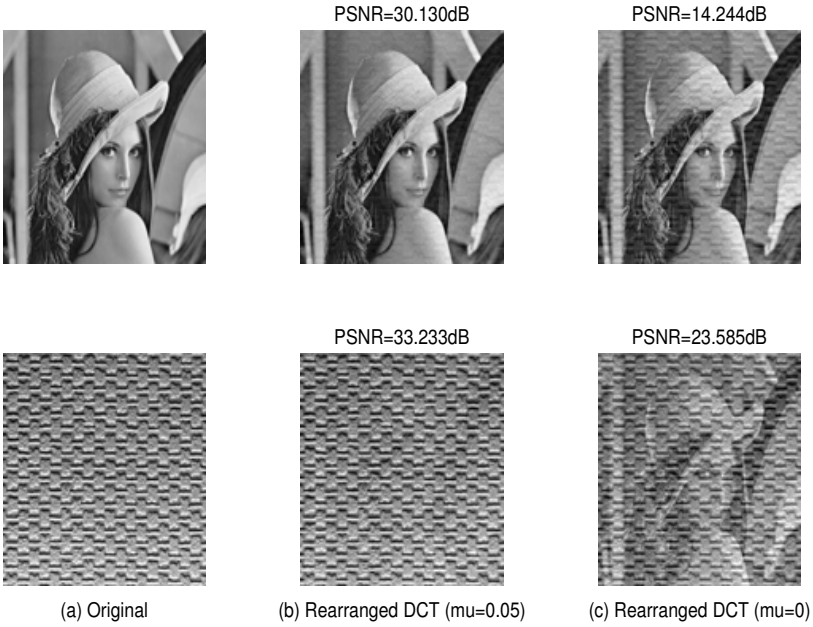


Figure 3.5.7: The two source images *Lena* and *Texture* are shown in (a). The separation results are shown in (b) and (c). The comparison results demonstrate the importance of the singularity aware process.

this, in Fig. 3.5.7(b) the initial dictionary is generated from an over-complete DCT dictionary but contains more high frequency codewords. Such choice can lead to better separation results. At the same time, the very similar dictionary codewords may introduce the risk of singularity issue.

The major difference between Fig. 3.5.7(b) and (c) is that: in Fig. 3.5.7(b) the regularized SimCO process ($\mu = 0.05$) is introduced, while in Fig. 3.5.7(c) there is no regularization term in the dictionary learning stage. As one can see from Fig. 3.5.7(b) performs much better than Fig. 3.5.7(c). By checking the condition number when the regularized term is not introduced ($\mu = 0$), the value stays in a high level as expected (larger than 40 in this example). This confirms the necessity of considering the singularity issue in BSS and the effectiveness of the proposed singularity aware approach.

3.6 Conclusions

In conclusion, we introduced a development of the blind source separation algorithms based on dictionary learning. In particular, we focus on the SparseBSS algorithm and the optimization procedures. We compared SparseBSS in details with the related benchmark algorithm BMMCA. We also discussed the important observation of the singularity issue, which is a major reason for the failure of dictionary learning algorithms and hence dictionary learning based BSS algorithms. Afterwards, two available approaches are presented to address this problem. We designed *smoothed SimCO* adapted to the smooth technique in Chapter 2. It has comparable results than the regularized SimCO.

Chapter 4

Robust Face Recognition

4.1 Introduction

The aim of robust face recognition problem is to recognize a test face image that may be corrupted by arbitrary noise [81, 77]. It has been demonstrated that sparse signal processing can solve this problem with impressive performance. Mathematically, a vector $\boldsymbol{x}$ is sparse if only a small fraction of components in $\boldsymbol{x}$ are significant while the majority of the components are zero or close to zero. Sparse recovery problem is to solve the linear inverse problem

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}, \tag{4.1.1}$$

where the observation $\boldsymbol{y} \in \mathbb{R}^m$ and the mixing matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ are given, the unknown signal $\boldsymbol{x} \in \mathbb{R}^n$ is assumed to be sparse, and the noise $\boldsymbol{w} \in \mathbb{R}^n$ is often white Gaussian. In the robust face recognition setting [81, 77], the vector $\boldsymbol{y}$ is the test face image, the matrix $\boldsymbol{A}$ is derived from training samples, the sparse vector $\boldsymbol{x}$ contains feature coefficients and $\boldsymbol{w}$ is noise (assumed to be

relatively sparse).

There are many algorithms to solve the sparse recovery problem. They can be divided into two categories: greedy algorithms and ℓ_1-minimization approaches. Greedy algorithms, such as OMP [60], SP [20], are fast while may not work well in some cases. ℓ_1-minimization is an efficient alternative approach to the sparse recovery problem. It has been a hugely successful approach in the past decade. Despite those existing methods, in this Chapter, we are particularly interested AMP algorithms [26, 27, 64], which are based on loopy belief propagation. Those alternative algorithms deliver both low computational cost and performance guarantees while $\boldsymbol{A}$ has i.i.d. Gaussian entries of zero mean. Unfortunately, the transform matrix $\boldsymbol{A}$ in the face recognition problem is not Gaussian. Experiments in[81] gave pessimistic results.

Recently, several variants of Generalized AMP (GAMP) algorithm [64] have been proposed to handle non-Gaussian mixing matrices. The ADMM-GAMP algorithm [65] has provable convergence guarantees with arbitrary measurement matrix. It requires solving an additional least squares problem in each iteration. That makes the ADMM-GAMP algorithm lacks of computational efficiency. Swept AMP (SwAMP)[55] offers more robust results as it requires a sequential updating procedure rather in parallel. Also, it is not a fast approach compared with other variants in the literature. Vila et. al. [74] proposed an adaptive version of Damped-GAMP [66]. This so called AD-GAMP method adaptively updates the damping coefficient, which is determined by the peak-to-average ratio of the squared singular values in $\boldsymbol{A}$. In AD-GAMP, it partially updates the variables tuned by the damping coefficient. In this Chapter, we study the AD-GAMP algorithm for the robust face recognition problem.

Those GAMP based methods assume known prior information about the

signal. For example, the sparse signal is Bernoulli-Gaussian, the measurement noise is additive Gaussian, etc. However, the hyper-parameters in those probability distributions are often not known a prior in practice. Gaussian Mixture-GAMP (GM-GAMP) [73] use EM method to estimate the hyper-parameters. It assumes the sparse signal is Gaussian mixture distributed and the noise is AWGN. Bernoulli-Gaussian GAMP (BG-GAMP) [72] assumes a BG distributed sparse signal, which is a special case of GM in GM-GAMP [73].

The main contribution [1] of this Chapter include:

- Successfully solve the robust face recognition problem using the AMP framework. AD-GAMP is adapted to address the issue that the mixing matrix $\boldsymbol{A}$ in face recognition is far from the standard Gaussian random matrix.

