LEARNING FROM STRUCTURED DATA: THEORY, ALGORITHMS, AND APPLICATIONS

By

JIE SHEN

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Ping Li and Pranjal Awasthi

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October 2018

ABSTRACT OF THE DISSERTATION

LEARNING FROM STRUCTURED DATA: THEORY, ALGORITHMS, AND APPLICATIONS

By JIE SHEN

Dissertation Director:

Ping Li and Pranjal Awasthi

The last few years have witnessed the rise of the big data era, which features the prevalence of data sets that are high-dimensional, noisy, and dynamically generated. As a consequence, the gap between the limited availability of computational resources and the rapid pace of data generation has become ubiquitous in real-world applications, and has in turn made it indispensable to develop provable learning algorithms with efficient computation, economic memory usage, and noise-tolerant mechanisms.

Our work is driven inherently by practical large-scale problems, and the goal is to understand the fundamental limits imposed by the characteristics of the problems (e.g., high-dimensional, noisy, sequential), explore the benefits of geometric structures (e.g., sparsity, low rank), and offer scalable optimization tools to balance the trade-off between model accuracy, computational efficiency and sample complexity.

In the dissertation, we mainly investigate three important problems, as stated below.

- **High-dimensional statistics.** The large demand of learning from high-dimensional data where the number of attributes (i.e., data dimension) is of the same order as the number of samples or even larger has stimulated a large body of novel statistical paradigms, in which typically a low-dimensional structure is presumed to make the estimation possible. As an example, for rare diseases there is few patient data for research but usually they are caused by a small portion of factors such as the physical environment. It is henceforth crucial to distinguish the determinants from a large pool of possible attributes, namely the problem of variable selection (also known as support recovery). While it has been established that many convex programs consistently select the desired variables under certain conditions, the computational bottleneck has arguably hindered the application of these estimators to modern

data analytics. In this regard, we study a family of non-convex approaches and show that it admits a superior time complexity, a near-optimal sample size, and a broader range of applications.

- **Online and stochastic optimization.** It has been recognized that the challenges of the big data root not only in the high dimensionality, but also in the rapid pace of data generation. For example, there are millions of tweets per day, for which even storing the data becomes expensive. In order to process the huge volume of data, any practical algorithm has to be scalable, in the sense that the model updating and evaluation have to be online and efficient. This motivates us to examine problems from the perspective of optimization theory. In particular, we focus on those involving a low-rank or sparse structure, which has a variety of applications such as recommender systems and image de-noising. We develop novel algorithms whose time complexity is linear with the sample size, allowing a real-time response. Another salient feature is that the memory cost is a constant, i.e., independent of the sample size, making them an appealing mitigation to large-scale machine learning systems. Theoretically, we prove that the solution produced by our algorithms is accurate and is robust to various types of noise.

- **Estimation from quantized data.** While a large body of early work emphasizes on the observation model with real values, in practical applications the observations are often extremely quantized. Such low-bit observations not only save the storage, but also ease the process of data acquisition. For instance, ratings at Netflix are changed to either "thumbs up" or "thumbs down", since it is sometimes hard for an user to rate one star or two stars if he dislikes a movie. We study the problem of binary matrix completion, where many entries of the matrix are missing and the goal is to predict them. We present efficient and robust algorithms, together with a rigorous analysis and a comprehensive empirical illustration.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisors Ping Li and Pranjal Awasthi. Throughout the four years of my Ph.D., Ping gives me complete freedom to choose the research topics and offers high-level guidance, while Pranjal always serves as the source of knowledge for detailed problems. I learned a great deal during the discussion with them: research, teaching, and even academic career. It is in a large part to their encouragement and support that I had an extremely enjoyable experience at Rutgers and was able to work on a broad range of interesting problems.

I would also like to thank the excellent faculty members at Rutgers: William Steiger (who provided lots of reassuring advice on the career), Cun-Hui Zhang (who is a brilliant statistician and a prolific source of knowledge beyond statistics), Pierre Bellec (who is always approachable and happy to hear any of my ideas), Han Xiao, Thu D. Nguyen, Matthew Stone, Abdeslam Boularias, Zheng Zhang, and Anand D. Sarwate.

Before coming to Rutgers, I already had research experience when I was pursuing my master degree. I was truly fortunate to have the mentorship of Huan Xu who has a lifelong influence on shaping me as a researcher. In fact, it was the collaboration with Huan that I had the flavor of machine learning for the first time, which drove me to delve into the fascinating domain. It was also a delight to work with Yong Yu and Shuicheng Yan on computer vision – an amazing research area that attracted me to study computer science.

Many thanks to my peers and friends for their thought-provoking discussion on research: Guangcan Liu (from whom I learned a lot about scientific writing), Jian Wang (an expert in signal processing), Martin Slawski (really kind and methodical), Anshumali Shrivastava, Jiashi Feng, Yu-Xiang Wang, Xiao-Tong Yuan, Ji Zhang, Lezi Wang, Atul Dhingra, Nicholas Kleene.

Last but not least, I thank my family – especially my parents. They are always supportive to any of my decisions. My special thanks go to my co-author Jing Wang, who is now my beloved wife. She is always patient to listen to any of my feelings and gives me invaluable feedback, as well as her constant emotional support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the unprecedented growth of massive data sets in modern data analytics, problems being investigated in machine learning and statistics often feature high-dimensional, sequential, and quantized data. For instance, in YouTube millions of new videos are uploaded every day (i.e., sequential), each of which contains tremendous contents (i.e., high-dimensional) while the user feedback is simply "thumbs up" or "thumbs down" (i.e., quantized). These problem characteristics pose new challenges to computer scientists and statisticians, both in the statistical and computational aspects.

On the statistical side, a long-term research line is to understand the fundamental limits imposed by the properties of the problems, and to determine the sample size under which accurate estimation of model parameters is possible. The question has been well-understood if there are sufficient samples or the sample size tends to infinity, which is known as asymptotic analysis. However, in the high-dimensional regime, the number of observations is typically of the same order of, or even smaller than the number of unknown parameters. Classical results immediately break down in this situation. As a matter of fact, it is not possible to identify the model without further information.

On the computational side, parameter estimation usually boils down to solving an optimization problem, either convex or non-convex, and the regard (or objective) is to design efficient algorithms that achieve the optimal computational efficiency. While there have been numerous solvers developed in the last decades, they are typically not scalable to very large-scale data sets since the computational cost is polynomial in the problem size. Perhaps a more serious issue is that these elegant solvers are batch methods in nature, meaning that it is rather expensive, or even impossible to apply them to streaming data.

The dissertation concerns exactly both aspects, and our goal is to leverage the hidden structure of the problem ir order to design provable and efficient algorithms for real-world applications. The major observation made throughout the dissertation is that in most of the applications, the data lying in a high-dimensional space typically exhibits a low-dimensional structure, captured by some notion of sparsity. Such an appealing behavior essentially reduces the problem complexity, eliminates the statistical issue of model identifiability, and guides the design of efficient algorithms. To be more concrete, we will present our novel results for three research topics that play a crucial role in machine learning applications: **(1)** high-dimensional statistics, which aims to detect useful attributes from a large pool of features; **(2)** online and stochastic optimization, which serves as the fundamental technique for large-scale learning systems; and **(3)** estimation from quantized data, which allows statistical inference from extremely simple feedback (or observation).

1.1 High-Dimensional Statistics

Consider the linear regression model

$$\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{e},$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ is the design matrix, $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ is the unknown parameter to be estimated, and $\boldsymbol{y} \in \mathbb{R}^n$ is the observation corrupted by the unknown noise $\boldsymbol{e} \in \mathbb{R}^n$. A fundamental question to ask is when it is possible to estimate $\bar{\boldsymbol{x}}$ from the knowledge of $(\boldsymbol{A}, \boldsymbol{y})$. This problem has a very long history in statistics and classical results assert that as soon as the sample size n is sufficiently large, the estimation error tends to zero. However, in many applications, typically there is a limited number of samples. For example, consider the Lou Gehrig's disease. Each year there are only 2 out of one hundred thousand people are affected by this disease. Therefore, for researchers almost no data is available for carrying out medical diagnosis. This partially explains the unfortunate fact that the causal elements are still not known. It is also the case that patients are worried about privacy, and do not want to share their personal information for research.

When there is insufficient data, classical results fail to offer consistency guarantees for parameter estimation. In fact, when $n \ll d$, the model is even not identifiable, meaning that there exist many (or possibly infinite) different parameters $\bar{\boldsymbol{x}}$ that produce the same response $\boldsymbol{y}$. Estimation in this regime is thus referred to as high-dimensional statistics. While it is not possible to perform

parameter estimation in general cases, the key assumption that the underlying parameter is sparse quickly changes the story. Here, we say $\bar{\boldsymbol{x}}$ is sparse if most of its components are zero. The major motivation to consider such a special structure is that usually only a small portion of the features contribute to model prediction, while the remaining might be redundant or have only minor influence. The community then moved on to incorporate such prior knowledge into algorithm design, either convex or non-convex. From a mathematical perspective, the assumption essentially reduces the problem to predicting the positions and values of the non-zero elements of $\bar{\boldsymbol{x}}$. If we omit the computational cost, we can enumerate all possible positions and fit the data restricted to these coordinates. Until now, a number of efficient algorithms have been proposed which admit comparable statistical error rate to the brute-force approach [41, 140, 37, 142, 103, 119].

1.1.1 Our Contributions

We study in Section 2.2 the problem of recovering the support of a sparse signal from its compressed measurements, i.e., estimating the positions of the non-zero elements of $\bar{\boldsymbol{x}}$ when $n \ll d$. We first point out that previous work suffers a high computational cost for the sake of support recovery, or incurs unbounded iteration complexity. From the perspective of optimization, the algorithms developed in these work are not scalable to massive data sets. Then we offer a novel analysis of a popular non-convex algorithm, and provably show that under standard conditions, the algorithm is guaranteed to recover the support in few iterations, and has a low computational cost per iteration. In order to study the trade-off between computational complexity and statistical accuracy, we generalize our analysis to a family of non-convex algorithms in Section 2.3. It is shown that if the data is good enough (in some sense), then it is possible to design algorithms that attain the best of the two worlds. Otherwise, one has to trade the running time for model accuracy (or vice versa). Our analysis is applied to, and justified by prevalent statistical models, and is further accompanied by illustrative experiments.

1.2 Online and Stochastic Optimization

Orthogonal to the high-dimensional regime, in a variety of machine learning applications it is ubiquitous to process data sets containing billions of samples, or their sizes grow dynamically in a rapid

pace. For example, there are thousands of tweets appearing online every minute, and the system needs to detect the trending topics for user recommendation. In this regard, the primary concerns are designing algorithms that are capable of performing model updating and prediction in an online and real-time fashion, with a cheap computational cost – typically linear in the sample size.

1.2.1 Sparsity-Constrained Minimization and Our Results

We will study three general problems involving structured constraints that are either sparse or low-rank. The first problem is sparsity-constrained optimization:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}; Z_1^n), \quad \text{s.t. } \|\boldsymbol{x}\|_0 \leq s,$$

where $Z_1^n := \{Z_i\}_{i=1}^n$ is the set of training samples and $s < d$ is a positive integer. If we consider the linear regression model in the preceding section, then $Z_1^n = (\boldsymbol{A}, \boldsymbol{y})$ and $F(\boldsymbol{x}; Z_1^n) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$. The sparsity constraint is imposed to guarantee model interpretation. Note that the constraint is non-convex in nature, which is the major barrier to carry out theoretical analysis of convergence. This is in stark contrast to the Lasso program [140]

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}; Z_1^n), \quad \text{s.t. } \|\boldsymbol{x}\|_1 \leq \lambda.$$

The ℓ_1-norm in the above expression is convex and encourages sparsity with a proper choice of $\lambda > 0$. Yet, the main issue of Lasso is that it is in general difficult to characterize the sparsity of the solution. Thus we choose to stay at the non-convex formulation.

Unlike the compressed sensing problem where $n \ll d$, here we think of both quantities as large. It is known that when the data is not good (in some sense), the computational cost of batch methods is quadratic in n, making existing solvers not scalable [78].

Our Contributions

In Chapter 3 we present the first stochastic solver for the ℓ_0-norm constrained non-convex program. It converges to a global optimum with a linear rate and has a low computational complexity. The total running time, even when the data is not good, is linear in the sample size. Prior to our work, even

the convergence behavior was not clear for such stochastic solvers. Along with the development of our algorithm, we also offer a tight bound for the expansiveness of the hard thresholding operator that is invoked to project iterates onto the ℓ_0-ball. We show that with the established bound, most of the theoretical results for hard thresholding based algorithms can be significantly improved.

1.2.2 Low-Rank Matrix Recovery and Our Results

In the last decades, problems involving a low-rank structure have been widely investigated and have found successful applications in recommender systems and image denoising. In recommender systems, we are given a set of users and items (e.g., movies). Each user can rate the movie that he watched, and the goal is to anticipate the potential movies that a user might be interested in. Hence, if we organize the ratings as a matrix, where each row corresponds to an individual and each column represents a movie, then item recommendation reduces to filling in the missing entries of the matrix.

Mathematically, suppose that the groundtruth rating matrix is $\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n) \in \mathbb{R}^{d\times n}$, and Ω is the set indexing the positions of observed entries (i.e., the ratings from users). We are interested in recovering $\boldsymbol{Z}$ from $\boldsymbol{Z}_\Omega$. Alternatively, the observations $\{y_{ij}\}$ are given by

$$y_{ij} = \boldsymbol{c}_i^\top \boldsymbol{Z} \boldsymbol{c}_j,\ (i,j) \in \Omega,$$

where $\{\boldsymbol{c}_i\}$ is the canonical basis. This is quite reminiscent of the compressed sensing problem, and we know that extra assumptions have to be made in order to estimate $\boldsymbol{Z}$. For example, the underlying matrix $\boldsymbol{Z}$ is assumed to be low-rank in many problems [58]. To aid intuition, recall that any matrix $\boldsymbol{Z}$ can be decomposed into $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{V}^\top$. In the context of item recommendation, the ith row of $\boldsymbol{U}$ is associated with the ith user whereas the jth row of $\boldsymbol{V}$ is associated with the jth item, and the rating y_{ij} is determined by both. Then a low-rank structure would imply that there are not too many factors influencing the user's evaluation on an item.

In [36], it was proved that under suitable conditions, an exact estimate of $\boldsymbol{Z}$ can be obtained by solving the following convex program:

$$\min_{\boldsymbol{X}\in\mathbb{R}^{d\times n}} \|\boldsymbol{X}\|_*, \quad \text{s.t. } (\boldsymbol{X})_\Omega = (\boldsymbol{Z})_\Omega.$$

Note that the nuclear norm $\|\boldsymbol{X}\|_*$ is the sum of the singular values of $\boldsymbol{X}$, which is a convex relaxation of the rank function. A more general formulation is to consider a noisy matrix recovery model where each observation is corrupted:

$$\min_{\boldsymbol{X}\in\mathbb{R}^{d\times n},\boldsymbol{E}\in\mathbb{R}^{d\times n}} \|\boldsymbol{X}\|_* + \lambda h(\boldsymbol{E}), \quad \text{s.t. } \boldsymbol{Z} = \boldsymbol{X} + \boldsymbol{E},$$

where $h(\cdot)$ is some regularization determined by the structure of the noise $\boldsymbol{E}$. For example, if $\boldsymbol{E}$ is assumed to be sparse, we may choose $h(\boldsymbol{E}) = \sum_{i,j} |e_{ij}|$ as shown in [34]. Note that the missing entries correspond to the case that the noise $\boldsymbol{E}$ happens to annihilate the clean data $\boldsymbol{X}$ at the positions indexed by Ω.

Our Contributions

While there have been a large number of algorithms for solving the nuclear-norm based convex programs, they either incur a computational cost quadratic in the sample size, or have to load all the data into memory for model updating. For this reason, it is prohibitive to apply them to large-scale problems. In Chapter 4, we present an efficient online minimization algorithm that deals with samples in an online manner. Namely, our memory cost is independent of the sample size. Further, the computational complexity is linear in the sample size which is the best one could wish for in most cases. We also prove that the solution produced by our algorithm asymptotically converges to a stationary point of the expected loss.

1.2.3 Low-Rank Subspace Clustering and Our Results

Clustering plays a crucial role in unsupervised machine learning, where the goal is to group similar data points into a number of clusters. While there are many successful algorithms such as k-means, they often do not capture the intrinsic structure of the data. For instance, it is not clear how to integrate the low-rank constraint into k-means. [56] proposed a novel formulation, which attempts to express each data point $\boldsymbol{z}_i$ as a linear combination of the remaining:

$$\boldsymbol{Z} = \boldsymbol{Z}\boldsymbol{X}, \ \boldsymbol{X} \in \mathbb{R}^{n\times n}.$$

The solution $\boldsymbol{X}$ is obviously not unique, and low-rank representation [93] seeks the one with the lowest rank

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times n}} \|\boldsymbol{X}\|_*, \quad \text{s.t. } \boldsymbol{Z}=\boldsymbol{Z}\boldsymbol{X},\ \boldsymbol{X}\in\mathbb{R}^{n\times n}.$$

Since $\operatorname{rank}(\boldsymbol{Z}) \leq \operatorname{rank}(\boldsymbol{Z}\boldsymbol{X})$, a low-rank solution $\boldsymbol{X}$ implies the recovery of a low-rank structure of the data. In this light, the above program is particularly powerful when the true data is low-rank, as reported in their original work. However, the main shortcoming is that the variable is an $n \times n$ matrix. Thus, the storage and computation are both quadratic in the sample size.

Our Contributions

We present the first practical algorithm for the low-rank representation problem in Chapter 5. It reduces the memory footprint from $\mathcal{O}\left(n^2\right)$ to $\mathcal{O}(dr)$, where r is the rank of the data matrix naturally satisfying $r < d \ll n$. Its computational cost is linear in the sample size, which is orders of magnitude more efficient than state-of-the-art solvers. More importantly, our algorithm can be implemented in an online fashion, and can cluster data points that are dynamically generated. We prove that the gains in computational and in memory efficiency do not sacrifice too much in model accuracy.

1.3 Estimation from Quantized Data

Quantization is the process of converting an input from continuous or real-valued domains to a discrete set. The classification problem can indeed be regarded as learning from quantized data: given a set of training data $\{\boldsymbol{a}_i\}_{i=1}^n \subset \mathbb{R}^d$ and their labels $\{y_i\}_{i=1}^n$, where

$$y_i = \operatorname{sign}(\boldsymbol{a}_i \cdot \bar{\boldsymbol{x}}),$$

estimate the underlying parameter $\bar{\boldsymbol{x}}$. In learning theory, this problem is known as learning half-spaces dating back to [123], whereas in signal processing, it is called 1-bit compressed sensing [25, 70, 116]. Such low-bit observations not only save the storage, but also ease the process of data acquisition and algorithm implementation on hardware. For example, after watching a video in YouTube, it is more transparent to ask the user whether he likes or dislikes the video than to have

him give a rating ranging from 1 to 10.

Formally, we study the 1-bit matrix completion problem. It follows the standard setup in matrix completion: many entries are missing in a large matrix. However, now the observations are assumed to be quantized to either 1 or -1:

$$y_{ij} = \delta_{ij} \cdot \operatorname{sign}\left(m_{ij} + e_{ij}\right), \ (i,j) \in \Omega,$$

where e_{ij} is some noise before quantization, δ_{ij} is the sign flipping noise, and $\boldsymbol{M} = (m_{ij})$ is the true, low-rank matrix to be estimated. Owing to the quantization, the observation $\boldsymbol{Y} = (y_{ij})$ is no longer low-rank. Hence one cannot apply the traditional technique to solve the problem.

1.3.1 Our Contributions

In Chapter 6 We propose a convex program to estimate the true matrix $\boldsymbol{M}$ in which each observed entry is corrupted by both pre-quantization and sign flipping noise. We provably show that our estimator is robust to both kinds of noise. An upper and lower bound of the statistical error rate is presented, showing that the sample complexity of our method is near-optimal.

1.4 Acknowledgment of Previous Publications

The results of Chapter 2 and Chapter 3, which have been published in ICML'17 [129], NIPS'17 [130], and to be published in JMLR [131], are joint work with Ping Li. The results of Chapter 4 and Chapter 5, which have been published in NIPS'14 [133], Machine Learning [134] and ICML'16 [132], are joint work with Huan Xu and Ping Li. Finally, the results of Chapter 6 are joint work with Pranjal Awasthi and Ping Li, and are under review.

1.5 Notations

Before delivering the main results, we mention some notations and conventions that are involved throughout the dissertation. We use bold lowercase letters, e.g., $\boldsymbol{v}$, to denote a vector (either column or row) and its ith element is denoted by v_i. Several norms will be used for a vector $\boldsymbol{v} \in \mathbb{R}^d$: the ℓ_2-norm $\|\boldsymbol{v}\| := \sqrt{\sum_{i=1}^d v_i^2}$, the ℓ_1-norm $\|\boldsymbol{v}\|_1 := \sum_{i=1}^d |v_i|$, and the infinity norm $\|\boldsymbol{v}\|_\infty :=$

$\max_{1\le i\le d}|v_i|$. The support set of $\boldsymbol{v}$, i.e., indices of non-zero entries, is denoted by $\operatorname{supp}(\boldsymbol{v})$, while that of the k largest elements (in magnitude) is denoted by $\operatorname{supp}(\boldsymbol{v},k)$. The cardinality of $\operatorname{supp}(\boldsymbol{v})$ is written as $|\operatorname{supp}(\boldsymbol{v})|$ or $\|\boldsymbol{v}\|_0$.

We write bold capital letters such as $\boldsymbol{M}$ for matrices and its (i,j)-th entry is denoted by m_{ij}. The ith row and jth column of a matrix $\boldsymbol{M}$ are denoted by $\boldsymbol{m}(i)$ and $\boldsymbol{m}_j$ respectively. The transpose of a matrix $\boldsymbol{M}$ is denoted by $\boldsymbol{M}^\top$. For a square matrix $\boldsymbol{M}$, its trace is denoted by $\operatorname{Tr}(\boldsymbol{M})$. Suppose that $\boldsymbol{M}\in\mathbb{R}^{n\times d}$ has rank r. Let $\boldsymbol{M}=\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ be the singular value decomposition of $\boldsymbol{M}$, where $\boldsymbol{U}\in\mathbb{R}^{n\times r}$ contains the left singular vectors, $\boldsymbol{V}\in\mathbb{R}^{d\times r}$ contains the right singular vectors, and $\boldsymbol{S}\in\mathbb{R}^{r\times r}$ is a diagonal matrix containing the singular values. Then the Frobenius norm of $\boldsymbol{M}$ is given by $\|\boldsymbol{M}\|_F:=\sqrt{\sum_{i=1}^r s_{ii}^2}=\sqrt{\sum_{i=1}^n\sum_{j=1}^d m_{ij}^2}$, the spectrum norm is $\|\boldsymbol{M}\|:=\max_{1\le i\le r}s_{ii}$ and the nuclear norm is written as $\|\boldsymbol{M}\|_*=\sum_{i=1}^r s_{ii}$.

The capital upright letter C and its subscript variants (e.g., $\mathrm{C}_0,\mathrm{C}_1$) are reserved for absolute constants whose values may change from appearance to appearance.

Chapter 2

High-Dimensional Statistics: From Convex to Non-Convex

2.1 Background

This chapter is concerned with the problem of recovering an *arbitrary* sparse signal from a set of its (compressed) measurements. We say that a signal $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ is s-sparse if there are no more than s non-zeros in $\bar{\boldsymbol{x}}$. This problem, together with its many variants, have found a variety of successful applications in bioinformatics [111], statistics [140, 55], signal processing [41, 51, 50, 39] and mathematical science [40], to name just a few. Of particular interest are **(1)** $\bar{\boldsymbol{x}}$ is the true signal and only a small number of linear measurements are given, referred to as compressed sensing; **(2)** $\bar{\boldsymbol{x}}$ is the global optimum of sparsity-constrained non-convex programs.

Mathematically, the observation model of compressed sensing is as follows:

$$\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{e}, \tag{2.1}$$

where the sensing matrix $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ can be chosen by the user and $\boldsymbol{e} \in \mathbb{R}^n$ is some unknown noise. The fundamental question to ask is when the signal $\bar{\boldsymbol{x}}$ can be estimated from the knowledge of $\boldsymbol{A}$ and the response $\boldsymbol{y} \in \mathbb{R}^n$. If the sample size n is greater than the ambient dimension d, it is well recognized that the problem can be solved by empirical risk minimization. Nevertheless, compressed sensing considers the high-dimensional regime that n is much smaller than d, which

means that (2.1) is an underdetermined system. Thus, in general it is not possible to recover the underlying signal $\bar{\boldsymbol{x}}$. Motivated by the observation that in a wide range of applications only a small number of attributes contribute to model prediction [111], the signal $\bar{\boldsymbol{x}}$ is typically assumed to be sparse. Such a simple structure radically changes the premise, and a natural approach is to enumerate all possible support sets of $\bar{\boldsymbol{x}}$ (i.e., the positions of its non-zero elements), followed by minimizing the empirical risk restricted on the support set. This is, unfortunately, NP-hard. However, what makes high-dimensional statistics and compressed sensing a nice story is that, in spite of the fact that the natural approach is NP-hard, there are powerful convex formulations that perform almost as well. For example, when the observation $\boldsymbol{y}$ is noise-free, [41] proposed the following program termed basis pursuit:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{x}\|_1, \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}.$$

It was shown in [37] that the optimal solution returned by basis pursuit exactly recovers $\bar{\boldsymbol{x}}$ under some conditions. For noisy observation models, one can make use of the Lasso estimator [140] given by the optimum of the following program:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_1,$$

where $\lambda > 0$ is a hyper-parameter to be tuned. Again, it was shown that Lasso enjoys appealing statistical property: its global optimum is close enough to the true signal [146]. Compared to the brute-force approach, solving a convex program is more efficient. In fact, the computational complexity is polynomial with respect to the problem size (n, d).

Parallel to the development of convex programs, a large body of work is devoted to greedy algorithms that are computationally more efficient [141, 45, 20, 28, 62, 60]. These algorithms usually start with an initial guess, and attempt to gradually refine the estimate by gradient descent or boosting. For instance, as one of the earliest pursuit algorithms, orthogonal matching pursuit (OMP) [115] repeatedly picks a coordinate as the potential support of a solution. While OMP may fail for some deterministic sensing matrices, [141, 142] showed that it recovers the true signal with high probability when using random matrices $\boldsymbol{A}$ such as Gaussian. Inspired by the success of OMP, the

two concurrent work of compressive sampling matching pursuit (CoSaMP) [103] and subspace pursuit (SP) [45] made improvement by selecting multiple coordinates followed by a pruning step in each iteration, and the recovery condition was framed under the well-known restricted isometry property (RIP) [37]. Interestingly, the more careful selection strategy of CoSaMP and SP leads to an optimal sample complexity. The iterative hard thresholding (IHT) algorithm [46, 19, 20] gradually refines the iterates by gradient descent along with truncation. [60] then developed a concise algorithm termed hard thresholding pursuit (HTP), which combined the idea of CoSaMP and IHT, and showed that HTP is superior to both in terms of the RIP condition.

Another quintessential example is the sparsity-constrained minimization program recently considered in machine learning [159, 10, 75, 131], for which the goal is to efficiently learn the global sparse minimizer $\bar{\boldsymbol{x}}$ from a set of training data. Formally, these work concerns solving

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}; Z_1^n), \quad \text{s.t. } \|\boldsymbol{x}\|_0 \leq s, \tag{2.2}$$

where $Z_1^n := \{Z_i\}_{i=1}^n$ is the set of training data. Typical examples include the sparse linear regression, sparse logistic regression, and sparse support vector machine (SVM), as described below:

- Sparse Linear Regression: for all $1 \leq i \leq n$, we have $Z_i = (\boldsymbol{a}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ and the loss function $F(\boldsymbol{x}; Z_1^n) = \frac{1}{2n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2$ is the least-squares;
- Sparse Logistic Regression: for all $1 \leq i \leq n$, we have $Z_i = (\boldsymbol{a}_i, y_i) \in \mathbb{R}^d \times \{+1, -1\}$ and the negative log-likelihood is penalized, i.e., $F(\boldsymbol{x}; Z_1^n) = \frac{1}{n} \sum_{i=1}^n \log\left(1 + \exp\left(-y_i \boldsymbol{a}_i \cdot \boldsymbol{x}\right)\right)$;
- Sparse SVM: for all $1 \leq i \leq n$, we have $Z_i = (\boldsymbol{a}_i, y_i) \in \mathbb{R}^d \times \{+1, -1\}$ and $F(\boldsymbol{x}; Z_1^n) = \frac{1}{n} \sum_{i=1}^n \max\{0, 1 - y_i \boldsymbol{a}_i \cdot \boldsymbol{x}\}$ is the hinge loss.

Though in most cases, the underlying signal can be categorized into either of the two classes, we note that it could also be other object such as the parameter of generalized linear models [105]. Hence, for a unified analysis, this chapter copes with an arbitrary sparse signal and the results to be established quickly apply to the special instances above.

It is also worth mentioning that while one can characterize the performance of an algorithm and can evaluate the obtained estimate from various aspects, we are specifically interested in the quality of support recovery. Recall that for sparse recovery problems, there are two prominent metrics: the

ℓ_2-distance and the ℓ_0-distance. The former one essentially requires that for a given error $\epsilon > 0$, the solution $\hat{\boldsymbol{x}}$ returned by an algorithm should approximate $\bar{\boldsymbol{x}}$ in ℓ_2-metric, namely

$$\|\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}\| \leq \epsilon.$$

Theoretical results phrased in terms of the ℓ_2-metric is also referred to as parameter estimation, on which most of the previous work emphasized. Under this metric, many popular algorithms, e.g., the Lasso [140, 146] and hard thresholding based algorithms [46, 20, 103, 45, 60, 131], are guaranteed with accurate approximation up to the noise level. For linear regression, it is sometimes useful to consider the prediction accuracy [119]

$$\|\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{A}\bar{\boldsymbol{x}}\| \leq \epsilon,$$

though under some conditions it is implied by the performance of parameter estimation.

Support recovery is another important factor to evaluate an algorithm, which is also known as feature selection or variable selection. It measures the discrepancy of the support set:

$$\|\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}\|_0 \leq \epsilon.$$

As one of the earliest work, [141] offered sufficient and necessary conditions under which orthogonal matching pursuit and basis pursuit identify the support. The theory was then developed by [164, 161, 146] for the Lasso estimator and by [157] for the garrotte estimator. Typically, recovering the support of a target signal is more challenging than parameter estimation. For instance, [17] showed that the restricted eigenvalue condition suffices for the Lasso to produce an accurate estimate whereas in order to recover the sign pattern, a more stringent mutual incoherence condition has to be imposed [146]. However, as has been recognized, if the support is detected precisely by a method, then the solution admits the optimal statistical rate [146]. In this regard, research on support recovery continues to be a central theme in recent years [162, 163, 158, 24, 113].

Our work follows this line and studies the support recovery performance of hard thresholding based algorithms, which enjoy superior computational efficiency to the convex programs when manipulating a huge volume of data [143]. To be concrete, we will study the hard thresholding

pursuit algorithm. It was originally presented by [60] for recovering the true signal in compressed sensing [50]. [159] suggested using the HTP algorithm for general sparsity-constrained machine learning problems, and they showed that the solution returned by HTP converges with a geometric rate. Very recently, a rigorous theoretical analysis on when HTP guarantees support recovery was independently carried out by [24] and [158]. In [24], they considered the compressed sensing problem and illustrated that HTP recovers the support of the true signal in finite iterations if the restricted isometry property (RIP) condition holds [37]. [158] showed that in some situations, HTP guarantees support recovery without assuming the RIP condition.

Although these appealing theoretical results characterize the behavior of HTP in particular regimes, it turns out that a thorough understanding on when HTP identifies the support of an *arbitrary* sparse signal is missing in the literature. To be more precise, the RIP condition used in [24] amounts to imposing a small condition number for the underlying problem, which is not practical for machine learning applications where the condition number may grow with the sample size. To guarantee support recovery of an s-sparse signal, [158] required that the signal of interest is the unique global minimizer of a sparsity-constrained program (which invokes the RIP condition), or that HTP maintains denser iterates (which hinders a fast update). This poses an interesting question of whether HTP is able to recover the support without the RIP assumption, or the optimality of the signal, or the relaxed sparsity.

We will also investigate the trade-off between computational efficiency and statistical accuracy of hard thresholding based algorithms. To this end, we appeal to the partial hard thresholding operator [74] which unifies a family of non-convex algorithms. We establish general results which in turn indicate the best known iteration complexity for specific instances such as HTP and orthogonal matching pursuit with replacement (OMPR) [73]. It is also worth mentioning that, though our analysis hinges on the PHT operator, the support recovery results to be established are stronger than the results in [74] since they only showed parameter estimation of PHT. Finally, while a couple of previous work considered signals that are not exactly sparse [24], we in this chapter focus on the sparse case. Extensions to the generic signals are left as interesting future directions.

2.1.1 Contributions

We make the following contributions in this chapter. First, we explore in Section 2.2 the support recovery performance of HTP, and provably show that for well-conditioned (to be clarified) problems, it exactly recovers the support in few iterations. For the cases where features are heavily correlated, we prove that with a slight modification of the algorithm, all the desired features can be detected. The proofs can be found in Section 2.B, with a few technical lemmas provided in Section 2.A. Then we move on to the more general PHT algorithm in Section 2.3, and present the first analysis of the trade-off between computational efficiency and model accuracy for a family of non-convex methods. The proofs are deferred to Section 2.C. We also illustrate that our theoretical results hold for a wide range of statistical models, which is further justified by a comprehensive set of numerical experiments.

2.1.2 Notation

For an integer d, suppose that $\Omega \subset \{1, 2, \ldots, d\}$ is an index set. Then for $\boldsymbol{v} \in \mathbb{R}^d$, $\boldsymbol{v}_\Omega$ can either be explained as an $|\Omega|$-dimensional vector or a d-dimensional vector with the elements outside of Ω set to zero. The s-sparse vector $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ is the target signal we aim to recover, and we reserve the capital letter S for its support. We define $\bar{\boldsymbol{x}}_{\min} > 0$ as the absolute value of the smallest element (in magnitude) of $\bar{\boldsymbol{x}}_S \in \mathbb{R}^s$. With a slight abuse of the notation, $\nabla_k F(\bar{\boldsymbol{x}})$ should be explained as the vector consisting of the top k elements (in magnitude) of $\nabla F(\bar{\boldsymbol{x}})$ rather than the kth component of $\nabla F(\bar{\boldsymbol{x}})$.

2.2 Support Recovery with Hard Thresholding Pursuit

In this section, we introduce the problem setting and some preliminary consequences. Then we present the deterministic results regarding the support recovery performance, followed by a discussion that related these to concrete statistical models. To be clear, the target signal $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ we consider in this chapter is only endowed with sparsity.

Let us first motivate the HTP algorithm. Consider the sparsity-constrained optimization program (2.2). While obtaining an exact solution is NP-hard in general, it is possible to obtain an approximate one in an efficient manner. To this end, a natural and popular optimization method is

gradient descent [107], which starts with an initial guess $\boldsymbol{x}^0$ and updates it by

$$\boldsymbol{x}^t \leftarrow \boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}),\ t = 1,\ 2,\ \ldots$$

Above $\eta > 0$ is the step size to be tuned. Yet, notice that there is a sparsity constraint in (2.2). While the zero-norm is discrete and non-convex, projection onto the zero-norm ball is quite simple: we keep the largest s elements (in magnitude) and set the remaining to zero. Such a projection, denoted by $\mathcal{H}_k(\cdot)$, is called hard thresholding. As a matter of fact, for any $\boldsymbol{v} \in \mathbb{R}^d$,

$$\mathcal{H}_k(\boldsymbol{v}) := \underset{\boldsymbol{u} \in \mathbb{R}^d}{\arg\min} \ \|\boldsymbol{u} - \boldsymbol{v}\|, \quad \text{s.t. } \|\boldsymbol{u}\|_0 \leq k. \tag{2.3}$$

Hence, we may combine gradient descent, which is able to decrease the objective function value, and hard thresholding, which ensures the feasibility of all iterates. That gives the IHT algorithm

$$\begin{aligned} \boldsymbol{b}^t &\leftarrow \boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}), \\ \boldsymbol{x}^t &\leftarrow \mathcal{H}_k\left(\boldsymbol{b}^t\right). \end{aligned}$$

Note that we allow the sparsity k of the iterates to be greater than, or equal to the true sparsity s. This offers more flexibility since in practical applications, s may be unknown. On the other hand, we will show that such a relaxation is crucial for some difficult problems.

The IHT algorithm works well in practice, and enjoys elegant theoretical guarantee [20, 75]. However, with a more careful analysis, [60] pointed out that if the support of $\boldsymbol{x}^t$ happens to be the true support S, then we can eliminate the sparsity constraint, and fully minimize the objective function $F(\boldsymbol{x})$ restricted on the support set of $\boldsymbol{x}^t$. This gives the HTP algorithm as follows:

$$\begin{aligned} &\text{(HTP1)} \quad \boldsymbol{b}^t \leftarrow \boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}), \\ &\text{(HTP2)} \quad S^t \leftarrow \operatorname{supp}\left(\boldsymbol{b}^t, k\right), \\ &\text{(HTP3)} \quad \boldsymbol{x}^t \leftarrow \underset{\operatorname{supp}(\boldsymbol{x}) \subset S^t}{\arg\min} F(\boldsymbol{x}), \end{aligned}$$

where we recall that $\operatorname{supp}\left(\boldsymbol{b}^t, k\right)$ denotes the support set of the top k elements of $\boldsymbol{b}^t$. Below are two useful properties of the algorithm:

- each iterate $\boldsymbol{x}^t$ is k-sparse;
- it terminates when $S^t = S^{t-1}$ at some stage t.

The sparsity of the iterate inherently controls the statistical rate of the problem, hence offering a near-optimal sample complexity. We will discuss it in more detail in the theoretical analysis. The termination condition was utilized in previous work [60, 158] to establish iteration complexity.

In addition to the above algorithm proposed originally in [60], we will also consider a more realistic scenario. That is, for all $t \geq 1$, the iterate $\boldsymbol{x}^t$ in (HTP3) is subject to

$$F(\boldsymbol{x}^t) - F(\boldsymbol{x}_*^t) \leq \epsilon, \tag{2.4}$$

where $\epsilon > 0$ is a pre-defined accuracy parameter and $\boldsymbol{x}_*^t$ is the global minimizer of $F(\boldsymbol{x})$ restricted on S^t. The reason for our consideration is that when solving a general machine learning problem, it is typically expensive or impossible to obtain the exact minimizer $\boldsymbol{x}_*^t$. This is not an issue in compressed sensing where $F(\boldsymbol{x})$ is the least-squares loss, but is surely a concern for other problems such as logistic regression. Related to the inexact solutions, a natural question to ask is how the accuracy parameter ϵ affects the recovery performance of HTP, additively or progressively. Another issue coming up with the inexact iterates is that the usually employed stopping criterion $S^t = S^{t-1}$ may not be applicable, which makes part of the analysis in [158] invalid. Note that when exact solutions are available, HTP becomes stationary as soon as the detected support does not change, since the solutions are entirely determined by the support. Allowing approximate iterates quickly changes the situation and many stochastic solvers, e.g., stochastic gradient descent, introduce randomness, rendering (HTP3) outputs different results even restricted on the same support.

2.2.1 Preliminary Results: Parameter Estimation

Our analysis depends on the following two properties of the function $F(\boldsymbol{x})$.

Definition 2.1 (Restricted Strong Convexity)**.** A differentiable function $F(\boldsymbol{x})$ is said to satisfy the property of restricted strong convexity (RSC) at the sparsity level $K > 0$ with parameter $\rho_K^- > 0$,

if for all vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ with $\|\boldsymbol{x}-\boldsymbol{x}'\|_0 \leq K$,

$$F(\boldsymbol{x}) - F(\boldsymbol{x}') - \left\langle \nabla F(\boldsymbol{x}'), \boldsymbol{x}-\boldsymbol{x}' \right\rangle \geq \frac{\rho_K^-}{2} \left\|\boldsymbol{x}-\boldsymbol{x}'\right\|^2.$$

Definition 2.2 (Restricted Smoothness)**.** A differentiable function $F(\boldsymbol{x})$ is said to satisfy the property of restricted smoothness (RSS) at the sparsity level $K > 0$ with parameter $\rho_K^+ > 0$, if for all vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ with $\|\boldsymbol{x}-\boldsymbol{x}'\|_0 \leq K$,

$$F(\boldsymbol{x}) - F(\boldsymbol{x}') - \left\langle \nabla F(\boldsymbol{x}'), \boldsymbol{x}-\boldsymbol{x}' \right\rangle \leq \frac{\rho_K^+}{2} \left\|\boldsymbol{x}-\boldsymbol{x}'\right\|^2.$$

In particular, we require that the RSC condition holds at the sparsity level $2k+s$ and the RSS condition holds at the sparsity level $3k$. That is, we assume

$(A1)$ $F(\boldsymbol{x})$ satisfies the RSC property with parameter ρ_{2k+s}^-;

$(A2)$ $F(\boldsymbol{x})$ satisfies the RSS property with parameter ρ_{3k}^+.

Note that the RSC and RSS conditions are now standard and are widely utilized for establishing performance guarantees for a variety of popular algorithms, see, for example, [105, 2, 96]. For brevity, throughout the chapter we write $\rho^- := \rho_{2k+s}^-$ and $\rho^+ := \rho_{3k}^+$. We also denote $\kappa := \rho^+/\rho^-$ which is called the condition number of the problem.

By examining these conditions for the compressed sensing problem (2.1), we may have some high-level intuition why the assumptions are vital. By algebra, RSC essentially requires that

$$(\boldsymbol{x}-\boldsymbol{x}')^\top \nabla^2 F(\boldsymbol{x}')(\boldsymbol{x}-\boldsymbol{x}') \geq \rho^- \left\|\boldsymbol{x}-\boldsymbol{x}'\right\|^2.$$

Now substituting $F(\boldsymbol{x})$ with $\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2$ immediately gives

$$\left\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{A}\boldsymbol{x}'\right\|^2 \geq \rho^- \left\|\boldsymbol{x}-\boldsymbol{x}'\right\|^2,$$

or equivalently

$$\left\|\boldsymbol{y}-\boldsymbol{y}'\right\|^2 \geq \rho^- \left\|\boldsymbol{x}-\boldsymbol{x}'\right\|^2.$$

This implies that for different input, we always have different output. Therefore, RSC guarantees

that the model is always identifiable. It is worth mentioning that the strong convexity is assumed in a restricted sense, namely only for sparse directions. This is a significant feature since in the high-dimensional regime $n \ll d$, the Hessian matrix of $F(\boldsymbol{x})$ is highly degenerate, and strong convexity does not hold everywhere.