- Motivated by the nature of Wright et al.'s framework [77], we model the unknown signal $\boldsymbol{x}$ using a statistical model involving Bernoulli-Gaussian priors. The major difference between our model and the benchmark [77] is that in this work the sparse signal is divided into two segments — one corresponds to the feature coefficients and the other is linked to anomalies to achieve robustness — and different segments have different hyper-parameters. Hence, it terms Dual BG-GAMP in this Chapter.

- Then a Dual EM method is employed to estimate the unknown hyper-parameters associated with the two segments. With the EM and AD-GAMP coupled together, our method achieves better recognition performance than the ℓ_1-minimization benchmarks in the review paper [81],

[1] This work was supported in part by Defence Science and Technology Laboratory (Dstl) under Grant No: DSTLX-1000081291.

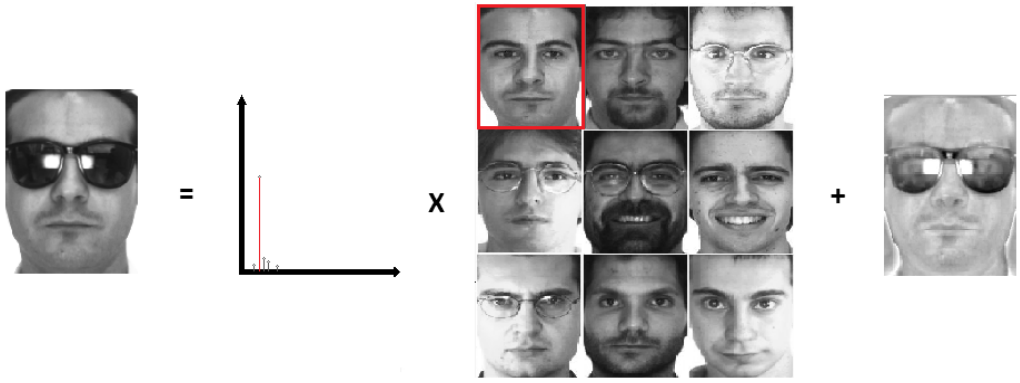


Figure 4.2.1: Overview of the SRC framework. The test image (left), which is occluded by a sunglasses. It is equal to the sparse linear combinations of the training images (middle) plus error image (right). The sparse coefficient (red) indicate the corresponding true identity, which is bounded in a red box in the training images (middle). This graph is only for demonstration. There are hundreds or even thousands of training images in the test.

much better than the pessimistic results of the original AMP [81]. Simulation results also demonstrate that the algorithm is quite robust to the initial values of hyper-parameters, and exhibits low computational cost thanks to the efficiency of the AMP framework.

The remainder of this Chapter is organized as follows. Section 4.2 introduces the robust face recognition problem and AMP algorithm. Section 4.3 is devoted to describing the proposed Dual update method based on AMP for robust face recognition. The empirical performance improvement is demonstrated in Section 4.4. This work is concluded in the last Section.

4.2 Preliminary Research

4.2.1 Robust Face Recognition

Unlike traditional dictionary learning approaches, the authors of [77] let the training samples be the dictionary in the *sparse representation based classification* (SRC) framework. Each testing image is assumed to be a sparse linear combination of the training set. The mathematical model is as follows, $\boldsymbol{y}_0 = [\boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A}_s][\boldsymbol{x}_{1,0}^T, \boldsymbol{x}_{2,0}^T, ..., \boldsymbol{x}_{s,0}^T]^T = \boldsymbol{A}\boldsymbol{x}_0$, where $\boldsymbol{y} \in \mathbb{R}^m$ is the vectorized test image, the sub-matrix $\boldsymbol{A}_i \in \mathbb{R}^{m\times l}$ and each block $\boldsymbol{x}_{i,0} \in \mathbb{R}^l$ for $i \in [1, ..., s]$. Each column of $\boldsymbol{A}$ is a vectorized training image. Here, the $\boldsymbol{A}_i$ contains l different images all for the ith identity. For simplicity, let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ here. An overview of this framework is shown in Fig. 4.2.1. In this case, the columns of transform matrix $\boldsymbol{A}$ are correlated, hence the AMP algorithms do not have convergence guarantees.

In [77, 76], the authors consider two fundamental issues in face recognition problem. Firstly, the role of feature extraction. In other words, one is aiming to project high dimensional testing data into low dimensional feature spaces, which is still informative for sparse representation. Secondly, the occlusion is an obstacle to recognition. In practice, a fraction of test images is often corrupted. In [77, 76], the robust SRC model is,

$$\boldsymbol{y} = \boldsymbol{y}_0 + \boldsymbol{e}_0 = [\boldsymbol{A},\ \boldsymbol{I}] \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{e}_0 \end{bmatrix}, \tag{4.2.1}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is a down sampled vectorized image with sparse occlusion $\boldsymbol{e}_0$, and $\boldsymbol{I} \in \mathbb{R}^{m\times m}$ is an identity matrix. Eq. (4.2.1) can then be simplified as,

$$\boldsymbol{y} = \boldsymbol{\Phi} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \end{bmatrix}, = \boldsymbol{\Phi}\boldsymbol{x}, \tag{4.2.2}$$

where $\boldsymbol{\Phi} = [\boldsymbol{A},\ \boldsymbol{I}] \in \mathbb{R}^{m\times(n+m)}$ and the lower part of the sparse coefficient $\boldsymbol{x}_1 = \boldsymbol{e}_0 \in \mathbb{R}^m$. Then, it considers the following ℓ_1 problem,

$$\hat{\boldsymbol{x}} = \arg\min \|\boldsymbol{x}\|_1 \ \ s.t.\ \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}.$$

After one get the estimated sparse coefficient $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{x}}_0^T, \hat{\boldsymbol{x}}_1^T]^T$, we can assign the tested object to i^* by applying the Sparsity Concentration Index (SCI) in [77],

$$i^* = \arg\max_i = \frac{s \cdot \|\delta_i(\hat{\boldsymbol{x}}_0)\|_1 \,/\, \|\hat{\boldsymbol{x}}_0\|_1 - 1}{s-1},$$

where $\delta_i(\cdot)$ is an operator that keeps the i-th block of $(\cdot)$. In other words, the SCI finds best block of $\hat{\boldsymbol{x}}_0$, which has the most number of non zero elements concentrated in $\delta_i(\hat{\boldsymbol{x}}_0)$ out of s blocks.

4.2.2 AMP/GAMP for SRC

The AMP is a powerful tool to solve the ℓ_1 problem since it exhibits both low reconstruction error and low computational complexity compared with benchmarks. However, this mechanism only achieves the desired asymptotical optimal performance when the linear transform is standard Gaussian. The GAMP accommodates more general signal models. Here, we consider GAMP for simplicity. GAMP is also flexible to couple with the EM approach to learn the unknown hyper-parameters such as sparsity values, which is more applicable to real-time applications.