The RSS property enter our analysis mainly through the condition number κ. As will be clear, this quantity reflects the correlation among the attributes, and a large value indicates a difficult instance, which may require more computational resource and samples for the success of estimation.

Our first result states that if (HTP3) outputs exact solutions, then HTP decreases the function value with a geometric rate before the stopping criterion (i.e., $S^t = S^{t-1}$) is met. Formally, we have the following proposition.

Proposition 2.1. *Consider the HTP algorithm with exact solutions in (HTP3). Assume* $(A1)$ *and* $(A2)$*, pick* $\eta < 1/\rho^+$ *in (HTP1) and set* $k = s$ *in (HTP2). Then before HTP terminates, it holds that for all* $t \geq 1$*,*

$$F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) \leq \mu \left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}}) \right),$$

where

$$\mu = 1 - \frac{2\eta\rho^-(1-\eta\rho^+)}{1+s} \in (0,1).$$

Note that we did not assume the optimality of $\bar{\boldsymbol{x}}$ with respect to the function $F(\boldsymbol{x})$. In other words, Proposition 2.1 holds even for $F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) < 0$. It is also worth mentioning that by the proposition, we can deduce

$$F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) \leq \mu^t \left(F(\boldsymbol{x}^0) - F(\bar{\boldsymbol{x}}) \right).$$

However, the above inequality does not imply the convergence of $\{F(\boldsymbol{x}^t)\}_{t\geq 1}$, since $F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}})$ is not bounded from below. Rather, it is invoked to establish parameter estimation for HTP.

The following proposition shows that when the conditions in Proposition 2.1 are satisfied, we have an accurate estimate on the signal in the ℓ_2-metric.

Proposition 2.2. *Assume the same conditions as in Proposition 2.1. Then before HTP terminates, the following holds for* $t \geq 1$*:*

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \sqrt{2\kappa}(\sqrt{\mu})^t \left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{3}{\rho^-} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\|,$$

where μ *is given in Proposition 2.1.*

In the literature, a variety of work has established theoretical guarantees on parameter estimation, either under the RIP condition [24] or by relaxing the sparsity [158]. In contrast, neither of the conditions are assumed in Proposition 2.2, owing to a careful analysis on the connection between $\nabla F(\boldsymbol{x}^t)$ and $\bar{\boldsymbol{x}}$. See Section 2.B for the proof. However, we point out that such an appealing behavior is not guaranteed if (HTP3) does not output exact solutions, and in this case, we have to relax the sparsity or use the RIP condition. In particular, suppose that

$$\boldsymbol{x}_*^t = \underset{\mathrm{supp}(\boldsymbol{x}) \subset S^t}{\arg\min} F(\boldsymbol{x}),$$

and (HTP3) outputs $\boldsymbol{x}^t$ that obeys

$$\mathrm{supp}\left(\boldsymbol{x}^t\right) \subset S^t,\ F(\boldsymbol{x}^t) - F(\boldsymbol{x}_*^t) \leq \epsilon. \tag{2.5}$$

Note that this is a realistic scenario because even for simple functions, e.g., $F(\boldsymbol{x})$ is the logistic loss, convex solvers only ensure ϵ-approximate solutions. The major issue coming up with the ϵ-approximate solutions is that the gradient of $F(\boldsymbol{x})$ evaluated at $\boldsymbol{x}^t$ does not vanish on the support S^t, which makes the analysis of Proposition 2.2 invalid. Yet, we can still bound it under proper conditions.

Lemma 2.3. *Assume* $(A2)$ *and* (2.5)*. Then at any iteration* $t \geq 1$*, we have*

$$\left\|\nabla_{S^t} F(\boldsymbol{x}^t)\right\| \leq \sqrt{2\rho^+ \epsilon}.$$

Based on the lemma, we show the following RIP-based result for parameter estimation.

Proposition 2.4. *Consider the HTP algorithm with inexact solutions* (2.5)*. Suppose that the condition number* $\kappa < 1.25$ *and set* $k = s$ *in (HTP2). Then picking* $\eta = \eta'/\rho^+$ *with* $\kappa - 0.25 < \eta' < 1$

guarantees

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq (\sqrt{2}(\kappa - \eta'))^t \left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{6\kappa}{\rho^-} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\| + \frac{4\sqrt{\rho^+ \epsilon}}{\rho^-}.$$

As the RIP condition is hard to fulfill for many machine learning problems, [75] proposed to relax the sparsity parameter $k = \mathcal{O}\left(\kappa^2 s\right)$ in order to alleviate it. [131] further showed that by relaxing the sparsity, a stochastic solver is able to produce an accurate solution for sparsity-constrained programs. Inspired by their interesting work, we derive the following result for HTP.

Proposition 2.5. *Consider the HTP algorithm with inexact solutions* (2.5)*. Pick* $\eta < 1/\rho^+$ *and let* $k \geq 2s + \frac{8s}{(\eta\rho^-)^2}$*. Then*

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \sqrt{2\kappa}(\sqrt{\mu})^t \left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{3}{\rho^-} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\| + \sqrt{\frac{4\epsilon}{\rho^-(1-\mu)}},$$

where

$$\mu = 1 - \frac{\eta\rho^-(1 - \eta\rho^+)}{2}.$$

Weak Support Recovery

Propositions 2.4 and 2.5 offer useful results for parameter estimation. Namely it is guaranteed that

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \alpha \cdot \beta^t + \gamma,$$

where the detailed values of α, β and γ are given in the propositions. On the other hand, we have the following well-known result that relates parameter estimation to support recovery.

Lemma 2.6. *If* $\|\boldsymbol{x} - \bar{\boldsymbol{x}}\| < \bar{\boldsymbol{x}}_{\min}$*, then* $\operatorname{supp}(\boldsymbol{x}) = \operatorname{supp}(\bar{\boldsymbol{x}})$.

The proof follows from algebra and the definition of $\bar{\boldsymbol{x}}_{\min}$. The lemma was broadly used to show support recovery [146], and in principle we can impose

$$\alpha \cdot \beta^t < \frac{\bar{\boldsymbol{x}}_{\min}}{2}, \quad \gamma < \frac{\bar{\boldsymbol{x}}_{\min}}{2},$$

which results in $\operatorname{supp}\left(\boldsymbol{x}^t\right) = \operatorname{supp}\left(\bar{\boldsymbol{x}}\right)$. However, the primary issue is that the iteration complexity is $\mathcal{O}\left(\log(1/\bar{\boldsymbol{x}}_{\min})\right)$, where the quantity $\bar{\boldsymbol{x}}_{\min}$ is entirely unknown. For practitioners, such a bound does not guide the execution of the algorithm.

2.2.2 Main Results: Strong Support Recovery

This section is dedicated to a deterministic analysis on the performance of support recovery. In particular, we focus on the iteration complexity and show that it does not depend on the unknown quantity $\bar{\boldsymbol{x}}_{\min}$. We first treat the exact case, i.e., (HTP3) outputs exact solutions, along with a detailed comparison with previous work in the literature. Then we demonstrate that even (HTP3) is solved approximately, support recovery is still possible provided that ϵ is small enough compared to the magnitude of the target signal.

The following theorem is one of the main results in the section. It justifies that under proper conditions, HTP recovers the support of $\bar{\boldsymbol{x}}$ using finite iterations.

Theorem 2.7. *Consider the HTP algorithm with exact solutions in (HTP3). Assume* $(A1)$ *and* $(A2)$. *Pick* $\eta < 1/\rho^+$ *in (HTP1) and* $k = s$ *in (HTP2). Then HTP either terminates early, or recovers the support of* $\bar{\boldsymbol{x}}$ *using at most*

$$t_{\max} = \left(\frac{3\log\kappa}{\log(1/\mu)} + \frac{2\log(2/(1-\lambda))}{\log(1/\mu)} + 2 \right) \|\bar{\boldsymbol{x}}\|_0 \tag{2.6}$$

iterations, provided that

$$\bar{\boldsymbol{x}}_{\min} \geq \frac{2\sqrt{2}+\sqrt{\kappa}}{\rho^- \lambda} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\| \tag{2.7}$$

for some constant $\lambda \in (0,1)$. *Above, the quantity* μ *is given by*

$$\mu = 1 - \frac{2\rho^-\eta(1-\eta\rho^+)}{1+s} \in (0,1).$$

Below we discuss the important messages conveyed by the theorem and contrast our result with prior work. For ease of exposition, we write $\eta = \eta'/\rho^+$ for some constant $\eta' \in (0,1)$, and it quickly indicates that $\mu = 1 - \Theta(1/\kappa)$.

Iteration Complexity

We remind that the first term in (2.6) plays the most crucial role, since it dominates the other two for sufficiently large κ. In the regime where κ itself is bounded by a constant from above, the iteration complexity is simply explained as $\mathcal{O}\left(\|\bar{\boldsymbol{x}}\|_0\right)$. Asymptotically, we can show that the iteration complexity is dominated by $\kappa \log \kappa$ as κ tends to infinity, that is,

$$t_{\max} = \mathcal{O}\left(\|\bar{\boldsymbol{x}}\|_0 \kappa \log \kappa\right).$$

This follows from a simple calculation on the Taylor expansion of $\log(1/\mu)$ at the point $x = 1$, with μ being replaced with $1 - \Theta(1/\kappa)$. Note that the number of iterations we obtained for support recovery is as few as that for accurate parameter estimation (see Proposition 2.2). It is also worth mentioning that the linear dependency on the sparsity of $\bar{\boldsymbol{x}}$ is near-optimal, because in the worst case HTP may take several steps to pick only one correct support.

Conditions

We also emphasize that the condition (2.7) is now ubiquitous for analyzing the support recovery performance. The quantity $\bar{\boldsymbol{x}}_{\min}$ involved is natural, because a signal with large magnitude is easier to recover than those with small or vanishing components. To see why $\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\|$ is used to lower bound the magnitude of $\bar{\boldsymbol{x}}$, let us consider the compressed sensing problem (2.1) as an example. In order to recover the true parameter $\bar{\boldsymbol{x}}$, we may choose $F(\boldsymbol{x})$ as the least-squares, of which the derivative evaluated at $\boldsymbol{x} = \bar{\boldsymbol{x}}$ is given by

$$\nabla F(\bar{\boldsymbol{x}}) = \boldsymbol{A}^\top \left(\boldsymbol{A}\bar{\boldsymbol{x}} - \boldsymbol{y}\right) = -\boldsymbol{A}^\top \boldsymbol{e}.$$

Hence,

$$\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\| = \left\| \boldsymbol{A}_\Omega^\top \boldsymbol{e} \right\|,$$

where $\boldsymbol{A}_\Omega$ is a submatrix of $\boldsymbol{A}$ with columns indexed by Ω. Then the RSC and RSS conditions assert that

$$\sqrt{\rho^-}\,\|\boldsymbol{e}\| \le \|\nabla_{k+s} F(\bar{\boldsymbol{x}})\| \le \sqrt{\rho^+}\,\|\boldsymbol{e}\|.$$

Table 2.1: **Comparison to previous work on HTP-style algorithms.**

	Target sparse signal	RIP-free	No sparsity relaxation	Support recovery
[60]	true signal	✗	✓	✗
[159]	arbitrary	✗	✗	✗
[75]	optimal solution	✓	✗	✗
[24]	true signal	✗	✓	✓
[158, Theorem 1]	optimal solution	✗	✓	✓
[158, Theorem 3]	arbitrary	✓	✗	✓
Proposed Theorem 2.7	arbitrary	✓	✓	✓

Therefore, imposing the condition (2.7) amounts to distinguishing the true signal from the noise.

Comparison to Prior Work

We contrast our result to the state-of-the-art work [158]. To recover a sparse signal $\bar{\boldsymbol{x}}$, [158] required the condition number $\kappa < 1.14$, which is not applicable to general machine learning problems. In addition, support recovery was established only for a carefully chosen $F(\boldsymbol{x})$, i.e., $\bar{\boldsymbol{x}}$ must be the unique global minimizer of $F(\boldsymbol{x})$ subject to a sparsity constraint (see Theorem 1 therein). Such a requirement dramatically excludes many popular and simple choices of $F(\boldsymbol{x})$. For example, let us again examine the compressed sensing problem. With the presence of noise, it is almost impossible for $\bar{\boldsymbol{x}}$ to be the global optimum of $F(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$. Hence, one cannot apply the theoretical result of [158] to justify the performance of HTP. In comparison, our theorem ensures that support recovery is possible as far as the selected $F(\boldsymbol{x})$ fulfills the condition (2.7). Though Theorem 3 in [158] does not assume the RIP condition or the optimality of $\bar{\boldsymbol{x}}$ with respect to $F(\boldsymbol{x})$, it requires a relaxed sparsity parameter $k = \mathcal{O}\left(\kappa^2 s\right)$, whereas Theorem 2.7 asserts that $k = s$ suffices. We also note that the iteration complexity was not provided by [158] in the relaxed sparsity case, whereas we clearly state the dependency on all the parameters.

Compared with [24], it is not hard to see that the problem considered here is more general, since we aim to recover an arbitrary sparse signal while they targeted the true parameter of compressed sensing. [24] also imposed the RIP condition that is not invoked here. [73, 75] presented interesting HTP-style algorithms with analysis on parameter estimation, but a guarantee on support recovery was not considered. We summarize the comparison in Table 2.1.

Weakness

We remark that though Theorem 2.7 is free of the RIP condition and the relaxed sparsity, it implicitly requires that HTP should not terminate too early. Otherwise, HTP may fail to recover the support. We believe that it is a very interesting future direction to give a lower bound on the iteration complexity of HTP. In the sequel, we strengthen our result by providing sufficient conditions which prevents HTP from early stopping.

Improvements

We move on to the practical scenario where the results to be established also apply to the exact case. As a reminder, due to the assumption $(A1)$, (HTP3) is virtually solving a convex program. Yet, since $F(\boldsymbol{x})$ is a general function, (HTP3) can only be solved approximately by, e.g., gradient descent [107], stochastic gradient descent [23], or the more recent variance reduced variant [78]. An interesting question to ask is, whether support recovery is possible under such a "noisy" setting, and how the optimization error ϵ enters the conditions for this end.

The following theorem presents an affirmative answer, though the RIP condition is assumed.

Theorem 2.8. *Consider the HTP algorithm with ϵ-approximate solutions in (HTP3). Assume $(A1)$ and $(A2)$. Suppose that the condition number $\kappa < 1.25$. Pick $\eta = \eta'/\rho^+$ with $\kappa - 0.25 < \eta' < 1$ and set $k = s$ in (HTP2). Then HTP recovers the support of $\bar{\boldsymbol{x}}$ using at most*

$$t_{\max} = \left(\frac{\log \kappa}{\log(1/\mu)} + \frac{\log(\sqrt{2}/(1-\lambda))}{\log(1/\mu)} + 2 \right) \|\bar{\boldsymbol{x}}\|_0 \tag{2.8}$$

iterations, provided that

$$\bar{\boldsymbol{x}}_{\min} \geq \frac{\sqrt{2} + 3\sqrt{2}\kappa}{\rho^- \lambda} \|\nabla_{k+s} F(\bar{\boldsymbol{x}})\| + \frac{4}{\rho^- \lambda} \sqrt{\rho^+ \epsilon} \tag{2.9}$$

for some constant $\lambda \in (0, 1)$. Above, the quantity μ is given by

$$\mu = \sqrt{2}(\kappa - \eta') \in (0, \sqrt{2}/4).$$

Since the condition number is assumed to be well bounded, it follows that the iteration com-

plexity is a constant multiple of the sparsity, i.e., $\mathcal{O}\left(\|\bar{\boldsymbol{x}}\|_0\right)$. By examining the $\bar{\boldsymbol{x}}_{\min}$-condition (2.9), we find that the optimization error ϵ does not propagate in a progressive manner. Rather, it enters the condition as an additive error. By comparing (2.9) to (2.7), the exact case, one may argue that (2.9) is more stringent because it requires $\bar{\boldsymbol{x}}_{\min} \geq \kappa \|\nabla_{k+s} F(\bar{\boldsymbol{x}})\|$ while (2.7) imposes $\bar{\boldsymbol{x}}_{\min} \geq \sqrt{\kappa} \|\nabla_{k+s} F(\bar{\boldsymbol{x}})\|$. Yet, we point out that Theorem 2.8 is based on the RIP condition, i.e., $\kappa < 1.25$. So it is not appropriate to examine the asymptotic behavior for the condition (2.9).

Finally, we study under which RIP-free conditions can HTP guarantee support recovery in the face of approximate solutions. We have the following result.

Theorem 2.9. *Consider the HTP algorithm with ϵ-approximate solutions in (HTP3). Assume $(A1)$ and $(A2)$. Pick $\eta < 1/\rho^+$ and let $k \geq 2s + \frac{8s}{(\eta\rho^-)^2}$ in (HTP2). Then HTP recovers the support of $\bar{\boldsymbol{x}}$ using at most*

$$t_{\max} = \left(\frac{3 \log \kappa}{\log(1/\mu)} + \frac{4 \log(\sqrt{2}/(1-\lambda))}{\log(1/\mu)} + 2 \right) \|\bar{\boldsymbol{x}}\|_0 \tag{2.10}$$

iterations, provided that

$$\bar{\boldsymbol{x}}_{\min} \geq \frac{2\sqrt{2} + \sqrt{\kappa}}{\rho^- \lambda} \|\nabla_{k+s} F(\bar{\boldsymbol{x}})\| + \lambda^{-1} \left(\sqrt{\frac{2}{\rho^-(1-\mu)}} + \sqrt{\frac{2}{\rho^-}} \kappa \right) \sqrt{\epsilon} \tag{2.11}$$

for some constant $\lambda \in (0, 1)$. Above, the quantity μ is given by

$$\mu = 1 - \frac{\eta\rho^-(1 - \eta\rho^+)}{2} \in (0, 1).$$

To be clear, due to sparsity relaxation, Theorem 2.9 only ensures support inclusion, i.e., $S \subset S^{t_{\max}}$. In [158], they showed that under the condition

$$\bar{\boldsymbol{x}}_{\min} > 1.62 \sqrt{\frac{2(F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}_{\text{opt}}))}{\rho^-}},$$

HTP terminates with output $\boldsymbol{x}^t$ satisfying $\operatorname{supp}\left(\boldsymbol{x}^t, s\right) = S$. However, the iteration number t was not given. Either, it is not clear how large the difference $F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}_{\text{opt}})$ is, where $\boldsymbol{x}_{\text{opt}}$ is a global s-sparse minimizer of $F(\boldsymbol{x})$ and we recall that $\bar{\boldsymbol{x}}$ is an arbitrary signal.

In contrast to Theorem 2.8, the quantity $\sqrt{\epsilon}$ here is multiplied by the condition number κ,

which will consume more computational resources in order to fulfill the condition. This is not surprising because enlarging the support increases the chance of detecting the support but as a price, it also introduces more noise. Fortunately, under the RSC and RSS assumptions, first order solvers converges linearly. For instance, after $\mathcal{O}\left(\kappa \log(1/\epsilon)\right)$ steps, gradient descent guarantees an ϵ-approximate solution.

In view of the existing study on convex optimization (see, for example, [107]), together with Theorem 2.9, we can show that the total computational complexity of HTP is

$$\left(d + \kappa^2 s \log d + \kappa^3 s \log(1/\epsilon)\right) s\kappa \log \kappa. \tag{2.12}$$

To see this, note that (HTP1) consumes $\mathcal{O}\left(d\right)$ operations and (HTP2) costs $\mathcal{O}\left(k \log d\right)$. Using gradient descent to solve (HTP3) results in a complexity $\mathcal{O}\left(k\kappa \log(1/\epsilon)\right)$. Combining them together, we obtain the above.

We point out that though Theorem 2.7 and Theorem 2.8 need to know the sparsity s, one can set k to be a quantity smaller than s. In this case, our analysis shows that HTP recovers the support of the top-k elements. In realistic applications, usually the parameter k is tuned by cross-validation though.

2.2.3 Statistical Results

In this section, we relate our main results, Theorem 2.7 to Theorem 2.9, to concrete statistical models. In particular, we study two prevalent models: the sparse linear regression and the sparse logistic regression. Notably, it is known that similar statistical results can be built for low-rank matrix regression, sparse precision matrix estimation, as suggested in [105, 2].

Sparse Linear Regression

For sparse linear regression, the observation model is given by

$$\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{e}, \quad \|\bar{\boldsymbol{x}}\|_0 \leq s, \tag{2.13}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ is the design matrix, $\boldsymbol{y} \in \mathbb{R}^n$ is the response, $\boldsymbol{e} \in \mathbb{R}^n$ is some noise, and $\bar{\boldsymbol{x}}$ is the s-sparse true parameter we hope to estimate from the knowledge of $\boldsymbol{A}$ and $\boldsymbol{y}$. Note that when we have the additional constraint $n \ll d$, the model above is exactly that of compressed sensing (2.1).