In robust face recognition problem, the measurement matrix $\boldsymbol{\Phi}$ violates

two assumptions which are critical for AMP approaches. The first one is the non-zero mean assumption of the measurement matrix in practice. It has been shown in [13] that even with a small positive mean of the i.i.d measurement matrix, the algorithm may diverge. There are three ways of solving this problem. First, remove the mean of the matrix in pre-processing, which is common in image processing fields. Second, modify the update procedure from parallel to sequential, since the parallel update is more problematic [55][13]. Third, modify the mathematical model/measurement matrix to remove the mean, as in [74]. In our case, we remove all the means of the training and testing images. The non-zero mean value of the measurement matrix $\mathbf{\Phi}$ is dominated by the identity matrix. In this case, the mean value is roughly $\frac{m}{m(n+m)} = \frac{1}{n+m}$. If the number of training images is fixed to n, a larger sampling size m leads smaller mean value. If n or m is large enough, the mean value of the measurement matrix Φ is close to zero. Then, the non-zero mean issue will not affect the convergence of the algorithm. The second assumption of AMP is that the matrix $\mathbf{\Phi}$ is i.i.d, which is impractical for robust face recognition problem, i.e., the columns are correlated. A review of fast ℓ_1-minimization algorithms has been studied in [81]. The authors of [81] also added AMP in comparison. In their i.i.d Gaussian experiments, AMP is shown to be the fastest algorithm with near-machine precision. Not surprisingly, AMP fails as it is not capable of handling the general measurement matrix $\mathbf{\Phi}$ [81]. We shall address the correlation issue of the measurement matrix using the damping approach AD-GAMP [74] and learn the unknown hyper-parameters using the EM embedded BG-GAMP algorithm [72]. However, simply combined algorithm can not achieve better recognition rate than benchmark algorithms. Adapting to the structure of the sparse signal in SRC framework, we designed a new dual

updating approach based on the combination of those two algorithms. More details of our method are presented in next Section.

4.3 An AMP Based Method

4.3.1 Dual BG-GAMP

In this Chapter, we consider the two segments of the sparse signal in the robust SRC model (4.2.2) to have different hyper-parameters. Furthermore, we assume the two segments sparse coefficients $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are both Bernoulli-Gaussian distributed. Hence, it terms dual BG-GAMP here. One can then apply the EM embedded BG-GAMP algorithm [72] to learn the unknown hyper-parameters that associated with the sparse signal, e.g., the sparsities, mean values and the variances.

In our approach, we consider the upper part $\boldsymbol{x}_0$ and the lower part $\boldsymbol{x}_1$ of the sparse coefficient $\boldsymbol{x}$ that are ideally not identically distributed. In other words, we consider each of them has different hyper-parameters, i.e., sparsity levels ϵs, mean values θs and variances ϕs. Then, for the signal $\boldsymbol{x} = [\boldsymbol{x}_0^T, \boldsymbol{x}_1^T]^T \in \mathbb{R}^{(n+m)}$, which is assumed to be drawn i.i.d from the pdf

$$P_X(x_{jk}; \epsilon_k, \theta_k, \phi_k) = (1-\epsilon_k)\delta(x_{jk}) + \epsilon_k \mathcal{N}(x_{jk}; \theta_k, \phi_k), \tag{4.3.1}$$

where $\delta(\cdot)$ denotes the Dirac function, $k \in \{0, 1\}$, $j = 1, 2, ...,$ (n+m), and $\mathcal{N}(\cdot; \theta, \phi)$ is the Gaussian pdf. In this Chapter, we introduce parameter k that indicates which the element x_{jk} belongs to, either $\boldsymbol{x}_0$ or $\boldsymbol{x}_1$. In particular, if $k = 0$, then $j = 1, ..., n$, otherwise $j = (n+1), ..., (n+m)$. For the AWGN noise $\boldsymbol{w}$ is assumed to be independent of $\boldsymbol{x}$ with variance ψ, $P_W(w; \psi) = \mathcal{N}(w; 0, \psi)$.

In this case, we define the unknown hyper-parameters of the prior distribution as $\boldsymbol{q}_k \triangleq [\epsilon_k, \theta_k, \phi_k, \psi]$. It is noteworthy to mention if we drop the subscription k in Eq. (4.3.1), it becomes the standard BG in [72].

In the GAMP, one is aiming to estimate the input $\boldsymbol{x}$ and the noiseless output $\boldsymbol{z} = \boldsymbol{\Phi x}$ of the transform. The probabilistic relationships in the input and output models are defined in table (4.1) [72]. The standard BG input scalar estimation function g_{in} and the AWGN output scalar estimation function g_{out} are already given in [72] and [64], respectively.

4.3.2 Adaptive Damping

AMP/GAMP approach does not work well while the matrix $\boldsymbol{A}$ is general [81], e.g., column correlated in robust SRC model. Among the various ways of addressing this issue, we are interested in the AD-GAMP [74] approach. In Damped-GAMP, Rangoon et al. [66] introduced a damping parameter β to adjust the updates of adjacent iterations so that it converges under general transform. It is shown that the damping parameter is proportional to the peak-to-average ratio of the squared singular value of the transform matrix. If this ratio is sufficiently small, the GAMP converges. This scenario explains why AMP performs well when the transaction is large i.i.d Gaussian. The damping approach guarantees the convergence when the transaction is general while it slows down the progress of convergence. In [74], the authors proposed an adaptive damping GAMP scheme to find a good damping parameter to prevent slowing down the updates procedure too much.

In this Chapter, we consider the combination of the Dual BG-GAMP and the AD-GAMP approaches, which is shown in Algorithm 4.1. Here we assume the dual hyper-parameters satisfies Eq. (4.3.1), where it is different from the

$$
\begin{aligned}
P_{Z|Y}(z \mid y; \hat{z}, \mu^z) &= \frac{P_{Y|Z}(y \mid z)\mathcal{N}(z; \hat{z}, \mu^z)}{\int_{z'} P_{Y|Z}(y \mid z')\mathcal{N}(z'; \hat{z}, \mu^z)} \\
g_{out}(y, \hat{z}, \mu^z) &= \frac{y - \hat{z}}{\mu^z + \psi} \\
g'_{out}(y, \hat{z}, \mu^z) &= -\frac{1}{\mu^z + \psi} \\
P_{X|Y}(x \mid y; \hat{r}, \mu^r) &= \frac{P_X(x)\mathcal{N}(x; \hat{r}, \mu^r)}{\int_{x'} P_X(x')\mathcal{N}(x'; \hat{r}, \mu^r)} \\
\pi(\hat{r}, \mu^r; \boldsymbol{q}_k) &= \frac{1}{1 + (\frac{\epsilon_k}{1-\epsilon_k}\frac{\mathcal{N}(\hat{r}; \theta, \phi+\mu^r)}{\mathcal{N}(\hat{r}; 0, \phi+\mu^r)})^{-1}} \\
\gamma(\hat{r}, \mu^r; \boldsymbol{q}_k) &= \frac{\hat{r}/\mu^r + \theta/\phi}{1/\mu^r + 1/\phi} \\
\nu(\hat{r}, \mu^r; \boldsymbol{q}_k) &= \frac{1}{1/\mu^r + 1/\phi} \\
g_{in}(\hat{r}, \mu^r; \boldsymbol{q}_k) &= \pi(\hat{r}, \mu^r; \boldsymbol{q}_k)\gamma(\hat{r}, \mu^r; \boldsymbol{q}_k) \\
\mu^r g'_{in}(\hat{r}, \mu^r; \boldsymbol{q}_k) &= \pi(\hat{r}, \mu^r; \boldsymbol{q}_k)(\nu(\hat{r}, \mu^r; \boldsymbol{q}_k) \\
&\quad + |\gamma(\hat{r}, \mu^r; \boldsymbol{q}_k)|^2) \\
&\quad -(\pi(\hat{r}, \mu^r; \boldsymbol{q}_k))^2 |\gamma(\hat{r}, \mu^r; \boldsymbol{q}_k)|^2
\end{aligned}
$$

Table 4.1: Definitions: the input and output probabilistic relationships $P_{X|Y}$ and $P_{Z|Y}$. The associated BG input scalar estimation function and the AWGN output scalar estimation function [72][64].

original BG in [72]. In the GAMP, one is aiming to estimate the input $\boldsymbol{x}$ and the noiseless output $\boldsymbol{z} = \boldsymbol{\Phi x}$ of the transform. The probabilistic relationships in the input and output models are defined in [72]. The standard BG input scalar estimation function g_{in} and the AWGN output scalar estimation function g_{out} are already given in [72] and [64], respectively. As one can find in Eq. (6-7), (9-11), $\beta(t)$ is the adaptive damping parameter. At the very end of this algorithm, the damping parameter is tuned according to current estimation $\hat{\boldsymbol{x}}(t+1)$ and the MMSE cost $J(t+1)$, adaptively. Here, $J(t+1) = J_{Bethe}(t+1)$, which is the Bethe Free Energy function. Here, we refer [65][74] for more details about AD-GAMP. It is straightforward to obtain BG-GAMP algorithm by letting $\beta(t) = 1$ and ignore the adapting step in Algorithm 4.1.

4.3.3 Dual Expectation Maximization

We use EM algorithm to estimate the hyper-parameters in Algorithm (4.1). The EM [24][57] is a well-established method for maximum likelihood estimation with hidden variables. An explicit EM algorithm has been given in [72] for BG-GAMP. It updates hyper-parameters sequentially where updating one parameter by fixing all the other parameters simultaneously. The designed algorithm calls Algorithm 4.1 after each Dual EM update step. In other words, we upgrade the parameters in the outer algorithm (Dual EM) and perform the Dual BG-AD-GMAP using the new parameters in the inner algorithm. In our case, the EM update is,

$$\forall k : \boldsymbol{q}_k^{h+1} = \arg\max_{\boldsymbol{q}} E\{\ln P(\boldsymbol{x}_k, \boldsymbol{w}; \boldsymbol{q}_k) \mid \boldsymbol{y}; \boldsymbol{q}_k^h\},$$

Algorithm 4.1 Inner algorithm (Dual BG-AD-GAMP) with AWGN output.

Initialization:

$$\begin{aligned}
\forall j : \hat{x}_{jk}(1) &= \int_{\boldsymbol{x}_k} x_{jk} P_X(x_{jk}) \\
\forall j : \mu^x_{jk}(1) &= \int_{\boldsymbol{x}_k} |x_{jk} - \hat{x}_{jk}(1)|^2 P_X(x_{jk}) \\
\forall i : \hat{u}_i(0) &= 0
\end{aligned}$$

$\beta(1) = 1$, $\beta_{max} \in (0, 1]$, $\beta_{min} \in (0, \beta_{max}]$, $G_{pass} \geq 1$, $G_{fail} < 1$, $\varepsilon > 0$
for t = 1, 2, 3,...

$$\begin{aligned}
\forall i : \hat{z}_i(t) &= \sum_{j=1}^{(n+m)} \boldsymbol{\Phi}_{ij} \hat{x}_{jk}(t) && (4.3.2) \\
\forall j : \tilde{\boldsymbol{x}}_{jk}(t) &= \beta(t)\hat{x}_{jk}(t) + (1 - \beta(t))\tilde{x}_{jk}(t-1) && (4.3.3) \\
\forall i : \mu^z_i(t) &= \beta(t) \sum_{j=1}^{(n+m)} |\boldsymbol{\Phi}_{ij}|^2 \mu^x_{jk}(t) && (4.3.4) \\
&\quad + (1 - \beta(t))\mu^z_i(t-1) \\
\forall i : \hat{p}_i(t) &= \hat{z}_i(t) - \mu^z_i(t)\hat{u}_i(t-1) && (4.3.5) \\
\forall i : \hat{u}_i(t) &= \beta(t) g_{out}(y_i, \hat{p}_i(t), \mu^z_i(t)) && (4.3.6) \\
&\quad + (1 - \beta(t))\hat{u}_i(t-1) \\
\forall i : \mu^u_i(t) &= \beta(t)(-g'_{out}(y_i, \hat{p}_i(t), \mu^z_i(t))) && (4.3.7) \\
&\quad + (1 - \beta(t))\mu^u_i(t-1) \\
\forall j : \mu^r_{jk}(t) &= \beta(t)(\sum_{j=1}^{(n+m)} |\boldsymbol{\Phi}_{ij}|^2 \mu^u_i(t))^{-1} && (4.3.8) \\
&\quad + (1 - \beta(t))\mu^r_{jk}(t-1) \\
\forall j : \hat{r}_{jk}(t) &= \tilde{x}_{jk}(t) + \mu^r_{jk}(t) \sum_{i=1}^{m} \boldsymbol{\Phi}^*_{ij} \hat{u}_i(t) && (4.3.9) \\
\forall j : \mu^x_{jk}(t+1) &= \mu^r_{jk}(t) g'_{in}(\hat{r}_{jk}(t), \mu^r_{jk}(t)) && (4.3.10) \\
\forall j : \hat{x}_{jk}(t+1) &= g_{in}(\hat{r}_{jk}(t), \mu^r_{jk}(t)) && (4.3.11) \\
J(t+1) &= J_{Bethe}(t+1) && (4.3.12)
\end{aligned}$$

if $J(t+1) \leq \max\{J(\triangle t), ..., J(t)\}$ or $\beta(t) = \beta_{min}$
then if $\|\hat{\boldsymbol{x}}(t) - \hat{\boldsymbol{x}}(t+1)\| / \|\hat{\boldsymbol{x}}(t+1)\| < \varepsilon$
then stop
else $\beta(t+1) = \min\{\beta_{max}, G_{pass}\beta(t)\}$
$t = t + 1$
else $\beta(t) = \min\{\beta_{min}, G_{fail}\beta(t)\}$
end
end

where h denotes the iteration index. It it worth to note that one has to calculate the corresponding hyper-parameters of each $\boldsymbol{x}_k$, separately. Following [72], it is easy to obtain the updates for each hyper-parameters, which are shown in Algorithm 4.2. We therefore only show the differences, since we have to update both $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ in this case.