In order to (approximately) estimate the parameter, a natural approach is to optimize the following non-convex program:

$$\min_{\boldsymbol{x}} \; F(\boldsymbol{x}) := \frac{1}{2n}\sum_{i=1}^{n} \|y_i - \boldsymbol{a}_i \cdot \boldsymbol{x}\|^2, \quad \text{s. t. } \|\boldsymbol{x}\|_0 \le s. \tag{2.14}$$

For our analysis, we assume the following on the design matrix and the noise:

$(A3)$ $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n$ are independent and identically distributed (i.i.d.) Gaussian random vectors $N(\boldsymbol{0}, \boldsymbol{\Sigma})$. All the diagonal elements of $\boldsymbol{\Sigma}$ satisfy $\Sigma_{jj} \le 1$. The noise $\boldsymbol{e}$ is independent of $\boldsymbol{A}$ and its entries are i.i.d. Gaussian random variables $N(0, \sigma^2)$.

Lemma 2.10. *Consider the sparse linear regression model* (2.13) *and the program* (2.14). *Assume* $(A3)$. *Then for a sparsity level* K,

- *with probability at least* $1 - \exp(-\mathrm{C}_0 n)$,

$$\rho_K^- = \lambda_{\min}(\boldsymbol{\Sigma}) - \mathrm{C}_1\frac{K\log d}{n}, \quad \rho_K^+ = \lambda_{\max}(\boldsymbol{\Sigma}) + \mathrm{C}_2\frac{K\log d}{n};$$

- *with probability at least* $1 - \mathrm{C}_3/d$

$$\|\nabla_K F(\bar{\boldsymbol{x}})\| \le \mathrm{C}_4\sigma\sqrt{\frac{K\log d}{n}}.$$

Above, $\lambda_{\min}(\boldsymbol{\Sigma})$ *and* $\lambda_{\max}(\boldsymbol{\Sigma})$ *are the minimum and maximum singular values of* $\boldsymbol{\Sigma}$ *respectively.*

This is a standard result in the literature, and its proof can be found in [119, 125]. Note that for Theorem 2.7, as far as $n \ge 6\mathrm{C}_1(s\log d)/\lambda_{\min}(\boldsymbol{\Sigma})$, the quantity ρ_{2k+s}^- is always positive, meaning that $(A1)$ is satisfied with high probability. This is also true for $(A2)$. For the $\bar{\boldsymbol{x}}_{\min}$-condition therein, with a calculation we find that it is met with high probability provided that n is large enough.

Sparse Logistic Regression

For sparse logistic regression, the observation model is given by

$$\mathbb{P}(y_i \mid \boldsymbol{a}_i;\ \bar{\boldsymbol{x}}) = \frac{1}{1+\exp(-y_i \boldsymbol{a}_i^\top \bar{\boldsymbol{x}})}, \quad \|\bar{\boldsymbol{x}}\|_0 \le K, \quad \forall\, 1 \le i \le n, \tag{2.15}$$

where y_i is either 0 or 1. It then learns the parameter by minimizing the negative log-likelihood:

$$\min_{\boldsymbol{x}}\ F(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} \log\left(1+\exp(-y_i \boldsymbol{a}_i^\top \boldsymbol{x})\right), \quad \text{s.t. } \|\boldsymbol{x}\|_0 \le K,\ \|\boldsymbol{x}\| \le \omega. \tag{2.16}$$

There is a large body of work showing that the statistical property is rather analogous to that of linear regression. See, for example, [105, 159]. In fact, the statistical results apply to generalized linear models as well.

2.2.4 Experiments

The HTP algorithm has been studied for several years and has found plenty of successful applications. There is also a large volume of empirical study, e.g., [24], showing that HTP performs better in terms of computational efficiency and parameter estimation than compressive sampling matching pursuit [103], subspace pursuit [45], iterative hard thresholding [20], to name a few. Hence, the focus of our numerical study is to verify the theoretical findings.

The experimental settings are as follows:

- **Data.** In order to investigate the performance of HTP with both the exact and inexact solutions, we consider the linear regression model $\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}} + \sigma \boldsymbol{e}$, where $\bar{\boldsymbol{x}}$ is a 100-dimensional vector with a tunable sparsity s. The elements in the design matrix $\boldsymbol{A}$ and the noise $\boldsymbol{e}$ are i.i.d. normal variables. The response $\boldsymbol{y}$ is an N-dimensional vector. For a certain sparsity level s, the support of $\bar{\boldsymbol{x}}$ is chosen uniformly and the non-zero components of $\bar{\boldsymbol{x}}$ are i.i.d. normal variables. If not specified, we set the sample size $n = 100$ and the noise level $\sigma = 0.01$.

- **Evaluation metric.** In the experiments, we are mainly interested in examining the percentage of successful support recovery and the iteration number that guarantees it. We mark a trial as success if before HTP terminates, there is a solution $\boldsymbol{x}^t$ satisfying $\operatorname{supp}\left(\boldsymbol{x}^t\right) = \operatorname{supp}\left(\bar{\boldsymbol{x}}\right)$.

Otherwise, we mark it as failure. The iteration number is counted only for those success trials and we report the averaged result.

- **Solvers.** We choose the least-squares loss as the proxy function $F(\boldsymbol{x})$, for which an exact solution can be computed in (HTP3). We also implement the gradient descent (GD) algorithm to approximately solve (HTP3). In order to produce solutions with different optimization error ϵ, we run the GD algorithm with a various number of gradient oracle calls. In this way, we are able to examine how ϵ affects support recovery through the number of oracle calls.

- **Other settings.** The step size η in HTP is fixed as $\eta = 1$. We use the true sparsity for the sparsity parameter k in (HTP2). For each configuration of sparsity, we generate 100 independent copies of $\bar{\boldsymbol{x}}$. Hence, all the experiments are performed with 100 trials.

A notable aspect of our theoretical results is that after $\mathcal{O}\left(s\kappa \log \kappa\right)$ iterations, HTP captures the support. For the purpose of justification, we vary the sparsity s from 1 to 50, and plot the curve of the iteration number used to identify the support against the true sparsity s. Note that we use the same design matrix for all trials, hence a fixed condition number κ. The result is recorded in the left panel of Figure 2.1. As predicted by our theorem, the iteration number is (almost) linear with the sparsity. Interestingly, we also find that HTP uses far fewer steps than expected. For example, to recover the support of a 20-sparse signal, 4 iterations suffice in average, suggesting possible improvement of our theorems in special cases. Also note that for a given sparsity level, applying an inexact solver for (HTP3) does not increase the iteration number of HTP. This is not surprising since our theorem states that the optimization error in (HTP3) only enters the $\bar{\boldsymbol{x}}_{\min}$-condition. In other words, it only affects the percentage of success as shown in the right panel of Figure 2.1. Thanks to the linear convergence of gradient descent, it turns out that using 50 calls of gradient oracle guarantees a comparable performance with the exact solution.

Next, we tune the number of measurements n from 1 to 100, and study the support recovery performance against the choice of n. Here, the sparsity level s is fixed to $s = 5$. With the sub-gaussian design, we have shown that the condition number can be upper bounded by $(\mathrm{C}_1 n + s \log d)/(\mathrm{C}_2 n - s \log d)$. This indicates that the condition number is inversely proportional to N after a proper shifting, and hence the iteration number. The curves on the left panel of Figure 2.2 matches our assertion. In the right panel, a phase transition emerges [53]. That is, above

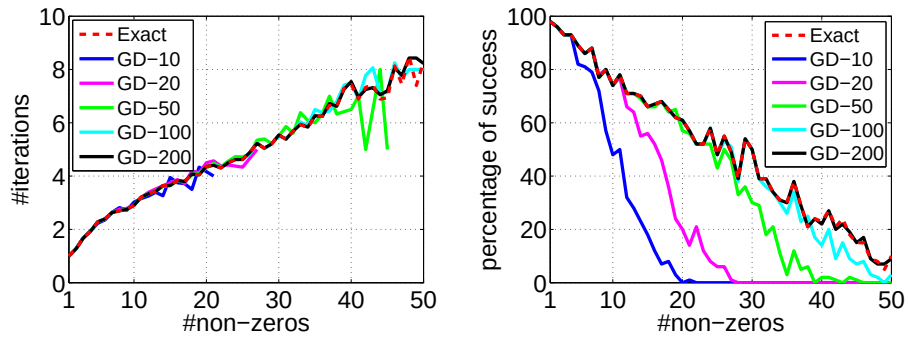


Figure 2.1: **HTP: The iteration number and percentage of success against the sparsity.**

a certain threshold (here the threshold is 20), support recovery is guaranteed with high probability while below that threshold, we have no hope to estimate the signal. We also find that when sufficient measurements are available, running GD with 10 gradient oracle calls already brings desirable performance.

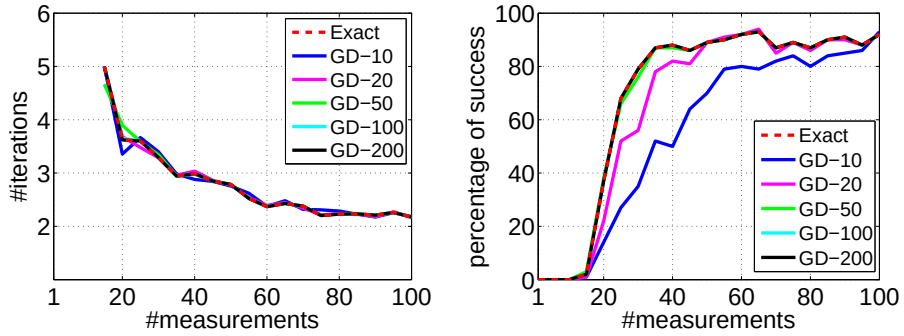


Figure 2.2: **HTP: The iteration number and percentage of success against the number of measurements.**

We remind that in Figure 2.1 and Figure 2.2, some values of #iterations are not plotted. For example, we do not have the iteration number for GD–50 in Figure 2.1 when $s \geq 45$. This is simply because all the trials are marked as failure. See the associated percentage of success curve.

Now let us return to the $\bar{\boldsymbol{x}}_{\min}$-condition of Theorem 2.9, i.e., Eq. (2.11). From Figure 2.1 and Figure 2.2, we conclude that as far as the optimization error is small enough, HTP with inexact iterates behaves comparably to that with exact solutions. For example, the "GD–200" curve (black solid) and the "Exact" curve (red dashed) in these two figures actually lie on top of each other even the RIP condition is not met (small n or large s).

2.3 Towards A Principled Analysis of Support Recovery

We have shown that HTP enjoys iteration complexity proportional to the sparsity of the underlying signal. This section generalizes the hard thresholding operator to partial hard thresholding (PHT) [74]. In this way, we are able to study the computational and statistical trade-off among the family of the algorithms using the PHT operator.

Formally, given a support set T and a freedom parameter $r > 0$, the PHT operator which is used to produce a k-sparse approximation to $\boldsymbol{b}$ is defined as follows:

$$\mathrm{PHT}_k\left(\boldsymbol{b}; T, r\right) := \underset{\boldsymbol{x}\in\mathbb{R}^d}{\arg\min} \left\|\boldsymbol{x}-\boldsymbol{b}\right\|, \ \mathrm{s.t.}\ \left\|\boldsymbol{x}\right\|_0 \leq k, \ \left|T \backslash \mathrm{supp}\left(\boldsymbol{x}\right)\right| \leq r. \tag{2.17}$$

The first constraint simply enforces a k-sparse solution. To gain intuition on the second one, consider that T is the support set of the last iterate of an iterative algorithm, for which $|T| \leq k$. Then the second constraint ensures that the new support set differs from the previous one by at most r positions. As a special case, one may have noticed that the PHT operator reduces to the standard hard thresholding when picking the freedom parameter $r \geq k$. On the other spectrum, if we look at the case where $r = 1$, the PHT operator yields the interesting algorithm termed orthogonal matching pursuit with replacement [73], which in general replaces one element in each iteration.

It has been shown in [74] that the PHT operator can be computed in an efficient manner for a general support set T and a freedom parameter r. In this section, our major focus will be on the case $|T| = k$. Then Lemma 1 of [74] indicates that $\mathrm{PHT}_k\left(\boldsymbol{b}; T, r\right)$ is given as follows:

$$\mathrm{top} = \mathrm{supp}\left(\boldsymbol{b}_{\overline{T}}, r\right), \ \mathrm{PHT}_k\left(\boldsymbol{b}; T, r\right) = \mathcal{H}_k\left(\boldsymbol{b}_{T\cup\mathrm{top}}\right), \tag{2.18}$$

where $\mathcal{H}_k\left(\cdot\right)$ is the standard hard thresholding operator that sets all but the k largest absolute components of a vector to zero.

Equipped with the PHT operator, we are now in the position to describe a general iterative greedy algorithm, termed PHT(r) where r is the freedom parameter in (2.17). At the t-th iteration, the algorithm reveals the last iterate $\boldsymbol{x}^{t-1}$ as well as its support set S^{t-1}, and returns a new solution

as follows:

$$\begin{aligned}
\boldsymbol{b}^t &= \boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}), \\
\boldsymbol{y}^t &= \mathrm{PHT}_k\left(\boldsymbol{b}^t; S^{t-1}, r\right),\ S^t = \mathrm{supp}\left(\boldsymbol{y}^t\right), \\
\boldsymbol{x}^t &= \underset{\boldsymbol{x}\in\mathbb{R}^d}{\arg\min}\, F(\boldsymbol{x}),\ \mathrm{s.t.}\ \mathrm{supp}\left(\boldsymbol{x}\right) \subset S^t.
\end{aligned}$$

Above, we note that $\eta > 0$ is a step size and $F(\boldsymbol{x})$ is a proxy function which should be carefully chosen (to be clarified later). Typically, the sparsity parameter k equals s, the sparsity of the target signal $\bar{\boldsymbol{x}}$. In this section, we again consider a more general choice of k which leads to novel results. One may have observed that in the context of sparsity-constrained minimization (2.2), the proxy function $F(\boldsymbol{x})$ used above is chosen as the objective function [159, 75]. In that scenario, the target signal is a global optimum and PHT(r) proceeds as projected gradient descent. Nevertheless, recall that our goal is to estimate an arbitrary signal $\bar{\boldsymbol{x}}$. It is not realistic to look for a function $F(\boldsymbol{x})$ such that our target happens to be its global minimizer. The remedy we will offer is characterizing a deterministic condition between $\bar{\boldsymbol{x}}$ and $\nabla F(\bar{\boldsymbol{x}})$ which is analogous to the signal-to-noise ratio condition, so that any function $F(\boldsymbol{x})$ fulfilling that condition suffices. In this light, we find that $F(\boldsymbol{x})$ behaves more like a proxy that guides the algorithm to a given target. Remarkably, our analysis also encompasses the situation considered in [159, 75].

2.3.1 Deterministic Analysis

The following proposition shows that under very mild conditions, PHT(r) either terminates or recovers the support of an arbitrary s-sparse signal $\bar{\boldsymbol{x}}$ in few iterations.

Proposition 2.11. *Consider the PHT(r) algorithm with $k = s$. Suppose that $F(\boldsymbol{x})$ is ρ_{2s}^--RSC and ρ_{2s}^+-RSS, and the step size $\eta \in (0, 1/\rho_{2s}^+)$. Let $\kappa := \rho_{2s}^+/\rho_{2s}^-$. Then PHT$(r)$ either terminates or recovers the support of $\bar{\boldsymbol{x}}$ within $O(s\kappa \log \kappa)$ iterations provided that $\bar{\boldsymbol{x}}_{\min} \geq \frac{4\sqrt{2}+2\sqrt{\kappa}}{\rho_{2s}^-} \left\|\nabla_{2s} F(\bar{\boldsymbol{x}})\right\|$.*

A few remarks are in order. First, we remind the reader that under the conditions stated above, it is *not* guaranteed that PHT(r) succeeds. We say that PHT(r) fails if it terminates at some time stamp t but $S^t \neq S$. This indeed happens if, for example, we feed it with a bad initial point and pick a very small step size. In particular, if $\boldsymbol{x}^0_{\min} > \eta \left\|\nabla F(\boldsymbol{x}^0)\right\|_\infty$, then the algorithm makes

no progress. The crux to remedy this issue is imposing a lower bound on η or looking at more coordinates in each iteration, which is the theme below. However, the proposition is still useful because it asserts that as far as we make sure that PHT(r) runs long enough (i.e., $\mathcal{O}\left(s\kappa \log \kappa\right)$ iterations), it recovers the support of an arbitrary sparse signal. We also note that neither the RIP condition nor a relaxed sparsity is assumed in this proposition. The $\bar{\boldsymbol{x}}_{\min}$-condition above is natural, as has been discussed in Section 2.2.

In the following, we strengthen Prop. 2.11 by considering the RIP condition which requires a well-bounded condition number.

Theorem 2.12. *Consider the PHT(r) algorithm with $k = s$. Suppose that $F(\boldsymbol{x})$ is ρ^{-}_{2s+r}-RSC and ρ^{+}_{2s+r}-RSS. Let $\kappa := \rho^{+}_{2s+r}/\rho^{-}_{2s+r}$ be the condition number which is smaller than $1 + 1/(\sqrt{2} + \nu)$ where $\nu = \sqrt{1 + s/r}$. Pick the step size $\eta = \eta'/\rho^{+}_{2s+r}$ such that $\kappa - \frac{1}{\sqrt{2}+\nu} < \eta' \leq 1$. Then PHT$(r)$ recovers the support of $\bar{\boldsymbol{x}}$ within*

$$t_{\max} = \left(\frac{\log \kappa}{\log(1/\beta)} + \frac{\log(\sqrt{2}/(1-\lambda))}{\log(1/\beta)} + 2 \right) \left\| \bar{\boldsymbol{x}} \right\|_0$$

iterations, provided that for some constant $\lambda \in (0, 1)$

$$\bar{\boldsymbol{x}}_{\min} \geq \frac{3\sqrt{s} + 6}{\lambda \rho^{-}_{2s+r}} \left\| \nabla_{s+r} F(\bar{\boldsymbol{x}}) \right\|.$$

Above, $\beta = (\sqrt{2} + \nu)(\kappa - \eta') \in (0, 1)$.

We remark several aspects of the theorem. The most important part is that Theorem 2.12 offers the theoretical justification that PHT(r) always recovers the support. This is achieved by imposing an RIP condition (i.e., bounding the condition number from the above) and using a proper step size.

We also make the iteration bound explicit, in order to examine the parameter dependency. First, we note that $t_{\max}$ scales approximately linearly with λ. This conforms the intuition because a small λ actually indicates a large signal-to-noise ratio, and hence easy to distinguish the support of interest from the noise. The freedom parameter r is mainly encoded in the coefficient β through the quantity ν. Observe that when increasing the scalar r, we have a small β, and hence fewer iterations. This is not surprising since a large value of r grants the algorithm more freedom to look at the current iterate. Indeed, in the best case, PHT(s) is able to recover the support in constant iterations while

PHT(1) has to take $\mathcal{O}(s)$ steps. However, if we investigate the $\bar{\boldsymbol{x}}_{\min}$-condition, we find that we need a stronger SNR condition to afford a large freedom parameter.

It is also interesting to contrast Theorem 2.12 to [158, 24], which independently built state-of-the-art support recovery results for HTP. As has been mentioned, [158] made use of the optimality of the target signal, which is a restricted setting compared to our result. Their iteration bound (see Theorem 1 therein), though provides an appealing insight, does not have a clear parameter dependence on the natural parameters of the problem (e.g., sparsity and condition number). [24] developed $\mathcal{O}(s)$ iteration complexity for compressed sensing. Again, they confined to a special signal whereas we carry out a generalization that allows us to analyze a family of algorithms.

Though the RIP condition has been ubiquitous in the literature, many researchers point out that it is not realistic in practical applications [17, 118, 125]. This is true for large-scale machine learning problems, where the condition number may grow with the sample size (hence one cannot upper bound it with a constant). A clever solution was first (to our knowledge) suggested by [75], where they showed that using the sparsity parameter $k = \mathcal{O}\left(\kappa^2 s\right)$ guarantees convergence of projected gradient descent. The idea was recently employed by [131, 158] to show an RIP-free condition for sparse recovery, though in a technically different way. The following theorem borrows this elegant idea to prove RIP-free results for PHT(r).

Theorem 2.13. *Consider the PHT(r) algorithm. Suppose that $F(\boldsymbol{x})$ is ρ_{2k}^--RSC and ρ_{2k}^+-RSS. Let $\kappa := \rho_{2k}^+/\rho_{2k}^-$ be the condition number. Further pick $k \geq s + \left(1 + \frac{4}{\eta^2(\rho_{2k}^-)^2}\right)\min\{s, r\}$ where $\eta \in (0, 1/\rho_{2k}^+)$. Then the support of $\bar{\boldsymbol{x}}$ is included in the iterate of PHT(r) within*

$$t_{\max} = \left(\frac{3\log\kappa}{\log(1/\mu)} + \frac{2\log(2/(1-\lambda))}{\log(1/\mu)} + 2\right)\|\bar{\boldsymbol{x}}\|_0$$

iterations, provided that for some constant $\lambda \in (0, 1)$,

$$\bar{\boldsymbol{x}}_{\min} \geq \frac{\sqrt{\kappa}+3}{\lambda\rho_{2k}^-}\left\|\nabla_{k+s}F(\bar{\boldsymbol{x}})\right\|.$$

Above, we have $\mu = 1 - \frac{\eta\rho_{2k}^-(1-\eta\rho_{2k}^+)}{2}$.

We discuss the salient features of Theorem 2.13 compared to Prop. 2.11 and Theorem 2.12. First, note that we can pick $\eta = \Theta(1/(2\rho_{2k}^+))$ in the above theorem, which results in $\mu = \Theta(1 - \frac{1}{\kappa})$.

So the iteration complexity is essentially given by $\mathcal{O}\left(s\kappa \log \kappa\right)$ that is similar to the one in Prop. 2.11. However, in Theorem 2.13, the sparsity parameter k is set to be $\Theta(s + \kappa^2 \min\{s, r\})$ which guarantees support inclusion. We pose an open question of whether the $\bar{\boldsymbol{x}}_{\min}$-condition might be refined, in that it currently scales with $\sqrt{\kappa}$ which is stringent for ill-conditioned problems. Another important consequence implied by the theorem is that the sparsity parameter k actually depends on the minimum of s and r. Consider $r = 1$ which corresponds to the OMPR algorithm. Then $k = \Theta(s + \kappa^2)$ suffices. In contrast, previous work of [75, 158, 131, 129] only obtained theoretical result for $k = \Theta(\kappa^2 s)$, owing to a restricted problem setting. We also note that even in the original OMPR paper [73] and its latest version [74], such an RIP-free condition was not established.

2.3.2 Statistical Results

Until now, all of our theoretical results are phrased in terms of deterministic conditions (i.e., RSC, RSS and $\bar{\boldsymbol{x}}_{\min}$). It is known that these conditions can be satisfied by prevalent statistical models such as linear regression and logistic regression. Here, we give detailed statistical results for sparse linear regression (2.13), and we refer the reader to [2, 75, 131, 129] for other applications.

Recall the results in Lemma 2.10. For Proposition 2.11, recall that the sparsity level of RSC and RSS is $2s$. Hence, if we pick the sample size $n = q \cdot 2\mathrm{C}_1 s \log d / \sigma_{\min}(\Sigma)$ for some $q > 1$, then

$$\frac{4\sqrt{2} + 2\sqrt{\kappa_{2s}}}{\rho_{2s}^{-}} \left\|\nabla_{2s} F(\bar{\boldsymbol{x}})\right\| \leq 4\omega \frac{2\sqrt{2} + \sqrt{\frac{\sigma_{\max}(\Sigma)}{\sigma_{\min}(\Sigma)}} \cdot \sqrt{\frac{1+\mathrm{C}_2/q\mathrm{C}_1}{1-1/q}}}{(1-1/q)\sqrt{q\mathrm{C}_1 \sigma_{\min}(\Sigma)}}.$$

The right-hand side is monotonically decreasing with q, which indicates that as soon as we pick q large enough, it becomes smaller than $\bar{\boldsymbol{x}}_{\min}$. To be more concrete, consider that the covariance matrix Σ is the identity matrix for which $\sigma_{\min}(\Sigma) = \sigma_{\max}(\Sigma) = 1$. Now suppose that $q \geq 2$, which gives an upper bound

$$\frac{4\sqrt{2} + 2\sqrt{\kappa_{2s}}}{\rho_{2s}^{-}} \left\|\nabla_{2s} F(\bar{\boldsymbol{x}})\right\| \leq \frac{8\omega(2\sqrt{2} + \sqrt{2 + \mathrm{C}_2/\mathrm{C}_1})}{\sqrt{q\mathrm{C}_1}}.$$

Thus, in order to fulfill the $\bar{\boldsymbol{x}}_{\min}$-condition in Prop. 2.11, it suffices to pick

$$q = \max\left\{2, \left(\frac{8\omega(2\sqrt{2}+\sqrt{2+\mathrm{C}_2/\mathrm{C}_1)}}{\sqrt{\mathrm{C}_1}\bar{\boldsymbol{x}}_{\min}}\right)^2\right\}.$$

For Theorem 2.12, it essentially asks for a well-conditioned design matrix at the sparsity level $2s+r$. Note that $\kappa_{2s+r} \geq \sigma_{\max}(\Sigma)/\sigma_{\min}(\Sigma)$, which in return requires a well-conditioned covariance matrix. Thus, to guarantee that $\kappa_{2s+r} \leq 1+\epsilon$ for some $\epsilon > 0$, it suffices to choose Σ such that $\sigma_{\max}(\Sigma)/\sigma_{\min}(\Sigma) < 1+\epsilon$ and pick $n = q \cdot \mathrm{C}_1(2s+r)\log d/\sigma_{\min}(\Sigma)$ with

$$q = \frac{1+\epsilon+\mathrm{C}_1^{-1}\mathrm{C}_2\sigma_{\max}(\Sigma)/\sigma_{\min}(\Sigma)}{1+\epsilon-\sigma_{\max}(\Sigma)/\sigma_{\min}(\Sigma)}.$$

Finally, Theorem 2.13 asserts support inclusion by expanding the support size of the iterates. Suppose that $\eta = 1/(2\rho_{2k}^{+})$, which results in $k \geq s + (16\kappa_{2k}^2+1)\min\{r,s\}$. Given that the condition number κ_{2k} is always greater than 1, we can pick $k \geq s + 20\kappa_{2k}^2\min\{r,s\}$. At a first sight, this seems to be weird in that k depends on the condition number κ_{2k} which itself relies on the choice of k. In the following, we present concrete sample complexity showing that this condition can be met. We will focus on two extreme cases: $r=1$ and $r=s$.

For $r=1$, we require $k \geq s + 20\kappa_{2k}^2$. Let us pick $n = 4\mathrm{C}_1 k\log d/\sigma_{\min}(\Sigma)$. In this way, we obtain $\rho_{2k}^{-} = \frac{1}{2}\sigma_{\min}$ and $\rho_{2k}^{+} \leq (1+\frac{\mathrm{C}_2}{2\mathrm{C}_1})\sigma_{\max}(\Sigma)$. It then follows that the condition number of the design matrix $\kappa_{2k} \leq (2+\frac{\mathrm{C}_2}{\mathrm{C}_1})\sigma_{\max}(\Sigma)/\sigma_{\min}(\Sigma)$. Consequently, we can set the parameter

$$k = s + 20\left(\left(2+\frac{\mathrm{C}_2}{\mathrm{C}_1}\right)\frac{\sigma_{\max}(\Sigma)}{\sigma_{\min}(\Sigma)}\right)^2.$$

Note that the above quantities depend only on the covariance matrix. Again, if Σ is the identity matrix, the sample complexity is $\mathcal{O}(s\log d)$.

For $r=s$, likewise $k \geq 20\kappa_{2k}^2 s$ suffices. Following the deduction above, we get

$$k = 20\left(\left(2+\frac{\mathrm{C}_2}{\mathrm{C}_1}\right)\frac{\sigma_{\max}(\Sigma)}{\sigma_{\min}(\Sigma)}\right)^2 s.$$

2.3.3 Simulation

We complement our theoretical results by performing numerical experiments in this section. In particular, we are interested in two aspects: first, the number of iterations required to identify the support of an s-sparse signal; second, the tradeoff between the iteration number and percentage of success resulted from different choices of the freedom parameter r.

We consider the compressed sensing model $\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}} + 0.01\boldsymbol{e}$, where the dimension $d = 200$ and the entries of $\boldsymbol{A}$ and $\boldsymbol{e}$ are i.i.d. normal variables. Given a sparsity level s, we first uniformly choose the support of $\bar{\boldsymbol{x}}$, and assign values to the non-zeros with i.i.d. normals. There are two configurations: the sparsity s and the sample size n. Given s and n, we independently generate 100 signals and test PHT(r) on them. We say PHT(r) succeeds in a trial if it returns an iterate with correct support within 10 thousands iterations. Otherwise we mark the trial as failure. Iteration numbers to be reported are counted only on those success trials. The step size η is fixed to be the unit, though one can tune it using cross-validation for better performance.

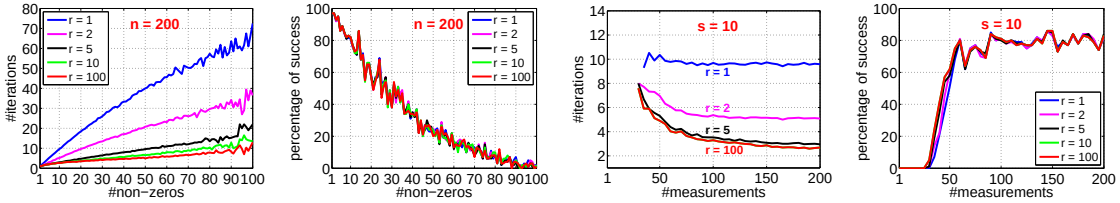


Figure 2.3: **PHT: Iteration number and success percentage against sparsity and sample size.**

To study how the iteration number scales with the sparsity in practice, we fix $n = 200$ and tune s from 1 to 100. We test different freedom parameter r on these signals. The results are shown in the leftmost figure in Figure 2.3. As our theory predicted, we observe that within $\mathcal{O}(s)$ iterations, PHT(r) precisely identifies the true support. In the second subfigure, we plot the percentage of success against the sparsity. It appears that PHT(r) lays on top of each other. This is possibly because we used a sufficiently large sample size.

Next, we fix $s = 10$ and vary n from 1 to 200. Surprisingly, from the rightmost figure, we do not observe performance degrade using a large freedom parameter. So we conjecture that the $\bar{\boldsymbol{x}}_{\min}$-condition we established can be refined.

Figure 2.3 also illustrates an interesting phenomenon: after a particular threshold, say $r = 5$, PHT(r) does not significantly reduces the iteration number by increasing r. This cannot be

explained by our theorems in the section. We leave it as a promising research direction.

2.4 Conclusion

In this chapter, we have studied the iteration complexity of the hard thresholding pursuit algorithm for recovering the support of an arbitrary s-sparse signal. We have shown that if the iterates of HTP are exact solutions, HTP recovers the support within $\mathcal{O}\left(s\kappa\log\kappa\right)$ iterations where κ is the condition number. In a more practical machine learning setting, we have proved that even with inexact solutions, support recovery is still possible with the same iteration bound. We also have presented a principled analysis on a family of hard thresholding algorithms. To facilitate our analysis, we appealed to the recently proposed partial hard thresholding operator. We have shown that under the RIP condition or the relaxed sparsity condition, the PHT(r) algorithm recovers the support of an arbitrary sparse signal $\bar{\boldsymbol{x}}$ within $O(\|\bar{\boldsymbol{x}}\|_0\,\kappa\log\kappa)$ iterations, provided that a generalized signal-to-noise ratio condition is satisfied. On account of our unified analysis, we have established the best known bound for HTP and OMPR. We have also illustrated that the simulation results agree with our finding that the iteration number is proportional to the sparsity.

There are several interesting future directions. First, it would be interesting to examine if we can close the logarithmic factor $\log\kappa$ in the iteration bound. Second, it is also useful to study RIP-free conditions for two-stage PHT algorithms such as CoSaMP. Finally, we pose the open question of whether one can improve the $\sqrt{\kappa}$ factor in the $\bar{\boldsymbol{x}}_{\min}$-condition.

2.A Technical Lemmas

In this section, we collect technical lemmas that will be invoked in the proof of our main results. Throughout our proof, we presume without loss of generality that the elements in $\bar{\boldsymbol{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_d)$ are in descending order by their magnitude, i.e., $|\bar{x}_1| \geq |\bar{x}_2| \geq \cdots \geq |\bar{x}_s|$ and $\bar{x}_i = 0$ for $s < i \leq d$. We also write $[n] := \{1, 2, \ldots, n\}$ for brevity.

To ease notation, we also expand the partial hard thresholding algorithm with freedom parameter

r as follows at the t-th iteration:

$$\begin{aligned}
\boldsymbol{b}^t &= \boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}) \\
J^t &= S^{t-1} \cup \operatorname{supp}\left(\nabla F(\boldsymbol{x}^{t-1}), r\right) \\
\boldsymbol{y}^t &= \mathcal{H}_k\left(() \, \boldsymbol{b}^t_{J^t}\right) \\
S^t &= \operatorname{supp}\left(\boldsymbol{y}^t\right) \\
\boldsymbol{x}^t &= \underset{\operatorname{supp}(\boldsymbol{x}) \subset S^t}{\arg\min} F(\boldsymbol{x})
\end{aligned}$$

The following lemma is a characterization of the co-coercivity of the objective function $F(\boldsymbol{x})$. A similar result was obtained in Corollary 8 of [110] but we present a refined analysis which is essential for our purpose.

Lemma 2.14. *For a given support set Ω, assume that the function $F(\boldsymbol{x})$ is $\rho^+_{|\Omega|}$-RSS and is ρ^-_K-RSC for some sparsity level K. Then, for all vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ with $|\operatorname{supp}(\boldsymbol{x} - \boldsymbol{x}') \cup \Omega| \leq K$, we have*

$$\left\|\nabla_\Omega F(\boldsymbol{x}') - \nabla_\Omega F(\boldsymbol{x})\right\|^2 \leq 2\rho^+_{|\Omega|}\big(F(\boldsymbol{x}') - F(\boldsymbol{x}) - \left\langle \nabla F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \right\rangle\big).$$

Proof. We define an auxiliary function

$$G(\boldsymbol{w}) := F(\boldsymbol{w}) - \langle \nabla F(\boldsymbol{x}), \boldsymbol{w} \rangle .$$

For all vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$, we have

$$\left\|\nabla G(\boldsymbol{w}) - \nabla G(\boldsymbol{w}')\right\| = \left\|\nabla F(\boldsymbol{w}) - \nabla F(\boldsymbol{w}')\right\| \leq \rho^+_{|\operatorname{supp}(\boldsymbol{w}-\boldsymbol{w}')|} \left\|\boldsymbol{w} - \boldsymbol{w}'\right\|,$$

which is equivalent to

$$G(\boldsymbol{w}) - G(\boldsymbol{w}') - \left\langle \nabla G(\boldsymbol{w}'), \boldsymbol{w} - \boldsymbol{w}' \right\rangle \leq \frac{\rho^+_r}{2} \left\|\boldsymbol{w} - \boldsymbol{w}'\right\|^2, \tag{2.19}$$

where $r := |\operatorname{supp}(\boldsymbol{w} - \boldsymbol{w}')|$. On the other hand, due to the RSC property of $F(\boldsymbol{x})$, we obtain

$$G(\boldsymbol{w}) - G(\boldsymbol{x}) = F(\boldsymbol{w}) - F(\boldsymbol{x}) - \langle \nabla F(\boldsymbol{x}), \boldsymbol{w} - \boldsymbol{x} \rangle \geq \frac{\rho^-_{|\operatorname{supp}(\boldsymbol{w}-\boldsymbol{x})|}}{2} \|\boldsymbol{w} - \boldsymbol{x}\|^2 \geq 0,$$

provided that $|\mathrm{supp}\,(\boldsymbol{w}-\boldsymbol{x})| \leq K$. For the given support set Ω, we pick $\boldsymbol{w} = \boldsymbol{x}' - \frac{1}{\rho^+_{|\Omega|}}\nabla_\Omega G(\boldsymbol{x}')$. Clearly, for such a choice of $\boldsymbol{w}$, we have $\mathrm{supp}\,(\boldsymbol{w}-\boldsymbol{x}) = \mathrm{supp}\,(\boldsymbol{x}-\boldsymbol{x}') \cup \Omega$. Hence, by assuming that $|\mathrm{supp}\,(\boldsymbol{x}-\boldsymbol{x}') \cup \Omega|$ is not larger than K, we get

$$\begin{aligned} G(\boldsymbol{x}) &\leq G\left(\boldsymbol{x}' - \frac{1}{\rho^+_{|\Omega|}}\nabla_\Omega G(\boldsymbol{x}')\right) \\ &\leq G(\boldsymbol{x}') + \left\langle \nabla G(\boldsymbol{x}'), -\frac{1}{\rho^+_{|\Omega|}}\nabla_\Omega G(\boldsymbol{x}')\right\rangle + \frac{1}{2\rho^+_{|\Omega|}}\left\|\nabla_\Omega G(\boldsymbol{x}')\right\|^2 \\ &= G(\boldsymbol{x}') - \frac{1}{2\rho^+_{|\Omega|}}\left\|\nabla_\Omega G(\boldsymbol{x}')\right\|^2, \end{aligned}$$

where the second inequality follows from (2.19). Now expanding $\nabla_\Omega G(\boldsymbol{x}')$ and rearranging the terms gives the desired result. □

Lemma 2.15 (Lemma 1 in [148])**.** *Let $\boldsymbol{u}$ and $\boldsymbol{b}$ be two distinct vectors and let $W = \mathrm{supp}\,(\boldsymbol{u}) \cap \mathrm{supp}\,(\boldsymbol{b})$. Also, let U be the support set of the top r (in magnitude) elements in $\boldsymbol{u}$. Then, the following holds for all $r \geq 1$:*

$$\langle \boldsymbol{u}, \boldsymbol{b}\rangle \leq \sqrt{\left\lceil \frac{|W|}{r}\right\rceil}\,\|\boldsymbol{u}_U\| \cdot \|\boldsymbol{b}_W\|.$$

Lemma 2.16. *Suppose that $F(\boldsymbol{x})$ is ρ^-_K-RSC and ρ^+_K-RSS for some sparsity level $K > 0$. Then for all $\theta \in \mathbb{R}$, all vectors $\boldsymbol{x}$, $\boldsymbol{x}' \in \mathbb{R}^d$ and for any Hessian matrix $\boldsymbol{H}$ of $F(\boldsymbol{x})$, we have*

$$\left|\left\langle \boldsymbol{x}, (\boldsymbol{I}-\theta\boldsymbol{H})\boldsymbol{x}'\right\rangle\right| \leq \phi_K\,\|\boldsymbol{x}\| \cdot \left\|\boldsymbol{x}'\right\|,$$

provided that $|\mathrm{supp}\,(\boldsymbol{x}) \cup \mathrm{supp}\,(\boldsymbol{x}')| \leq K$, and

$$\|((\boldsymbol{I}-\theta\boldsymbol{H})\boldsymbol{x})_\Omega\| \leq \phi_K\,\|\boldsymbol{x}\|, \quad \textit{if } |\Omega \cup \mathrm{supp}\,(\boldsymbol{x})| \leq K,$$

where

$$\phi_K = \max\left\{\left|\theta\rho^-_K - 1\right|, \left|\theta\rho^+_K - 1\right|\right\}.$$

Proof. Since $\boldsymbol{H}$ is a Hessian matrix, we always have a decomposition $\boldsymbol{H} = \boldsymbol{A}^\top\boldsymbol{A}$ for some matrix

$\boldsymbol{A}$. Denote $T = \operatorname{supp}(\boldsymbol{x}) \cup \operatorname{supp}(\boldsymbol{x}')$. By simple algebra, we have

$$\begin{aligned}
\left|\left\langle \boldsymbol{x}, (\boldsymbol{I} - \theta \boldsymbol{H})\boldsymbol{x}' \right\rangle\right| &= \left|\left\langle \boldsymbol{x}, \boldsymbol{x}' \right\rangle - \theta \left\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}' \right\rangle\right| \\
&\overset{\zeta_1}{=} \left|\left\langle \boldsymbol{x}, \boldsymbol{x}' \right\rangle - \theta \left\langle \boldsymbol{A}_T\boldsymbol{x}, \boldsymbol{A}_T\boldsymbol{x}' \right\rangle\right| \\
&= \left|\left\langle \boldsymbol{x}, (\boldsymbol{I} - \theta \boldsymbol{A}_T^\top \boldsymbol{A}_T)\boldsymbol{x}' \right\rangle\right| \\
&\leq \left\|\boldsymbol{I} - \theta \boldsymbol{A}_T^\top \boldsymbol{A}_T\right\| \cdot \|\boldsymbol{x}\| \cdot \left\|\boldsymbol{x}'\right\| \\
&\overset{\zeta_2}{\leq} \max\left\{\left|\theta\rho_K^- - 1\right|, \left|\theta\rho_K^+ - 1\right|\right\} \cdot \|\boldsymbol{x}\| \cdot \left\|\boldsymbol{x}'\right\|.
\end{aligned}$$

Here, ζ_1 follows from the fact that $\operatorname{supp}(\boldsymbol{x}) \cup \operatorname{supp}(\boldsymbol{x}') = T$ and ζ_2 holds because the RSC and RSS properties imply that the singular values of any Hessian matrix restricted on an K-sparse support set are lower and upper bounded by ρ_K^- and ρ_K^+, respectively.

For some index set Ω subject to $|\Omega \cup \operatorname{supp}(\boldsymbol{x})| \leq K$, let $\boldsymbol{x}' = ((\boldsymbol{I} - \theta\boldsymbol{H})\boldsymbol{x})_\Omega$. We immediately obtain

$$\left\|\boldsymbol{x}'\right\|^2 = \left\langle \boldsymbol{x}', (\boldsymbol{I} - \theta\boldsymbol{H})\boldsymbol{x} \right\rangle \leq \phi_K \left\|\boldsymbol{x}'\right\| \cdot \|\boldsymbol{x}\|,$$

indicating

$$\|((\boldsymbol{I} - \theta\boldsymbol{H})\boldsymbol{x})_\Omega\| \leq \phi_K \|\boldsymbol{x}\|.$$

The proof is complete. □

Lemma 2.17. *Suppose that $F(\boldsymbol{x})$ is ρ_K^--RSC and ρ_K^+-RSS for some sparsity level $K > 0$. For all vectors $\boldsymbol{x}$, $\boldsymbol{x}' \in \mathbb{R}^d$ and support set T such that $|\operatorname{supp}(\boldsymbol{x} - \boldsymbol{x}') \cup T| \leq K$, for all $\theta \in \mathbb{R}$*

$$\left\|\left(\boldsymbol{x} - \boldsymbol{x}' - \theta\nabla F(\boldsymbol{x}) + \theta\nabla F(\boldsymbol{x}')\right)_T\right\| \leq \phi_K \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|,$$

where $\phi_K = \max\left\{\left|\theta\rho_K^- - 1\right|, \left|\theta\rho_K^+ - 1\right|\right\}$.