In the Dual EM step, one has to consider the values of the hyper-parameters $\boldsymbol{q}_k$ for different k. In this Chapter, we proposed to update the $\boldsymbol{q}_k$ according to the structure of the sparse signal, as one can see from the Eq. 4.2.1. In our approach, $\boldsymbol{q}_1 \triangleq [\epsilon_1, \theta_1, \phi_1, \psi]$ is the hyper-parameters that associated with the feature coefficient $\boldsymbol{x}_0$, which is linear combination coefficients of the training images. For $\boldsymbol{q}_2 \triangleq [\epsilon_2, \theta_2, \phi_2, \psi]$, it is determined by the down sampling methods and the noise of the test images $\boldsymbol{x}_1$. It is natural to guess that $\boldsymbol{q}_1 \neq \boldsymbol{q}_2$ since $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ associate with the sparse feature and sparse occlusion, respectively. In order to compare the performances of the BG-AD-GMAP based algorithms, we set the all the initial value of the sparse vectors to be the same. We present the comparison in next Section.

4.4 Experiments

In this Section, we present two experiments to compare the performances of our method with the benchmark algorithms in the comprehensive review paper [81][2]. The first experiment is designed to compare the recognition rate of all tested algorithms. The second experiment corresponds to the comparison of the computational costs in order to achieve the best recognition rates in the first experiment.

[2]All the benchmark algorithms are available as Matlab toolbox: http://www.eecs.berkely.edu/~yang/software/l1benchmark/.

Algorithm 4.2 The outer (Dual EM) Algorithm. Following [72], we can obtain the updates of the hyper-parameters $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$.

$$
\begin{aligned}
\epsilon_1^{h+1} &= \frac{1}{n}\sum_{j=1}^{n}\pi(\hat{r}_{j1},\mu_{j1}^r;\boldsymbol{q}_1^h)\\
\epsilon_2^{h+1} &= \frac{1}{m}\sum_{j=n+1}^{n+m}\pi(\hat{r}_{j2},\mu_{j2}^r;\boldsymbol{q}_2^h)\\
\theta_1^{h+1} &= \frac{1}{n\epsilon_1^{h+1}}\sum_{j=1}^{n} g_{in}(\hat{r}_{j1},\mu_{j1}^r;\boldsymbol{q}_1^h)\\
\theta_2^{h+1} &= \frac{1}{m\epsilon_2^{h+1}}\sum_{j=n+1}^{n+m} g_{in}(\hat{r}_{j2},\mu_{j2}^r;\boldsymbol{q}_2^h)\\
\phi_1^{h+1} &= \frac{1}{n\epsilon_1^{h+1}}\sum_{j=1}^{n}\pi(\hat{r}_{j1},\mu_{j1}^r;\boldsymbol{q}_1^h)\\
&\quad\cdot\left(\left|\theta_1^h-\gamma(\hat{r}_{j1},\mu_{j1}^r;\boldsymbol{q}_1^h)\right|^2+\nu(\hat{r}_{j1},\mu_{j1}^r;\boldsymbol{q}_1^h)\right)\\
\phi_2^{h+1} &= \frac{1}{m\epsilon_2^{h+1}}\sum_{j=n+1}^{n+m}\pi(\hat{r}_{j2},\mu_{j2}^r;\boldsymbol{q}_2^h)\\
&\quad\cdot\left(\left|\theta_2^h-\gamma(\hat{r}_{j2},\mu_{j2}^r;\boldsymbol{q}_2^h)\right|^2+\nu(\hat{r}_{j2},\mu_{j2}^r;\boldsymbol{q}_2^h)\right)\\
\psi^{h+1} &= \frac{1}{m}\sum_{i=1}^{m}(|y_i-\hat{z}_i|^2+\mu_i^z)
\end{aligned}
$$

In the experiments, we explore the designed and benchmark algorithms using the Extended Yale B Face Database [35]. We choose 722 (19 images for each person) normal lighting conditioned images as the training data and another 266 images as the test data, which has more extreme lighting conditions. The images are re-sized from 192×168 to 32×28. A percentage of randomly chosen pixels from each of the test images are corrupted/replaced with i.i.d uniform distribution (e.g., uniform over [0,255] for the 8-bit images). We vary the percentages of corrupted pixels c from 10 to 90 percent. In this experiment, $\mathbf{\Phi} \in \mathbb{R}^{896\times1618}$, which has a high sampling rate to keep the mean value as small as possible. We test all the benchmark algorithms for all corruption cases within a fixed time limit, which is 80 seconds in our experiment. We use the SCI to calculate the recognition rate.

For the first experiment, we compared our approach with the Dual Augmented Lagrangian Method (DALM) [81], Primal Augmented Lagrangian Method (PALM) [81], Primal-dual Interior-point Algorithm (PDIPA) [49] and Truncated Newton Interior-point Method (TNIPM, known as L1LS) [48] in the review paper [81]. We also compared Dual BG-AD-GAMP (initialize $\epsilon_1 = \epsilon_2 = 0.08$ with Dual EM update) approach with the BG-AD-GAMP (initialize $\epsilon = 0.08$ with EM update) algorithm. In other words, we update the hyper-parameters of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ separately. In the experiments, we set both algorithms to have maximum 25 inner iterations and 200 outer iterations (EM step). In all the other algorithms, we let the number of iterations to 5000. The results of the recognition rates of the benchmark algorithms are shown in Fig. 4.4.1. As one can see from the figure, our algorithm has the best performance among all benchmarks in terms of recognition rate. For the BG-AD-GAMP, it has similar recognition rate DALM. Interestingly, Compared