Proof. In fact, for any two vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$, there always exists a quantity $t \in [0, 1]$, such that

$$\nabla F(\boldsymbol{x}) - \nabla F(\boldsymbol{x}') = \nabla^2 F\left(t\boldsymbol{x} + (1-t)\boldsymbol{x}'\right)(\boldsymbol{x} - \boldsymbol{x}').$$

Let $\boldsymbol{H} = \nabla^2 F(t\boldsymbol{x} + (1-t)\boldsymbol{x}')$. We write

$$\begin{aligned}\left\|\left(\boldsymbol{x} - \boldsymbol{x}' - \theta\nabla F(\boldsymbol{x}) + \theta\nabla F(\boldsymbol{x}')\right)_T\right\| &= \left\|(\boldsymbol{x} - \boldsymbol{x}' - \theta\boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}'))_T\right\| \\ &= \left\|((\boldsymbol{I} - \theta\boldsymbol{H})(\boldsymbol{x} - \boldsymbol{x}'))_T\right\| \\ &\leq \phi_K \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|,\end{aligned}$$

where the last inequality applies Lemma 2.16. □

Lemma 2.18. *Suppose that $F(\boldsymbol{x})$ is ρ_K^--RSC. Then for any vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ with $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq K$, the following holds:*

$$\left\|\boldsymbol{x} - \boldsymbol{x}'\right\| \leq \sqrt{\frac{2\max\{F(\boldsymbol{x}) - F(\boldsymbol{x}'), 0\}}{\rho_K^-}} + \frac{2\left\|(\nabla F(\boldsymbol{x}'))_\Omega\right\|}{\rho_K^-},$$

where $\Omega = \operatorname{supp}(\boldsymbol{x} - \boldsymbol{x}')$.

Proof. The RSC property immediately implies

$$\begin{aligned}F(\boldsymbol{x}) - F(\boldsymbol{x}') &\geq \left\langle \nabla F(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \right\rangle + \frac{\rho_K^-}{2}\left\|\boldsymbol{x} - \boldsymbol{x}'\right\|^2 \\ &\geq -\left\|\nabla_\Omega F(\boldsymbol{x}')\right\| \cdot \left\|\boldsymbol{x} - \boldsymbol{x}'\right\| + \frac{\rho_K^-}{2}\left\|\boldsymbol{x} - \boldsymbol{x}'\right\|^2.\end{aligned}$$

Discussing the sign of $F(\boldsymbol{x}) - F(\boldsymbol{x}')$ and solving the above quadratic inequality completes the proof. □

Proposition 2.19. *Suppose that $\bar{\boldsymbol{x}}$ is s-sparse and for all $t \geq 1$, $\boldsymbol{x}^t$ is k-sparse. Further assume that $F(\boldsymbol{x})$ is ρ_{k+s}^--RSC and ρ_{k+s}^+-RSS. Let $\kappa := \rho_{k+s}^+/\rho_{k+s}^-$. If for all $t \geq 1$*

$$F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) \leq \mu_t\left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})\right) + \tau,$$

where $0 < \mu_t \leq \mu$ for some $0 < \mu < 1$, $\tau \geq 0$, then we have

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \sqrt{2\kappa}(\sqrt{\mu_1\mu_2\ldots\mu_t})\left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{3}{\rho_{k+s}^-}\left\|\nabla_{k+s}F(\bar{\boldsymbol{x}})\right\| + \sqrt{\frac{2\tau}{\rho_{k+s}^-(1-\mu)}}.$$

Proof. The RSS property implies that

$$\begin{aligned} F(\boldsymbol{x}^0) - F(\bar{\boldsymbol{x}}) &\leq \left\langle \nabla F(\bar{\boldsymbol{x}}), \boldsymbol{x}^0 - \bar{\boldsymbol{x}} \right\rangle + \frac{\rho_{k+s}^+}{2} \left\| \boldsymbol{x}^0 - \bar{\boldsymbol{x}} \right\|^2 \\ &\leq \frac{\rho_{k+s}^+}{2} \left\| \boldsymbol{x}^0 - \bar{\boldsymbol{x}} \right\|^2 + \frac{1}{2\rho_{k+s}^+} \left\| \nabla_{k+s} F(\bar{\boldsymbol{x}}) \right\|^2 + \frac{\rho_{k+s}^+}{2} \left\| \boldsymbol{x}^0 - \bar{\boldsymbol{x}} \right\|^2 \\ &\leq \rho_{k+s}^+ \left\| \boldsymbol{x}^0 - \bar{\boldsymbol{x}} \right\|^2 + \frac{1}{2\rho_{k+s}^+} \left\| \nabla_{k+s} F(\bar{\boldsymbol{x}}) \right\|^2. \end{aligned}$$

Denote $\mu_{1:t} = \mu_1 \mu_2 \dots \mu_t$. We obtain

$$F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) \leq \mu_{1:t} \rho_{k+s}^+ \left\| \boldsymbol{x}^0 - \bar{\boldsymbol{x}} \right\|^2 + \frac{1}{2\rho_{k+s}^+} \left\| \nabla_{k+s} F(\bar{\boldsymbol{x}}) \right\|^2 + \frac{\tau}{1-\mu}.$$

By Lemma 2.18, we have

$$\begin{aligned} &\left\| \boldsymbol{x}^t - \bar{\boldsymbol{x}} \right\| \\ &\leq \sqrt{\frac{2}{\rho_{k+s}^-}} \sqrt{\mu_{1:t} \rho_{k+s}^+ \left\| \boldsymbol{x}^0 - \bar{\boldsymbol{x}} \right\|^2 + \frac{1}{2\rho_{k+s}^+} \left\| \nabla_{k+s} F(\bar{\boldsymbol{x}}) \right\|^2 + \frac{\tau}{1-\mu}} + \frac{2}{m} \left\| \nabla_{k+s} F(\bar{\boldsymbol{x}}) \right\| \\ &\leq \sqrt{2\kappa} (\sqrt{\mu_{1:t}}) \left\| \boldsymbol{x}^0 - \bar{\boldsymbol{x}} \right\| + \sqrt{\frac{1}{\rho_{k+s}^- \rho_{k+s}^+}} \left\| \nabla_{k+s} F(\bar{\boldsymbol{x}}) \right\| + \frac{2}{\rho_{k+s}^-} \left\| \nabla_{k+s} F(\bar{\boldsymbol{x}}) \right\| + \sqrt{\frac{2\tau}{\rho_{k+s}^- (1-\mu)}} \\ &\leq \sqrt{2\kappa} (\sqrt{\mu_{1:t}}) \left\| \boldsymbol{x}^0 - \bar{\boldsymbol{x}} \right\| + \frac{3}{\rho_{k+s}^-} \left\| \nabla_{k+s} F(\bar{\boldsymbol{x}}) \right\| + \sqrt{\frac{2\tau}{\rho_{k+s}^- (1-\mu)}}. \end{aligned}$$

The proof is complete. □

Lemma 2.20. *Suppose that $F(\boldsymbol{x})$ is ρ_K^--RSC and ρ_K^+-RSS for some sparsity level $K > 0$. Let $\kappa := \rho_K^+ / \rho_K^-$. For all vectors $\boldsymbol{x}$, $\boldsymbol{x}' \in \mathbb{R}^d$ with $|\operatorname{supp}(\boldsymbol{x}) \cup \operatorname{supp}(\boldsymbol{x}')| \leq K$, we have*

$$\begin{aligned} \left\| \boldsymbol{x} - \boldsymbol{x}' \right\| &\leq \kappa \left\| \boldsymbol{x}'_{\overline{\Omega}} \right\| + \frac{1}{\rho_K^-} \left\| (\nabla F(\boldsymbol{x}) - \nabla F(\boldsymbol{x}'))_\Omega \right\|, \\ \left\| (\boldsymbol{x} - \boldsymbol{x}')_\Omega \right\| &\leq \left(1 - \frac{1}{\kappa} \right) \left\| \boldsymbol{x} - \boldsymbol{x}' \right\| + \frac{1}{\rho_K^-} \left\| (\nabla F(\boldsymbol{x}) - \nabla F(\boldsymbol{x}'))_\Omega \right\|. \end{aligned}$$

where Ω is the support set of $\boldsymbol{x}$.

Proof. We begin with bounding the ℓ_2-norm of the difference of $\boldsymbol{x}$ and $\boldsymbol{x}'$. Let $T = \operatorname{supp}(\boldsymbol{x}')$. For

any positive scalar $\theta \in \mathbb{R}$ we have

$$\begin{aligned}
\left\|(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\|^2 &= \left\langle \boldsymbol{x}-\boldsymbol{x}'-\theta\nabla F(\boldsymbol{x})+\theta\nabla F(\boldsymbol{x}'),(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\rangle \\
&\quad + \theta\left\langle \nabla F(\boldsymbol{x})-\nabla F(\boldsymbol{x}'),(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\rangle \\
&\le \left\|(\boldsymbol{x}-\boldsymbol{x}'-\theta\nabla F(\boldsymbol{x})+\theta\nabla F(\boldsymbol{x}'))_{\Omega}\right\|\cdot\left\|(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\| \\
&\quad + \theta\left\|\left(\nabla F(\boldsymbol{x})-\nabla F(\boldsymbol{x}')\right)_{\Omega}\right\|\cdot\left\|(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\| \\
&\le \left\|\boldsymbol{x}-\boldsymbol{x}'-\theta(\nabla F(\boldsymbol{x}))_{\Omega\cup T}+\theta(\nabla F(\boldsymbol{x}'))_{\Omega\cup T}\right\|\cdot\left\|(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\| \\
&\quad + \theta\left\|\left(\nabla F(\boldsymbol{x})-\nabla F(\boldsymbol{x}')\right)_{\Omega}\right\|\cdot\left\|(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\| \\
&\le \phi_K\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|\cdot\left\|(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\|+\theta\left\|\left(\nabla F(\boldsymbol{x})-\nabla F(\boldsymbol{x}')\right)_{\Omega}\right\|\cdot\left\|(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\|,
\end{aligned}$$

where we recall that ϕ_K is given in Lemma 2.16. Dividing both sides by $\|(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\|$ gives

$$\left\|(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\| \le \phi_K\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|+\theta\left\|(\nabla F(\boldsymbol{x})-\nabla F(\boldsymbol{x}'))_{\Omega}\right\|.$$

On the other hand,

$$\begin{aligned}
\left\|\boldsymbol{x}-\boldsymbol{x}'\right\| &\le \left\|(\boldsymbol{x}-\boldsymbol{x}')_{\Omega}\right\|+\left\|(\boldsymbol{x}-\boldsymbol{x}')_{\overline{\Omega}}\right\| \\
&\le \phi_K\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|+\theta\left\|(\nabla F(\boldsymbol{x})-\nabla F(\boldsymbol{x}'))_{\Omega}\right\|+\left\|\boldsymbol{x}'_{\overline{\Omega}}\right\|.
\end{aligned}$$

Hence, we have

$$\left\|\boldsymbol{x}-\boldsymbol{x}'\right\| \le \frac{1}{1-\phi_K}\left\|\boldsymbol{x}'_{\overline{\Omega}}\right\|+\frac{\theta}{1-\phi_K}\left\|(\nabla F(\boldsymbol{x})-\nabla F(\boldsymbol{x}'))_{\Omega}\right\|.$$

Picking $\theta = 1/\rho_K^+$, we have $\phi_K = 1-\frac{1}{\kappa}$. Plugging these into the above and noting that $\rho_K^+ \ge \rho_K^-$ complete the proof. □

Lemma 2.21. *Consider the HTP algorithm with exact solution in (HTP3), or the PHT(r) algorithm. Assume $F(\boldsymbol{x})$ is ρ_{k+s}^--RSC. Then for all $t \ge 1$,*

$$\left\|\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2 \ge 2\rho_{k+s}^-\delta_t\left(F(\boldsymbol{x}^{t-1})-F(\bar{\boldsymbol{x}})\right),$$

where

$$\delta_t = \frac{\left|S^t \backslash S^{t-1}\right|}{|S^t \backslash S^{t-1}| + |S \backslash S^{t-1}|}.$$

Proof. The lemma holds clearly for either $S^t = S^{t-1}$ or $F(\boldsymbol{x}^t) \leq F(\bar{\boldsymbol{x}})$. Hence, in the following we only prove the result by assuming $S^t \neq S^{t-1}$ and $F(\boldsymbol{x}^t) > F(\bar{\boldsymbol{x}})$. Due to the RSC property, we have

$$F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^{t-1}) - \left\langle \nabla F(\boldsymbol{x}^{t-1}), \bar{\boldsymbol{x}} - \boldsymbol{x}^{t-1} \right\rangle \geq \frac{\rho_{k+s}^-}{2} \left\| \bar{\boldsymbol{x}} - \boldsymbol{x}^{t-1} \right\|^2,$$

which implies

$$\begin{aligned} \left\langle \nabla F(\boldsymbol{x}^{t-1}), -\bar{\boldsymbol{x}} \right\rangle &\geq \frac{\rho_{k+s}^-}{2} \left\| \bar{\boldsymbol{x}} - \boldsymbol{x}^{t-1} \right\|^2 + F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}}) \\ &\geq \sqrt{2\rho_{k+s}^-} \left\| \bar{\boldsymbol{x}} - \boldsymbol{x}^{t-1} \right\| \sqrt{F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})}. \end{aligned}$$

By invoking Lemma 2.15 with $\boldsymbol{u} = \nabla F(\boldsymbol{x}^{t-1})$ and $\boldsymbol{b} = -\bar{\boldsymbol{x}}$ therein, we have

$$\begin{aligned} \left\langle \nabla F(\boldsymbol{x}^{t-1}), -\bar{\boldsymbol{x}} \right\rangle &\leq \sqrt{\frac{|S \backslash S^{t-1}|}{|S^t \backslash S^{t-1}|} + 1} \left\| \nabla_{S^t \backslash S^{t-1}} F(\boldsymbol{x}^{t-1}) \right\| \cdot \left\| \bar{\boldsymbol{x}}_{S \backslash S^{t-1}} \right\| \\ &= \sqrt{\frac{|S \backslash S^{t-1}|}{|S^t \backslash S^{t-1}|} + 1} \left\| \nabla_{S^t \backslash S^{t-1}} F(\boldsymbol{x}^{t-1}) \right\| \cdot \left\| (\bar{\boldsymbol{x}} - \boldsymbol{x}^t)_{S \backslash S^{t-1}} \right\| \\ &\leq \sqrt{\frac{|S \backslash S^{t-1}|}{|S^t \backslash S^{t-1}|} + 1} \left\| \nabla_{S^t \backslash S^{t-1}} F(\boldsymbol{x}^{t-1}) \right\| \cdot \left\| \bar{\boldsymbol{x}} - \boldsymbol{x}^t \right\|. \end{aligned}$$

It is worth mentioning that the first inequality above holds because $\nabla F(\boldsymbol{x}^{t-1})$ is supported on $\overline{S^{t-1}}$ and $S^t \backslash S^{t-1}$ contains the $\left|S^t \backslash S^{t-1}\right|$ number of largest (in magnitude) elements of $\nabla F(\boldsymbol{x}^{t-1})$. Therefore, we obtain the result. □

Lemma 2.22. *Assume that $F(\boldsymbol{x})$ satisfies the properties of RSC and RSS at sparsity level $k+s+r$. Let $\rho^- := \rho_{k+s+r}^-$ and $\rho^+ := \rho_{k+s+r}^+$. Consider the support set $J^t = S^{t-1} \cup \operatorname{supp}\left(\nabla F(\boldsymbol{x}^{t-1}), r\right)$. We have for any $0 < \theta \leq 1/\rho^+$,*

$$\left\| \bar{\boldsymbol{x}}_{\overline{J^t}} \right\| \leq \nu(1 - \theta\rho^-) \left\| \boldsymbol{x}^{t-1} - \bar{\boldsymbol{x}} \right\| + \frac{\nu}{\rho^-} \left\| \nabla_{s+r} F(\bar{\boldsymbol{x}}) \right\|,$$

where $\nu = \sqrt{1+s/r}$. *In particular, picking* $\theta = 1/\rho^+$ *gives*

$$\left\|\bar{\boldsymbol{x}}_{\overline{J^t}}\right\| \leq \nu\left(1-\frac{1}{\kappa}\right)\left\|\boldsymbol{x}^{t-1}-\bar{\boldsymbol{x}}\right\| + \frac{\nu}{\rho^-}\left\|\nabla_{s+r}F(\bar{\boldsymbol{x}})\right\|.$$

Proof. Let $T = \operatorname{supp}\left(\nabla F(\boldsymbol{x}^{t-1}), r\right)$. Then $J^t = S^{t-1}\cup T$ and $S^{t-1}\cap T = \emptyset$. Since T contains the top r elements of $\nabla F(\boldsymbol{x}^{t-1})$, we have that each element in $T\backslash S$ is larger (in magnitude) than that in $S\backslash T$. In particular, we observe for $T \neq S$ that

$$\frac{1}{|T\backslash S|}\left\|\left(\nabla F(\boldsymbol{x}^{t-1})\right)_{T\backslash S}\right\|^2 \geq \frac{1}{|S\backslash T|}\left\|\left(\nabla F(\boldsymbol{x}^{t-1})\right)_{S\backslash T}\right\|^2,$$

which implies

$$\left\|\left(\nabla F(\boldsymbol{x}^{t-1})\right)_{T\backslash S}\right\| \geq \sqrt{\frac{r-|T\cap S|}{s-|T\cap S|}}\left\|\left(\nabla F(\boldsymbol{x}^{t-1})\right)_{S\backslash T}\right\| \geq \sqrt{\frac{r}{s}}\left\|\left(\nabla F(\boldsymbol{x}^{t-1})\right)_{S\backslash T}\right\|.$$

Since $\nabla F(\boldsymbol{x}^{t-1})$ is supported on $\overline{S^{t-1}}$, the LHS reads as

$$\left\|\left(\nabla F(\boldsymbol{x}^{t-1})\right)_{T\backslash S}\right\| = \left\|\left(\nabla F(\boldsymbol{x}^{t-1})\right)_{T\backslash(S\cup S^{t-1})}\right\| = \frac{1}{\theta}\left\|\left(\boldsymbol{x}^{t-1}-\theta\nabla F(\boldsymbol{x}^{t-1})-\bar{\boldsymbol{x}}\right)_{T\backslash(S\cup S^{t-1})}\right\|.$$

Now we look at the RHS. It follows that

$$\begin{aligned}\left\|\left(\nabla F(\boldsymbol{x}^{t-1})\right)_{S\backslash T}\right\| &= \left\|\left(\nabla F(\boldsymbol{x}^{t-1})\right)_{S\backslash(T\cup S^{t-1})}\right\| \\ &= \frac{1}{\theta}\left\|\left(\boldsymbol{x}^{t-1}-\theta\nabla F(\boldsymbol{x}^{t-1})-\bar{\boldsymbol{x}}\right)_{S\backslash(T\cup S^{t-1})} + \bar{\boldsymbol{x}}_{S\backslash(T\cup S^{t-1})}\right\| \\ &\geq \frac{1}{\theta}\left\|\bar{\boldsymbol{x}}_{S\backslash(T\cup S^{t-1})}\right\| - \frac{1}{\theta}\left\|\left(\boldsymbol{x}^{t}-\theta\nabla F(\boldsymbol{x}^{t})-\bar{\boldsymbol{x}}\right)_{S\backslash(T\cup S^{t-1})}\right\|.\end{aligned}$$

Hence,

$$\begin{aligned}
\left\|\bar{\boldsymbol{x}}_{\overline{J^t}}\right\| &= \left\|\bar{\boldsymbol{x}}_{S\setminus(T\cup S^{t-1})}\right\| \\
&\le \sqrt{\frac{s}{r}}\left\|\left(\boldsymbol{x}^{t-1}-\theta\nabla F(\boldsymbol{x}^{t-1})-\bar{\boldsymbol{x}}\right)_{T\setminus(S\cup S^{t-1})}\right\| + \left\|\left(\boldsymbol{x}^{t-1}-\theta\nabla F(\boldsymbol{x}^{t-1})-\bar{\boldsymbol{x}}\right)_{S\setminus(T\cup S^{t-1})}\right\| \\
&\le \sqrt{\frac{s}{r}}\left\|\left(\boldsymbol{x}^{t-1}-\theta\nabla F(\boldsymbol{x}^{t-1})-\bar{\boldsymbol{x}}\right)_{T\setminus S}\right\| + \left\|\left(\boldsymbol{x}^{t-1}-\theta\nabla F(\boldsymbol{x}^{t-1})-\bar{\boldsymbol{x}}\right)_{S\setminus T}\right\| \\
&\le \nu\left\|\left(\boldsymbol{x}^{t-1}-\theta\nabla F(\boldsymbol{x}^{t-1})-\bar{\boldsymbol{x}}\right)_{T\Delta S}\right\| \\
&\le \nu\left\|\left(\boldsymbol{x}^{t-1}-\theta\nabla F(\boldsymbol{x}^{t-1})-\bar{\boldsymbol{x}}+\theta\nabla F(\bar{\boldsymbol{x}})\right)_{T\Delta S}\right\| + \nu\theta\left\|(\nabla F(\bar{\boldsymbol{x}}))_{T\Delta S}\right\| \\
&\le \nu\phi_{k+s+r}\left\|\boldsymbol{x}^{t-1}-\bar{\boldsymbol{x}}\right\| + \nu\theta\left\|(\nabla F(\bar{\boldsymbol{x}}))_{T\Delta S}\right\|,
\end{aligned}$$

where $\nu=\sqrt{1+s/r}$ and the last inequality uses Lemma 2.17. For any $0<\theta\le 1/\rho^+$, we have

$$\left\|\bar{\boldsymbol{x}}_{\overline{J^t}}\right\| \le \nu(1-\theta\rho^-)\left\|\boldsymbol{x}^{t-1}-\bar{\boldsymbol{x}}\right\| + \frac{\nu}{\rho^-}\left\|\nabla_{s+r}F(\bar{\boldsymbol{x}})\right\|.$$

□

2.A.1 Crucial Lemmas

Lemma 2.23. *Consider the HTP algorithm, or the PHT(r) algorithm with $\eta<1/\rho^+_{2k}$. Assume that $F(\boldsymbol{x})$ is ρ^-_{2k}-RSC and ρ^+_{2k}-RSS. Further assume that the sequence of $\{\boldsymbol{x}^t\}_{t\ge0}$ satisfies*

$$\begin{aligned}
\left\|\boldsymbol{x}^t-\bar{\boldsymbol{x}}\right\| &\le \alpha\cdot\beta^t\left\|\boldsymbol{x}^0-\bar{\boldsymbol{x}}\right\|+\psi_1, \\
\left\|\boldsymbol{x}^t-\bar{\boldsymbol{x}}\right\| &\le \gamma\left\|\bar{\boldsymbol{x}}_{\overline{S^t}}\right\|+\psi_2,
\end{aligned}$$

for positive α, ψ_1, γ, ψ_2 and $0<\beta<1$. Suppose that at the n-th iteration ($n\ge0$), S^n contains the indices of top p (in magnitude) elements of $\bar{\boldsymbol{x}}$. Then, for any integer $1\le q\le s-p$, there exists an integer $\Delta\ge1$ determined by

$$\sqrt{2}\left|\bar{x}_{p+q}\right| > \alpha\gamma\cdot\beta^{\Delta-1}\left\|\bar{\boldsymbol{x}}_{\{p+1,\ldots,s\}}\right\|+\Psi$$

where

$$\Psi = \alpha\psi_2 + \psi_1 + \frac{1}{\underline{\rho}_{2k}} \left\| \nabla_2 F(\bar{\boldsymbol{x}}) \right\|,$$

such that $S^{n+\Delta}$ *contains the indices of top* $p+q$ *elements of* $\bar{\boldsymbol{x}}$ *provided that* $\Psi \leq \sqrt{2}\lambda \bar{\boldsymbol{x}}_{\min}$ *for some* $\lambda \in (0,1)$.

Proof. We aim at deriving a condition under which $[p+q] \subset S^{n+\Delta}$. To this end, it suffices to enforce

$$\min_{j\in[p+q]} \left| b_j^{n+\Delta} \right| > \max_{i\in\overline{S}} \left| b_i^{n+\Delta} \right|. \tag{2.20}$$

On one hand, for any $j \in [p+q]$,

$$\begin{aligned}
\left| b_j^{n+\Delta} \right| &= \left| \left(\boldsymbol{x}^{n+\Delta-1} - \eta \nabla F(\boldsymbol{x}^{n+\Delta-1}) \right)_j \right| \\
&\geq |\bar{x}_j| - \left| \left(\boldsymbol{x}^{n+\Delta-1} - \bar{\boldsymbol{x}} - \eta \nabla F(\boldsymbol{x}^{n+\Delta-1}) \right)_j \right| \\
&\geq |\bar{x}_{p+q}| - \left| \left(\boldsymbol{x}^{n+\Delta-1} - \bar{\boldsymbol{x}} - \eta \nabla F(\boldsymbol{x}^{n+\Delta-1}) \right)_j \right|.
\end{aligned}$$

On the other hand, for all $i \in \overline{S}$,

$$\left| b_i^{n+\Delta} \right| = \left| \left(\boldsymbol{x}^{n+\Delta-1} - \bar{\boldsymbol{x}} - \eta \nabla F(\boldsymbol{x}^{n+\Delta-1}) \right)_i \right|.$$

Hence, we know that to guarantee (2.20), it suffices to ensure for all $j \in [p+q]$ and $i \in \overline{S}$ that

$$|\bar{x}_{p+q}| > \left| \left(\boldsymbol{x}^{n+\Delta-1} - \bar{\boldsymbol{x}} - \eta \nabla F(\boldsymbol{x}^{n+\Delta-1}) \right)_j \right| + \left| \left(\boldsymbol{x}^{n+\Delta-1} - \bar{\boldsymbol{x}} - \eta \nabla F(\boldsymbol{x}^{n+\Delta-1}) \right)_i \right|.$$

Note that the right-hand side is upper bounded as follows:

$$\begin{aligned}
&\frac{1}{\sqrt{2}}\left|\left(\boldsymbol{x}^{n+\Delta-1}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}^{n+\Delta-1})\right)_j\right|+\frac{1}{\sqrt{2}}\left|\left(\boldsymbol{x}^{n+\Delta-1}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}^{n+\Delta-1})\right)_i\right| \\
\le&\left\|\left(\boldsymbol{x}^{n+\Delta-1}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}^{n+\Delta-1})\right)_{\{j,i\}}\right\| \\
\le&\left\|\left(\boldsymbol{x}^{n+\Delta-1}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}^{n+\Delta-1})+\eta\nabla F(\bar{\boldsymbol{x}})\right)_{\{j,i\}}\right\|+\eta\left\|(\nabla F(\bar{\boldsymbol{x}}))_{\{j,i\}}\right\| \\
\le&\,\phi_{2k}\left\|\boldsymbol{x}^{n+\Delta-1}-\bar{\boldsymbol{x}}\right\|+\eta\left\|\nabla_2 F(\bar{\boldsymbol{x}})\right\| \\
\le&\,\phi_{2k}\alpha\cdot\beta^{\Delta-1}\left\|\boldsymbol{x}^{n}-\bar{\boldsymbol{x}}\right\|+\phi\psi_1+\eta\left\|\nabla_2 F(\bar{\boldsymbol{x}})\right\|,
\end{aligned}$$

where ϕ_{2k} is given by Lemma 2.16. Note that $\phi_{2k}<1$ whenever $0<\eta<1/\rho_{2k}^{+}$. Moreover,

$$\left\|\boldsymbol{x}^{n}-\bar{\boldsymbol{x}}\right\|\le\gamma\left\|\bar{\boldsymbol{x}}_{\overline{S^n}}\right\|+\psi_2\le\gamma\left\|\bar{\boldsymbol{x}}_{\overline{[p]}}\right\|+\psi_2=\gamma\left\|\bar{\boldsymbol{x}}_{\{p+1,\ldots,s\}}\right\|+\psi_2.$$

Put all the pieces together, we have

$$\begin{aligned}
&\frac{1}{\sqrt{2}}\left|\left(\boldsymbol{x}^{n+\Delta-1}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}^{n+\Delta-1})\right)_j\right|+\frac{1}{\sqrt{2}}\left|\left(\boldsymbol{x}^{n+\Delta-1}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}^{n+\Delta-1})\right)_i\right| \\
\le&\,\alpha\gamma\cdot\beta^{\Delta-1}\left\|\bar{\boldsymbol{x}}_{\{p+1,\ldots,s\}}\right\|+\alpha\psi_2+\psi_1+\eta\left\|\nabla_2 F(\bar{\boldsymbol{x}})\right\| \\
\le&\,\alpha\gamma\cdot\beta^{\Delta-1}\left\|\bar{\boldsymbol{x}}_{\{p+1,\ldots,s\}}\right\|+\alpha\psi_2+\psi_1+\frac{1}{\rho_{2k}^{-}}\left\|\nabla_2 F(\bar{\boldsymbol{x}})\right\|.
\end{aligned}$$

Therefore, when

$$\sqrt{2}\left|\bar{\boldsymbol{x}}_{p+q}\right|>\alpha\gamma\cdot\beta^{\Delta-1}\left\|\bar{\boldsymbol{x}}_{\{p+1,\ldots,s\}}\right\|+\alpha\psi_2+\psi_1+\frac{1}{\rho_{2k}^{-}}\left\|\nabla_2 F(\bar{\boldsymbol{x}})\right\|,$$

we always have (2.20). Note that the above holds as far as $\Psi:=\alpha\psi_2+\psi_1+\frac{1}{\rho_{2k}^{-}}\left\|\nabla_2 F(\bar{\boldsymbol{x}})\right\|$ is strictly smaller than $\sqrt{2}\left|\bar{\boldsymbol{x}}_s\right|$. □

Theorem 2.24. *Assume the same conditions as in Lemma 2.23. Then HTP and PHT(r) successfully identify the support of $\bar{\boldsymbol{x}}$ using $\left(\frac{\log 2}{2\log(1/\beta)}+\frac{\log(\alpha\gamma/(1-\lambda))}{\log(1/\beta)}+2\right)s$ number of iterations.*

Proof. We partition the support set $S=[s]$ into K folds $S_1, S_2, \ldots, S_K$, where each S_i is defined

as follows:

$$S_i = \{s_{i-1}+1, \ldots, s_i\},\ \forall\, 1 \le i \le K.$$

Here, $s_0 = 0$ and for all $1 \le i \le K$, the quantity s_i is inductively given by

$$s_i = \max\left\{q:\ s_{i-1}+1 \le q \le s \text{ and } |\bar{x}_q| > \frac{1}{\sqrt{2}}\left|\bar{x}_{s_{i-1}+1}\right|\right\}.$$

In this way, we note that for any two index sets S_i and S_j, $S_i \cap S_j = \emptyset$ if $i \neq j$. We also know by the definition of s_i that

$$|\bar{x}_{s_i+1}| \le \frac{1}{\sqrt{2}}\left|\bar{x}_{s_{i-1}+1}\right|,\ \forall\, 1 \le i \le K-1. \tag{2.21}$$

Now we show that after a finite number of iterations, say n, the union of the S_i's is contained in S^n, i.e., the support set of the iterate $\boldsymbol{x}^n$. To this end, we prove that for all $0 \le i \le K$,

$$\bigcup_{t=0}^{i} S_t \subset S^{n_0+n_1+\cdots+n_i} \tag{2.22}$$

for some n_i's given below. Above, $S_0 = \emptyset$.

We pick $n_0 = 0$ and it is easy to verify that $S_0 \subset S^0$. Now suppose that (2.22) holds for $i-1$. That is, the index set of the top s_{i-1} elements of $\bar{\boldsymbol{x}}$ is contained in $S^{n_0+\cdots+n_{i-1}}$. Due to Lemma 2.23, (2.22) holds for i as long as n_i satisfies

$$\sqrt{2}\,|\bar{x}_{s_i}| > \alpha\gamma \cdot \beta^{n_i-1}\left\|\bar{\boldsymbol{x}}_{\{s_{i-1}+1,\ldots,s\}}\right\| + \Psi, \tag{2.23}$$

where Ψ is given in Lemma 2.23. Note that

$$\begin{aligned}
\left\|\bar{\boldsymbol{x}}_{\{s_{i-1}+1,\ldots,s\}}\right\|^2 &= \left\|\bar{\boldsymbol{x}}_{S_i}\right\|^2 + \cdots + \left\|\bar{\boldsymbol{x}}_{S_K}\right\|^2 \\
&\le (\bar{x}_{s_{i-1}+1})^2\,|S_i| + \cdots + (\bar{x}_{s_{r-1}+1})^2\,|S_K| \\
&\le (\bar{x}_{s_{i-1}+1})^2\left(|S_i| + 2^{-1}\,|S_{i+1}| + \cdots + 2^{i-K}\,|S_K|\right) \\
&< 2(\bar{x}_{s_i})^2\left(|S_i| + 2^{-1}\,|S_{i+1}| + \cdots + 2^{i-K}\,|S_K|\right),
\end{aligned}$$

where the second inequality follows from (2.21) and the last inequality follows from the definition of q_i. Denote for simplicity

$$W_i := |S_i| + 2^{-1} |S_{i+1}| + \cdots + 2^{i-K} |S_K| .$$

As we assumed $\Psi \leq \sqrt{2}\lambda \bar{\boldsymbol{x}}_{\min}$, we get

$$\alpha\gamma \cdot \beta^{n_i - 1} \left\| \bar{\boldsymbol{x}}_{\{s_{i-1}+1,\ldots,s\}} \right\| + \Psi < \sqrt{2}\alpha\gamma \, |\bar{x}_{s_i}| \, \beta^{n_i-1} \sqrt{W_i} + \sqrt{2}\lambda \, |\bar{x}_{s_i}| .$$

Picking

$$n_i = \log_{1/\beta} \frac{\alpha\gamma\sqrt{W_i}}{1-\lambda} + 2$$

guarantees (2.23). It remains to calculate the total number of iterations. In fact, we have

$$\begin{aligned}
t_{\max} &= n_0 + n_1 + \cdots + n_K \\
&= \frac{1}{2\log(1/\beta)} \sum_{i=1}^{K} \log W_i + K \cdot \frac{\log(\alpha\gamma/(1-\lambda))}{\log(1/\beta)} + 2K \\
&\overset{\zeta_1}{\leq} \frac{K}{2\log(1/\beta)} \log\left(\frac{1}{K} \sum_{i=1}^{K} W_i \right) + \left(\frac{\log(\alpha\gamma/(1-\lambda))}{\log(1/\beta)} + 2 \right) K \\
&\overset{\zeta_2}{\leq} \frac{K}{2\log(1/\beta)} \log\left(\frac{2}{K} \sum_{i=1}^{K} |S_i| \right) + \left(\frac{\log(\alpha\gamma/(1-\lambda))}{\log(1/\beta)} + 2 \right) K \\
&= \frac{K}{2\log(1/\beta)} \log \frac{2s}{K} + \left(\frac{\log(\alpha\gamma/(1-\lambda))}{\log(1/\beta)} + 2 \right) K \\
&\overset{\zeta_3}{\leq} \left(\frac{\log 2}{2\log(1/\beta)} + \frac{\log(\alpha\gamma/(1-\lambda))}{\log(1/\beta)} + 2 \right) s.
\end{aligned}$$

Above, ζ_1 immediately follows by observing that the logarithmic function is concave. ζ_2 uses the fact that after rearrangement, the coefficient of $|S_i|$ is $\sum_{j=0}^{i-1} 2^{-j}$ which is always smaller than 2. Finally, since the function $a \log(2s/a)$ is monotonically increasing with respect to a and $1 \leq a \leq s$, ζ_3 follows. $\square$

2.B Proofs for Section 2.2

With the technical lemmas, we are now in the position to prove the main results in Section 2.2.

2.B.1 Proof of Proposition 2.1

Proof. Due to the RSS property, we have

$$\begin{aligned} F(\boldsymbol{b}_{S^{t+1}}^{t+1}) - F(\boldsymbol{x}^t) &\leq \left\langle \nabla F(\boldsymbol{x}^t), \boldsymbol{b}_{S^{t+1}}^{t+1} - \boldsymbol{x}^t \right\rangle + \frac{\rho^+}{2} \left\| \boldsymbol{b}_{S^{t+1}}^{t+1} - \boldsymbol{x}^t \right\|^2 \\ &\overset{\zeta_1}{=} \left\langle \nabla_{S^{t+1}\backslash S^t} F(\boldsymbol{x}^t), \boldsymbol{b}_{S^{t+1}\backslash S^t}^{t+1} \right\rangle + \frac{\rho^+}{2} \Big(\left\| \boldsymbol{b}_{S^{t+1}\backslash S^t}^{t+1} \right\|^2 \\ &\quad + \left\| \boldsymbol{b}_{S^{t+1}\cap S^t}^{t+1} - \boldsymbol{x}_{S^{t+1}\cap S^t}^t \right\|^2 + \left\| \boldsymbol{x}_{S^t\backslash S^{t+1}}^t \right\|^2 \Big) \\ &\overset{\zeta_2}{\leq} \left\langle \nabla_{S^{t+1}\backslash S^t} F(\boldsymbol{x}^t), \boldsymbol{b}_{S^{t+1}\backslash S^t}^{t+1} \right\rangle + \rho^+ \left\| \boldsymbol{b}_{S^{t+1}\backslash S^t}^{t+1} \right\|^2 \\ &\overset{\zeta_3}{=} -\eta(1-\eta\rho^+) \left\| \nabla_{S^{t+1}\backslash S^t} F(\boldsymbol{x}^t) \right\|^2. \end{aligned}$$

Above, we observe that $\nabla F(\boldsymbol{x}^t)$ is supported on $\overline{S^t}$ and we simply docompose the support set $S^{t+1} \cup S^t$ into three mutually disjoint sets, and hence ζ_1 holds. To see why ζ_2 holds, we note that for any set $\Omega \subset S^t$, $\boldsymbol{b}_\Omega^{t+1} = \boldsymbol{x}_\Omega^t$. Hence, $\boldsymbol{b}_{S^{t+1}\cap S^t}^{t+1} = \boldsymbol{x}_{S^{t+1}\cap S^t}^t$. Moreover, since $\boldsymbol{x}_{S^t\backslash S^{t+1}}^t = \boldsymbol{b}_{S^t\backslash S^{t+1}}^{t+1}$ and any element in $\boldsymbol{b}_{S^t\backslash S^{t+1}}^{t+1}$ is not larger than that in $\boldsymbol{b}_{S^{t+1}\backslash S^t}^{t+1}$ (recall that S^{t+1} is obtained by hard thresholding), we have $\left\| \boldsymbol{x}_{S^t\backslash S^{t+1}}^t \right\| \leq \left\| \boldsymbol{b}_{S^{t+1}\backslash S^t}^{t+1} \right\|$ where we use the fact that $\left| S^t \backslash S^{t+1} \right| = \left| S^{t+1} \backslash S^t \right|$. Therefore, ζ_2 holds. Finally, we write $\boldsymbol{b}_{S^{t+1}\backslash S^t}^{t+1} = -\eta \nabla_{S^{t+1}\backslash S^t} F(\boldsymbol{x}^t)$ and obtain ζ_3.

Since $\boldsymbol{x}^{t+1}$ is a minimizer of $F(\boldsymbol{x})$ over the support set S^{t+1}, it immediately follows that

$$F(\boldsymbol{x}^{t+1}) - F(\boldsymbol{x}^t) \leq F(\boldsymbol{b}_{S^{t+1}}^{t+1}) - F(\boldsymbol{x}^t) \leq -\eta(1-\eta\rho^+) \left\| \nabla_{S^{t+1}\backslash S^t} F(\boldsymbol{x}^t) \right\|^2.$$

Now we invoke Lemma 2.21 and pick $\eta \leq 1/\rho^+$,

$$F(\boldsymbol{x}^{t+1}) - F(\boldsymbol{x}^t) \leq \eta(\eta\rho^+ - 1) \cdot \frac{2\rho^-}{1+s} \left(F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) \right),$$

which gives

$$F(\boldsymbol{x}^{t+1}) - F(\bar{\boldsymbol{x}}) \leq \mu \left(F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) \right),$$

where $\mu = 1 - \frac{2\rho^- \eta(1-\eta\rho^+)}{1+s}$. □

2.B.2 Proof of Proposition 2.2

Proof. This is a direct result by combining Proposition 2.1 and Proposition 2.19. □

2.B.3 Proof of Lemma 2.3

Proof. Let $\boldsymbol{x}_*^t = \arg\min_{\mathrm{supp}(\boldsymbol{x})\subset S^t} F(\boldsymbol{x})$. Since $\boldsymbol{x}^t$ and $\boldsymbol{x}_*^t$ are both supported on S^t, we apply Lemma 2.14 and obtain

$$\begin{aligned}\left\|\nabla_{S^t} F(\boldsymbol{x}^t)\right\|^2 &= \left\|\nabla_{S^t} F(\boldsymbol{x}^t) - \nabla_{S^t} F(\boldsymbol{x}_*^t)\right\|^2 \\ &\le 2\rho^+ \left(F(\boldsymbol{x}^t) - F(\boldsymbol{x}_*^t) - \left\langle \nabla F(\boldsymbol{x}_*^t), \boldsymbol{x}^t - \boldsymbol{x}_*^t \right\rangle\right) \\ &\le 2\rho^+ \epsilon.\end{aligned}$$

Above, the second inequality uses the fact that $\nabla_{S^t} F(\boldsymbol{x}_*^t) = 0$ and $F(\boldsymbol{x}^t) \le F(\boldsymbol{x}_*^t) + \epsilon$. □

2.B.4 Proof of Proposition 2.4

Proof. We have by Lemma 2.25 that

$$\left\|\bar{\boldsymbol{x}}_{\overline{S^{t+1}}}\right\| \le \sqrt{2}\rho \left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| + \frac{2}{\rho^-} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\|,$$

where $\rho = 1 - \eta\rho^-$. On the other hand, Lemma 2.20 together with Lemma 2.3 shows that

$$\left\|\boldsymbol{x}^{t+1} - \bar{\boldsymbol{x}}\right\| \le \kappa \left\|\bar{\boldsymbol{x}}_{\overline{S^{t+1}}}\right\| + \frac{1}{\rho^-} \left\|\nabla_k F(\bar{\boldsymbol{x}})\right\| + \frac{1}{\rho^-}\sqrt{2\rho^+\epsilon}.$$

Therefore,

$$\left\|\boldsymbol{x}^{t+1} - \bar{\boldsymbol{x}}\right\| \le \sqrt{2}\kappa\rho \left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| + \frac{3\kappa}{\rho^-} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\| + \frac{\sqrt{2\rho^+\epsilon}}{\rho^-}$$

We need to ensure

$$\sqrt{2}\kappa(1 - \eta\rho^-) < 1.$$

Let $\eta = \eta'/\rho^+$ with $\eta' < 1$. Then, the above holds provided that

$$\kappa < 1 + \frac{1}{\sqrt{2}} \text{ and } \eta' > \kappa - \frac{1}{\sqrt{2}}.$$

By induction and picking proper η' to make $\sqrt{2}\kappa(1-\eta\rho^-) < \sqrt{2}/4$, we have

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq (\sqrt{2}(\kappa - \eta'))^t \left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{6\kappa}{\rho^-} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\| + \frac{4\sqrt{\rho^+ \epsilon}}{\rho^-}.$$

This completes the proof. □

Lemma 2.25. *Let $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ be an s-sparse vector supported on S. For a k-sparse vector $\boldsymbol{x}$ supported on Q with $k \geq s$, let $\boldsymbol{b} = \boldsymbol{x} - \eta \nabla F(\boldsymbol{x})$ and let $T = \operatorname{supp}(\boldsymbol{b}, k)$. Suppose that the function $F(\boldsymbol{x})$ is ρ^-_{2k+s}-RSC and ρ^+_{2k+s}-RSS. Then we have*

$$\left\|\bar{\boldsymbol{x}}_{S\backslash T}\right\| \leq \nu \phi_{2k+s} \left\|\boldsymbol{x} - \bar{\boldsymbol{x}}\right\| + \nu\eta \left\|\nabla_{T\Delta S} F(\bar{\boldsymbol{x}})\right\|,$$

where $\nu = \sqrt{1 + s/k}$ and ϕ_{2k+s} is given by Lemma 2.16.