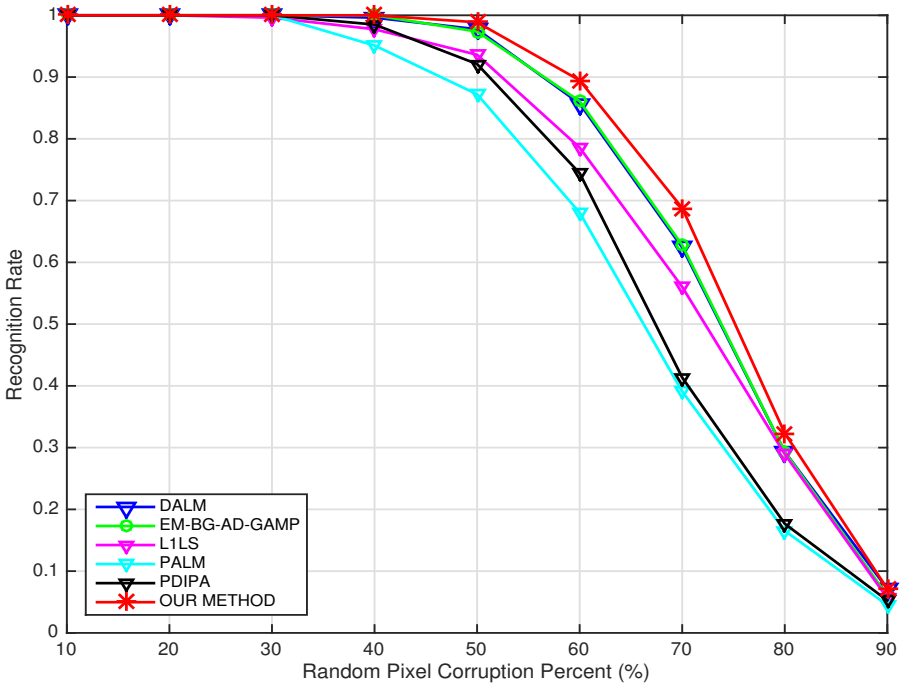


Figure 4.4.1: Recognition rates for different algorithms under different fractions of noise corruptions.

with our method, BG-AD-GAMP has lower recognition rate since it does not update the hyper-parameters separately.

In the second experiment, we test the convergence rate in terms of recognition rate. Here, we show the comparison of our method and DALM (best algorithm in the review paper [81]) under different fractions of corruptions c = 60%, 70%, 80% in Fig. 4.4.2. Our method achieves the best recognition rate as shown in Fig. 4.4.1 in 50 iterations. However, the DALM algorithm requires more iterations to reach its best.

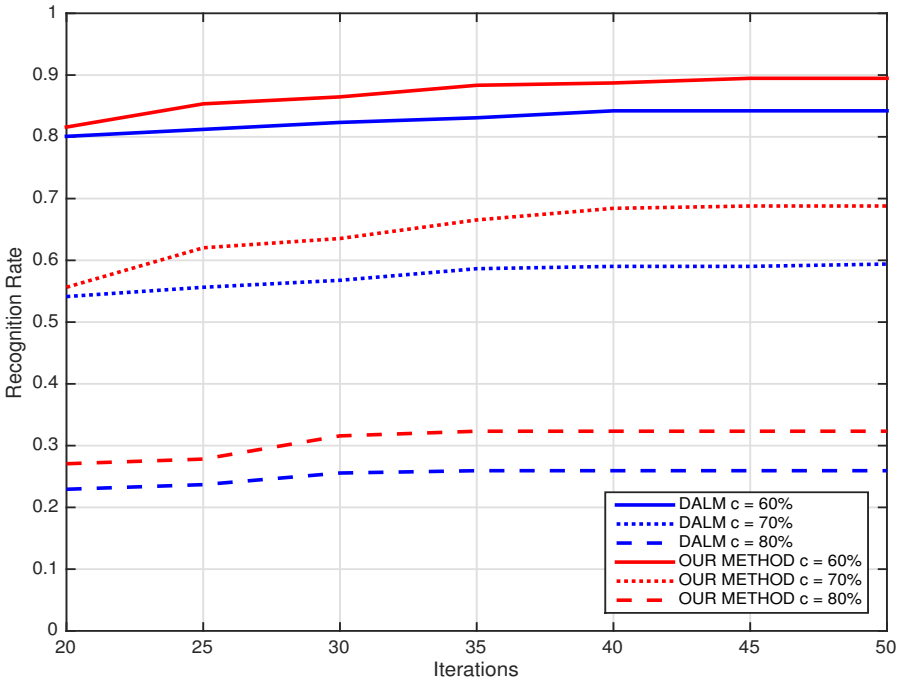


Figure 4.4.2: Comparison of our algorithm with DALM under different fractions of corrupted entries c = 60%, 70%, 80%. Our method: red lines. DALM algorithm: blue lines.

4.5 Conclusions

In summary, we adapt the combined AD-GAMP [74] and BG-GAMP [72] method to solve the robust face recognition problem. However, it achieved similar recognition rate with the best algorithm DALM. In order to improve the recognition performance, I introduced a dual updating approach for the unknown parameters. The designed algorithm has better performance in terms of recognition rate and has low computational cost.

Chapter 5

Conclusion and Future Work

Sparse representation is an interesting approach that extracts the hidden pattern from large data and reveals the anomalies. In this thesis, we studied several sparse representation methods and its applications.

For the first one, as discussed in Chapter 2, we focused on the low rank matrix completion problem, which refers to the spectral sparse. It can be applied to different real-time applications, ranging from the user recommendation system to the pedestrian detection. In this problem, we studied the so-called singularity issue in the ℓ_0-search problem. We study different methods to solve the singularity issue in ℓ_0-search. A rigorous analysis shows how the regularization technique may fail. The regularization term solves the discontinuous problem of the objective function. However, it will generate a local minimum at the neighbor of the singular points as a side effect and force the searching process away from the singular points. In order to address the singularity issue, we propose a continuous objective function to replace the original objective function. It opens a tunnel to letting the optimization process pass through the singular point. Furthermore, we use a quasi-Newton method to

implement the new approach since it exhibits low computational complexity and super linear convergence. The proposed method has the best performance among all tested algorithms even when the number of observations is close to the oracle rate. For future work, one may use the modified logistic function $s_\rho(\lambda)$ since it has similar 's' shaped curve instead of the modulation function $g_\rho(\lambda)$. Also, the logistic function has a nice property that its derivative can be expressed by $s'_\rho(\lambda) = s_\rho(\lambda)(1 - s_\rho(\lambda))$.

In the third Chapter, we apply dictionary learning algorithm to solve the BSS problem. Different from other dictionary based BSS algorithms, we use only one dictionary to sparsely represent different sources. We formulate the overall separation problem into two sub-problems and adapt the recently proposed SimCO optimization method [23] to solve both. The advantage of unifying the two stages is that, in practice, the same algorithm framework and codes can be used for both stages, thus significantly reducing the implementation effort. It was observed [23] that the singular point tend to be the major obstacle preventing the optimization process from converging to a global minimum. By adopting regularized SimCO, we are able to force the search path away from singular points and improve the performance. Also, we investigated the smoothed technique in Chapter 2 to solve the singular issue in SimCO, termed *smoothed SimCO*. Similarly, a continuous objective function is proposed to replace the original one. It has better performance than the regularized SimCO in the noiseless case and similar performance in the noise case. In dictionary learning, it remains open how to find an optimum choice of the redundancy factor $\tau = d/n$ of the over-complete dictionary. A higher redundancy factor leads to either more sparse representation or more precise reconstruction. Moreover, one has to consider the computational capabilities

when implementing the algorithms. From this point of view, it is better to keep the redundancy factor low. In the simulation, we have used 64 by 256 dictionary, which gives the redundancy factor $\tau = 256/64 = 4$. This choice is empirical: the sparse representation results are good and the computational cost is limited. A rigorous analysis on the selection of τ is still missing. The relation between the parameters λ, ϵ and noise standard deviation σ is also worth investigating. As presented in the first experiment on blind audio separation, the relation between λ and σ is discussed when the error bound ϵ is fixed in the sparse coding stage. One can roughly estimate the value of the parameter λ assuming the noise level is known a priori. Similar investigation is undertaken in [3], where the authors claim that when $\lambda \approx \sigma/30$, the algorithm achieved similar reconstruction performance under various σ's. From another perspective, the error bound ϵ is proportional to the noise standard deviation. It turns out that once a well approximated relation between ϵ and σ is obtained, one may get more precise estimation of parameter λ, rather than keeping ϵ fixed. This analysis, therefore, is counted as another open question. For the BSS problem in this thesis, we only considered the over-determined or even-determined mixing case, where the number of sources is larger than or equal to the number of mixtures. Moreover, the proposed framework can be extended potentially to a convolutive or underdetermined model, e.g., apply clustering method to solve the ill-posed inverse problem in underdetermined model; however, discussion on such an extension is beyond the scope of this thesis.

In the fourth Chapter, we studied the robust face recognition problem based on the variants of the AMP algorithm. We consider the sparse signal in two segments: sparse features and the anomalies. Different segments are assumed

to have different hyper-parameters. The EM algorithm is used to update the unknown parameters. The dual update algorithm has better performance in terms of recognition rate under certain noise levels and also exhibits low computational cost. For the robust face recognition problem, we proposed our approach for the SRC framework and used AMP based method to solve this problem. However, under this framework, we assume the additive noise (error) is dense while apply the AMP based method. For the original robust face recognition problem, it will be a very interesting research direction that considers a sparse additive noise in the GAMP.

Bibliography

[1] Netflix prize: http://www.netflixprize.com/index.html.

[2] H. Abdi and L. J. Williams. Principal component analysis. *WIREs Computational Statistics*, 2:433–459, 2010.

[3] V. Abolghasemi, S. Ferdowsi, and S. Sanei. Blind separation of image sources via adaptive dictionary learning. *IEEE Transactions on Image Processing*, 21(6):2921–2930, 2012.

[4] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[5] M. Aharon, M. Elad, and A. Brucketein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[6] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Inc, 1993.

[7] J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[8] A. Belouchrani and J. F. Cardoso. Maximum likelihood source separation for discrete sources. In *Proceedings of European Signal Processing Conference*, pages 768–771, 1994.

[9] J. Bobin, Y. Moudden, J. Starck, and M. Elad. Morphological diversity and source separation. *IEEE Signal Processing Letters*, 13(7):409–412, 2006.

[10] J. Bobin, J. Starck, J. Fadili, and Y. Moudden. Sparsity and morphological diversity in blind source separation. *IEEE transactions on Image Processing*, 16(11):2662–2674, 2007.

[11] N. Boumal and P.-A. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. In *Neural Information Processing Systems (NIPS)*, pages 406–414, 2011.

[12] M. Bronstein, M. Zibulevsky, and Y. Zeevi. Sparse ICA for blind separation of transmitted and reflected images. *International Journal of Imaging Science and Technology*, 15:84–91, 2005.

[13] F. Caltagirone, F. Krzakala, and L. Zdeborová. On convergence of approximate message passing. *Arxiv:1401.6384*, pages 1–5, 2014.

[14] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *arXiv:0912.3599*, 2009.

[15] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[16] E. J. Candès and T. Tao. The power of convex relaxation: near-

optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, May 2010.

[17] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, March 2008.

[18] J. F. Cardoso and A. Souloumiac. Blind beamforming for non-Gaussian signals. In *IEEE Proceedings of Radar and Signal Processing*, volume 140, pages 362–370, 1993.

[19] W. Dai, E. Kerman, and O. Milenkovic. A geometric approach to low-rank matrix completion. *IEEE Transactions on Information Theory*, 58(1):237–247, Jan. 2012.

[20] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, 55(5):2230–2249, May 2009.

[21] W. Dai, O. Milenkovic, and E. Kerman. Subspace evolution and transfer (SET) for low-rank matrix completion. *IEEE Transactions on Signal Processing*, 59(7):3120–3132, 2011.

[22] W. Dai, B. C. Rider, and Y. Liu. Volume growth and general rate quantization on Grassmann manifolds. In *IEEE Global Telecommunications Conference (Globecom)*, pages 1441 –1445, Nov. 2007.

[23] W. Dai, T. Xu, and W. Wang. Simultaneous codeword optimization (SimCO) for dictionary update and learning. *IEEE Transactions on Signal Processing*, 60(12):6340–6353, Dec. 2012.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of The Royal Statitisal Society, Series B*, 39(1):1–38, 1977.

[25] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[26] D. L. Donoho, A. Maleki, and A. Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences of the United States of America*, 106(45):18914–18919, 2009.

[27] D. L. Donoho, A. Maleki, and A. Montanari. Message passing algorithms for compressed sensing: I. Motivation and construction. In *IEEE Information Theory Workshop*, 2010.

[28] A. Edelman, T. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, April 1999.

[29] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[30] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, Dec. 2006.

[31] K. Engan, S. O. Aase, and J. H. Husoy. Method of optimal directions for frame design. In *Proceedings of IEEE International Conference on Acous-*

tics, Speech, and Signal Processing (ICASSP), volume 5, pages 2443–2446, 1999.

[32] K. Engan, K. Skretting, and J. Husoy. Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation . *Digital Signal Processing*, 17(1):32–49, 2007.

[33] M. Gaeta and J. L. Lacoume. Source separaion without prior knowledge: the maximum likelihood solution. In *Proceedings of European Signal Processing Conference*, pages 621–624, 1990.

[34] Q. Geng, H. Wang, and J. Wright. On the local correctness of ℓ_1 minimization for dictionary learning. *CoRR*, abs/1101.5672, 2011.

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

[36] I. F. Gorodnitsky, J. S. George, and B. D. Rao. Neuromagnetic source imaging with FOCUSS: A recursive weighted minimum norm algorithm. *Electroencephalography and Clinical Neurophysiology*, 95:231–251, 1995.

[37] R. Gribonval and S. Lesage. A survey of sparse component analysis for blind source separation: Principles, perspectives, and new challenges. In *Proceedings of European Symposium on Artificial Neural Networks*, pages 323–330, 2006.

[38] R. Gribonval and K. Schnass. Dictionary identification - sparse matrix-factorisation via ℓ_1-minimisation. *CoRR*, abs/0904.4774, 2009.

[39] J. P. Haldar and D. Hernando. Rank-constrained solutions to linear matrix equations using PowerFactorization. *IEEE Signal Processing Letters*, pages 16:584–587, 2009.

[40] N. J. A. Harvey, D. R. Karger, and S. Yekhanin. The complexity of matrix completion. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1103–1111. ACM Press, 2006.

[41] F. B. Hildebrand. *Advanced Calculus for Applications*. Prentice-Hall, 1976.

[42] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, May 1999.

[43] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. New York: Wiley-Interscience, May 2001.

[44] R. Jenatton, R. Gribonval, and F. Bach. Local stability and robustness of sparse dictionary learning in the presence of noise. *arXiv:1210.0685v1*, 2012.

[45] A. Jourjine, S. Rickard, and O. Yilmaz. Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2985–2988, 2000.

[46] K. Kayabol, E. E. Kuruoglu, and B. Sankur. Bayesian separation of images modeled with MRFs using MCMC. *IEEE Transactions on Image Processing*, 18(5):982–994, 2009.

[47] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, June 2010.

[48] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale ℓ_1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):606–617, Dec 2007.

[49] M. Kojima, N. Megiddo, and S. Mizuno. Theoretical convergence of large-step primal dual interior point algorithms for linear programming. *Mathematical Programing*, 59(1):1–21, 1993.

[50] M. Laan and S. Rose. Statistics ready for a revolution: Next generation of statisticians must build tools for massive data sets. *Amstat News*, 399:38–39, 2010.

[51] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *SIGCOMM*, pages 219–230, 2004.

[52] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[53] K. Lee and Y. Bresler. ADMiRA: atomic decomposition for minimum rank approximation. *IEEE Transactions on Information Theory*, 56(9):4402–4416, Sept. 2010.

[54] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, March 2010.

[55] A. Manoel, F. Krzakala, E. W. Tramel, and Z. Lenka. Sparse estimation with the swept approximated message-passing algorithm. *Arxiv:1406.4311*, pages 1–11, 2014.

[56] R. Meka, P. Jain, and I. S. Dhillon. Guaranteed rank minimization via singular value projection. *arXiv:0909.5457*, 2009.

[57] R. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

[58] J. Nocedal and S. J. Wright. *Numerical Optimization (2nd Edition)*. Springer New York, 2006.

[59] T. Papadopoulo and M. I. A. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *In Proceedings European Conference on Computer Vision (ECCV)*, pages 554–570. Springer, 2000.

[60] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 40–44, Nov 1993.

[61] G. Peyré, J. Fadili, and J-L. Starck. Learning adapted dictionaries for geometry and texture separation. In *Proceedings of SPIE Wavelet XII*, volume 6701, page 67011T, 2007.

[62] H. V. Poor and O. Hadjiliadis. *Quickest Detection*. Cambridge University Press, 2009.

[63] C. Qi, K. A. Gallivan, and P.-A. Absil. Riemannian BFGS algorithm with applications. In Moritz Diehl, Francois Glineur, Elias Jarlebring, and Wim Michiels, editors, *Recent Advances in Optimization and its Applications in Engineering*, pages 183–192. Springer Berlin Heidelberg, 2010.

[64] S. Rangan. Generalized approximate message passing for estimation with random linear mixing. In *Proceedings of IEEE International Symposium on Information Theory*, pages 2168–2172, 2011.

[65] S. Rangan, A. K. Fletcher, P. Schniter, and U. Kamilov. Inference for generalized linear models via alternating directions and bethe free energy minimization. *arXiv:1501.01797*, pages 1–20, 2015.

[66] S. Rangan, P. Schniter, and A. K. Fletcher. On the convergence of approximate message passing with arbitrary matrices. *arXiv:1402.3210*, 2014.

[67] B. Savas and L.-H. Lim. Quasi-Newton methods on Grassmannians and multilinear approximations of tensors. *SIAM Journal on Scientific Computing*, 32(6):3352–3393, 2010.

[68] K. Skretting and K. Engan. Recursive least squares dictionary learning algorithm. *IEEE Transactions on Signal Processing*, 58(4):2121–2130, Apr 2010.

[69] J. Starck, M. Elad, and D. L. Donoho. Redundant multiscale transforms and their application for morphological component analysis. *Advances in Imaging and Electron Physics*, 132:287–348, 2004.

[70] J. Tanner and W. Ke. Normalized iterative hard thresholding for matrix completion. *SIAM Journal on Scientific Computing*, 35(5), 2013.

[71] K. C. Toh and S. W. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6(3):615–640, 2010.

[72] J. Vila and P. Schniter. Expectation-maximization Bernoulli-Gaussian approximate message passing. In *Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 799–803, November 2011.

[73] J. Vila and P. Schniter. Expectation-maximization Gaussian-mixture approximate message passing. *IEEE Transactions on Signal Processing*, 61(1–19):4658–4672, 2013.

[74] J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborova. Adaptive damping and mean removal for the generalized approximate message passing algorithm. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2021–2025, 2015.

[75] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech and Language Processing*, 14(4):1462–1469, 2006.

[76] J. Wright, A. Ganesh, A. Yang, Z. Zhou, and Y. Ma. Sparsity and robustness in face recognition. *arXiv:1111.1014v1*, pages 1–12, 2011.

[77] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recog-

nition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[78] L. Xiong, X. Chen, and J. G. Schneider. Direct robust matrix factorizatoin for anomaly detection. In *International Conference on Data Mining (ICDM)*, pages 844–853. IEEE, 2011.

[79] T. Xu, W. Wang, and W. Dai. Sparse coding with adaptive dictionary learning for underdetermined blind speech separation. *Speech Communication*, 55(3):432–450, 2013.

[80] M. Yaghoobi, T. Blumensath, and M. E. Davies. Dictionary learning for sparse approximations with the majorization method. *IEEE Transactions on Signal Processing*, 57(6):2178–2191, 2009.

[81] A. Yang, Z. Zhou, A. G. Balasubramanian, S. S. Sastry, and Y. Ma. Fast ℓ_1 -minimization algorithms for robust face recognition. *IEEE Transactions on Image Processing*, 22(8):3234–3246, Aug 2013.

[82] X. Zhao, T. Xu, G. Zhou, W. Wang, and W. Dai. Joint image separation and dictionary learning. *18th International Conference on Digital Signal Processing, Santorini, Greece*, 2013.

[83] X. Zhao, G. Zhou, and W. Dai. Dictionary learning: a singularity problem and how to handle it. *In preparation.*

[84] X. Zhao, G. Zhou, and W. Dai. Smoothed SimCO for dictionary learning: handling the singularity issue. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[85] M. Zibulevsky and B. A. Pearlmutter. Blind source separation by sparse decomposition. *Neural Computation*, 13:863–882, 2001.