Proof. We note the fact that the support sets $T\backslash S$ and $S\backslash T$ are disjoint. Moreover, the set $T\backslash S$ contains $|T\backslash S|$ number of top $|T|$ elements of $\boldsymbol{b}$. Hence, we have

$$\frac{1}{|T\backslash S|} \left\|\boldsymbol{b}_{T\backslash S}\right\|^2 \geq \frac{1}{|S\backslash T|} \left\|\boldsymbol{b}_{S\backslash T}\right\|^2. \tag{2.24}$$

That is,

$$\left\|\boldsymbol{b}_{T\backslash S}\right\| \geq \sqrt{\frac{|T\backslash S|}{|S\backslash T|}} \left\|\boldsymbol{b}_{S\backslash T}\right\| = \sqrt{\frac{k - |T \cap S|}{s - |T \cap S|}} \left\|\boldsymbol{b}_{S\backslash T}\right\| \geq \sqrt{\frac{k}{s}} \left\|\boldsymbol{b}_{S\backslash T}\right\|.$$

Note that the above holds also for $T = S$. Since $\bar{\boldsymbol{x}}$ is supported on S, the left hand side reads as

$$\left\|\boldsymbol{b}_{T\backslash S}\right\| = \left\|(\boldsymbol{x} - \bar{\boldsymbol{x}} - \eta \nabla F(\boldsymbol{x}))_{T\backslash S}\right\|,$$

while the right hand side reads as

$$\begin{aligned}\left\|\boldsymbol{b}_{S\setminus T}\right\| &= \left\|(\boldsymbol{x}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}))_{S\setminus T}+\bar{\boldsymbol{x}}_{S\setminus T}\right\| \\ &\geq \left\|\bar{\boldsymbol{x}}_{S\setminus T}\right\| - \left\|(\boldsymbol{x}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}))_{S\setminus T}\right\|.\end{aligned}$$

Denote $\nu=\sqrt{1+s/k}$. In this way, we arrive at

$$\begin{aligned}\left\|\bar{\boldsymbol{x}}_{S\setminus T}\right\| &\leq \sqrt{\frac{s}{k}}\left\|(\boldsymbol{x}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}))_{T\setminus S}\right\| + \left\|(\boldsymbol{x}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}))_{S\setminus T}\right\| \\ &\leq \nu\left\|(\boldsymbol{x}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x}))_{T\Delta S}\right\| \\ &\leq \nu\left\|(\boldsymbol{x}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x})+\eta\nabla F(\bar{\boldsymbol{x}}))_{T\Delta S}\right\| + \nu\eta\left\|\nabla_{T\Delta S}F(\bar{\boldsymbol{x}})\right\| \\ &\leq \nu\left\|(\boldsymbol{x}-\bar{\boldsymbol{x}}-\eta\nabla F(\boldsymbol{x})+\eta\nabla F(\bar{\boldsymbol{x}}))_{T\cup Q\cup S}\right\| + \nu\eta\left\|\nabla_{T\Delta S}F(\bar{\boldsymbol{x}})\right\| \\ &\leq \nu\phi_{2k+s}\left\|\boldsymbol{x}-\bar{\boldsymbol{x}}\right\| + \nu\eta\left\|\nabla_{T\Delta S}F(\bar{\boldsymbol{x}})\right\|,\end{aligned}$$

where the second inequality follows from the fact that $ax+by\leq\sqrt{a^2+b^2}\sqrt{x^2+y^2}$ and we applied Lemma 2.17 for the last inequality. □

2.B.5 Proof of Proposition 2.5

Proof. Let $\boldsymbol{x}_*^t=\arg\min_{\mathrm{supp}(\boldsymbol{x})\subset S^t}F(\boldsymbol{x})$. Then

$$\begin{aligned}F(\boldsymbol{x}^t)-F(\boldsymbol{x}^{t-1}) &\leq F(\boldsymbol{x}_*^t)-F(\boldsymbol{x}^{t-1})+\epsilon \\ &\leq F(\boldsymbol{b}_{S^t}^t)-F(\boldsymbol{x}^{t-1})+\epsilon \\ &\leq -\frac{1-\eta\rho^+}{2\eta}\left\|\boldsymbol{b}_{S^t}^t-\boldsymbol{x}^{t-1}\right\|^2+\epsilon,\end{aligned}$$

where the last inequality follows from Lemma 2.26. Now we bound the term $\left\|\boldsymbol{b}^t_{S^t} - \boldsymbol{x}^{t-1}\right\|^2$. Note that $\boldsymbol{x}^{t-1}$ is supported on S^{t-1}. Hence,

$$\begin{aligned}\left\|\boldsymbol{b}^t_{S^t} - \boldsymbol{x}^{t-1}\right\|^2 &= \left\|\boldsymbol{x}^{t-1}_{S^t\cap S^{t-1}} - \eta\nabla_{S^t}F(\boldsymbol{x}^{t-1}) - \boldsymbol{x}^{t-1}\right\|^2 \\ &= \left\|-\boldsymbol{x}^{t-1}_{S^{t-1}\setminus S^t} - \eta\nabla_{S^t}F(\boldsymbol{x}^{t-1})\right\|^2 \\ &= \left\|\boldsymbol{x}^{t-1}_{S^{t-1}\setminus S^t}\right\|^2 + \eta^2\left\|\nabla_{S^t}F(\boldsymbol{x}^{t-1})\right\|^2 \\ &\geq \eta^2\left\|\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2.\end{aligned}$$

We thus have

$$F(\boldsymbol{x}^t) - F(\boldsymbol{x}^{t-1}) \leq -\frac{(1-\eta\rho^+)\eta}{2}\left\|\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2 + \epsilon.$$

Denote $\xi = \left\|\nabla_{S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|$. We claim that

$$\left\|\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2 \geq \rho^-\left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})\right) - 2\xi^2, \tag{2.25}$$

which, combined with Lemma 2.3, immediately shows

$$F(\boldsymbol{x}^t) - F(\boldsymbol{x}^{t-1}) \leq -\frac{(1-\eta\rho^+)\eta\rho^-}{2}\left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})\right) + 2\epsilon.$$

Using Proposition 2.19 completes the proof.

To show (2.25), we consider two exhausitive cases: $\left|S^t\setminus S^{t-1}\right| \geq s$ and $\left|S^t\setminus S^{t-1}\right| < s$, and prove that (2.25) holds for both cases.

Case I. $\left|S^t \backslash S^{t-1}\right| \geq s$. Due to the RSC property, we have

$$
\begin{aligned}
&\frac{\rho^-}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2 \\
&\leq F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})-\left\langle\nabla F(\boldsymbol{x}^{t-1}), \bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\rangle \\
&\leq F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{1}{2\rho^-}\left\|\nabla_{S\cup S^{t-1}} F(\boldsymbol{x}^{t-1})\right\|^2 \\
&= F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{1}{2\rho^-}\left\|\nabla_{S\setminus S^{t-1}} F(\boldsymbol{x}^{t-1})\right\|^2+\frac{1}{2\rho^-}\left\|\nabla_{S^{t-1}} F(\boldsymbol{x}^{t-1})\right\|^2 \\
&= F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{1}{2\rho^-}\left\|\nabla_{S\setminus S^{t-1}} F(\boldsymbol{x}^{t-1})\right\|^2+\frac{1}{2\rho^-}\xi^2.
\end{aligned}
$$

Therefore, we get

$$
\left\|\nabla_{S\setminus S^{t-1}} F(\boldsymbol{x}^{t-1})\right\|^2 \geq 2\rho^-\left(F(\boldsymbol{x}^{t-1})-F(\bar{\boldsymbol{x}})\right)-\xi^2.
$$

Since S^t contains the k largest absolute values of $\boldsymbol{b}^t$, and $\left|S^t \backslash S^{t-1}\right| \geq s \geq \left|S \backslash S^{t-1}\right|$, we have

$$
\left\|\boldsymbol{b}^t_{S^t\setminus S^{t-1}}\right\|^2 \geq \left\|\boldsymbol{b}^t_{S\setminus S^{t-1}}\right\|^2,
$$

which immediately implies (2.25) by noting the fact that $\boldsymbol{b}^t_{S^t\setminus S^{t-1}} = -\eta\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})$ and $\boldsymbol{b}^t_{S\setminus S^{t-1}} = -\eta\nabla_{S\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})$.

Case II. $\left|S^t \backslash S^{t-1}\right| < s$. Again, we use the RSC property to obtain

$$
\begin{aligned}
\frac{\rho^-}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2 &\leq F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})-\left\langle\nabla F(\boldsymbol{x}^{t-1}), \bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\rangle \\
&\leq F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{1}{\rho^-}\left\|\nabla_{S\cup S^{t-1}} F(\boldsymbol{x}^{t-1})\right\|^2 \\
&= F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{1}{\rho^-}\left\|\nabla_{S\setminus S^{t-1}} F(\boldsymbol{x}^{t-1})\right\|^2+\frac{1}{\rho^-}\xi^2 \\
&= F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{1}{\rho^-}\left\|\nabla_{S\setminus(S^t\cup S^{t-1})} F(\boldsymbol{x}^{t-1})\right\|^2 \\
&\quad+\frac{1}{\rho^-}\left\|\nabla_{(S^t\setminus S^{t-1})\cap S} F(\boldsymbol{x}^{t-1})\right\|^2+\frac{1}{\rho^-}\xi^2 \\
&\leq F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{1}{\rho^-}\left\|\nabla_{S\setminus(S^t\cup S^{t-1})} F(\boldsymbol{x}^{t-1})\right\|^2 \\
&\quad+\frac{1}{\rho^-}\left\|\nabla_{S^t\setminus S^{t-1}} F(\boldsymbol{x}^{t-1})\right\|^2+\frac{1}{\rho^-}\xi^2. \qquad (2.26)
\end{aligned}
$$

We consider the term $\left\|\nabla_{S\backslash(S^t\cup S^{t-1})}F(\boldsymbol{x}^{t-1})\right\|^2$ above. Actually, we have

$$\boldsymbol{b}^t_{S\backslash(S^t\cup S^{t-1})} = -\eta\nabla_{S\backslash(S^t\cup S^{t-1})}F(\boldsymbol{x}^{t-1}).$$

Since S^t contains the k largest absolute values of $\boldsymbol{b}^t$, we know that any component in $\boldsymbol{b}^t_\Omega$ is not larger than that in $\boldsymbol{b}^t_{S^t}$ subject to $\Omega\cap S^t=\emptyset$. In particular,

$$\frac{\left\|\boldsymbol{b}^t_{S\backslash(S^t\cup S^{t-1})}\right\|^2}{|S\backslash(S^t\cup S^{t-1})|}\le\frac{\left\|\boldsymbol{b}^t_{(S^t\cap S^{t-1})\backslash S}\right\|^2}{|(S^t\cap S^{t-1})\backslash S|}.$$

Note that $\left|S^t\backslash S^{t-1}\right|<s$ implies $\left|(S^t\cap S^{t-1})\backslash S\right|\ge k-2s$. Therefore,

$$\begin{aligned}\eta^2\left\|\nabla_{S\backslash(S^t\cup S^{t-1})}F(\boldsymbol{x}^{t-1})\right\|^2&\le\frac{s}{k-2s}\left\|\boldsymbol{x}^{t-1}_{(S^t\cap S^{t-1})\backslash S}-\eta\nabla_{(S^t\cap S^{t-1})\backslash S}F(\boldsymbol{x}^{t-1})\right\|^2\\&\le\frac{2s}{k-2s}\left\|\boldsymbol{x}^{t-1}_{(S^t\cap S^{t-1})\backslash S}\right\|^2+\frac{2s\eta^2}{k-2s}\xi^2\\&=\frac{2s}{k-2s}\left\|(\boldsymbol{x}^{t-1}-\bar{\boldsymbol{x}})_{(S^t\cap S^{t-1})\backslash S}\right\|^2+\frac{2s\eta^2}{k-2s}\xi^2\\&\le\frac{2s}{k-2s}\left\|\boldsymbol{x}^{t-1}-\bar{\boldsymbol{x}}\right\|^2+\frac{2s\eta^2}{k-2s}\xi^2.\end{aligned}$$

Plugging the above into (2.31), we obtain

$$\begin{aligned}\frac{\rho^-}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2\le F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{2s}{(k-2s)\eta^2\rho^-}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2\\+\frac{1}{\rho^-}\left\|\nabla_{S^t\backslash S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2+\frac{1}{\rho^-}\left(\frac{2s}{k-2s}+1\right)\xi^2.\end{aligned}$$

Picking $k\ge 2s+\frac{8s}{\eta^2m^2}$ gives

$$\begin{aligned}\frac{\rho^-}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2\le F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2\\+\frac{1}{\rho^-}\left\|\nabla_{S^t\backslash S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2+\left(\frac{\eta^2\rho^-}{4}+\frac{1}{\rho^-}\right)\xi^2.\end{aligned}$$

Since $\eta<1/\rho^+$, $\frac{(\eta\rho^-)^2}{4}+1<2$. Therefore, by re-arranging the above inequality, we prove the claim (2.25). □

Lemma 2.26. *Suppose that $\boldsymbol{x}$ is a k-sparse vector and let $\boldsymbol{b}=\boldsymbol{x}-\eta\nabla F(\boldsymbol{x})$. Let T be the support*

set that contains the k largest absolute values of $\boldsymbol{b}$. Assume that the function $F(\boldsymbol{x})$ is ρ_{2k}^{+}-restricted smooth, the we have the following:

$$F(\boldsymbol{b}_T) \leq F(\boldsymbol{x}) - \frac{1-\eta\rho_{2k}^{+}}{2\eta}\left\|\boldsymbol{b}_T - \boldsymbol{x}\right\|^2.$$

Proof. The RSS condition implies that

$$\begin{aligned} & F(\boldsymbol{b}_T) - F(\boldsymbol{x}) \\ \leq\ & \left\langle \nabla F(\boldsymbol{x}), \boldsymbol{b}_T - \boldsymbol{x}\right\rangle + \frac{\rho_{2k}^{+}}{2}\left\|\boldsymbol{b}_T - \boldsymbol{x}\right\|^2 \\ \leq\ & -\frac{1}{2\eta}\left\|\boldsymbol{b}_T - \boldsymbol{x}\right\|^2 + \frac{\rho_{2k}^{+}}{2}\left\|\boldsymbol{b}_T - \boldsymbol{x}\right\|^2, \end{aligned}$$

where the second inequality is due to the fact that

$$\begin{aligned} \left\|\boldsymbol{b}_T - \boldsymbol{b}\right\|^2 =\ & \left\|\boldsymbol{b}_T - \boldsymbol{x} + \eta\nabla F(\boldsymbol{x})\right\|^2 \\ \leq\ & \left\|\boldsymbol{x} - \boldsymbol{x} + \eta\nabla F(\boldsymbol{x})\right\|^2 \\ =\ & \left\|\eta\nabla F(\boldsymbol{x})\right\|^2, \end{aligned}$$

implying

$$2\eta\left\langle \nabla F(\boldsymbol{x}), \boldsymbol{b}_T - \boldsymbol{x}\right\rangle \leq -\left\|\boldsymbol{b}_T - \boldsymbol{x}\right\|^2.$$

This completes the proof. □

2.B.6 Proof of Theorem 2.7

In view of the exact (HTP3), we have by Lemma 2.20

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \kappa\left\|\bar{\boldsymbol{x}}_{\overline{S^t}}\right\| + \frac{1}{\rho^{-}}\left\|\nabla_k F(\bar{\boldsymbol{x}})\right\|.$$

With this observation, Lemma 2.23, Theorem 2.24, and specific result in Proposition 2.2, Theorem 2.7 follows immediately.

2.B.7 Proof of Theorem 2.8

The theorem follows from Lemma 2.23, Theorem 2.24, the specific result in Proposition 2.4 and Lemma 2.3.

2.B.8 Proof of Theorem 2.9

The theorem follows from Lemma 2.23, Theorem 2.24, the specific result in Proposition 2.5 and Lemma 2.3.

2.C Proofs for Section 2.3

This section generalizes the results of HTP, and present a more principled theoretical analysis that uncovers HTP and OMPR.

2.C.1 Proof of Proposition 2.11

Proof. Recall that we set $k = s$. Using the RSS property, we have

$$\begin{aligned}
F(\boldsymbol{z}^t_{S^t}) - F(\boldsymbol{x}^{t-1}) &\leq \left\langle \nabla F(\boldsymbol{x}^{t-1}), \boldsymbol{z}^t_{S^t} - \boldsymbol{x}^{t-1} \right\rangle + \frac{\rho^+_{2s}}{2} \left\| \boldsymbol{z}^t_{S^t} - \boldsymbol{x}^{t-1} \right\|^2 \\
&\overset{\zeta_1}{=} \left\langle \nabla_{S^t \setminus S^{t-1}} F(\boldsymbol{x}^{t-1}), \boldsymbol{z}^t_{S^t \setminus S^{t-1}} \right\rangle + \frac{\rho^+_{2s}}{2} \Big(\left\| \boldsymbol{z}^t_{S^t \setminus S^{t-1}} \right\|^2 \\
&\quad + \left\| \boldsymbol{z}^t_{S^t \cap S^{t-1}} - \boldsymbol{x}^{t-1}_{S^t \cap S^{t-1}} \right\|^2 + \left\| \boldsymbol{x}^{t-1}_{S^{t-1} \setminus S^t} \right\|^2 \Big) \\
&\overset{\zeta_2}{\leq} \left\langle \nabla_{S^t \setminus S^{t-1}} F(\boldsymbol{x}^{t-1}), \boldsymbol{z}^t_{S^t \setminus S^{t-1}} \right\rangle + \rho^+_{2s} \left\| \boldsymbol{z}^t_{S^t \setminus S^{t-1}} \right\|^2 \\
&\overset{\zeta_3}{=} -\eta(1 - \eta\rho^+_{2s}) \left\| \nabla_{S^t \setminus S^{t-1}} F(\boldsymbol{x}^{t-1}) \right\|^2 .
\end{aligned}$$

Above, we observe that $\nabla F(\boldsymbol{x}^{t-1})$ is supported on $\overline{S^{t-1}}$ and we simply docompose the support set $S^t \cup S^{t-1}$ into three mutually disjoint sets, and hence ζ_1 holds. To see why ζ_2 holds, we note that for any set $\Omega \subset S^{t-1}$, $\boldsymbol{z}^t_\Omega = \boldsymbol{x}^{t-1}_\Omega$. Hence, $\boldsymbol{z}^t_{S^t \cap S^{t-1}} = \boldsymbol{x}^{t-1}_{S^t \cap S^{t-1}}$. Moreover, since $\boldsymbol{x}^{t-1}_{S^{t-1} \setminus S^t} = \boldsymbol{z}^t_{S^{t-1} \setminus S^t}$ and any element in $\boldsymbol{z}^t_{S^{t-1} \setminus S^t}$ is not larger than that in $\boldsymbol{z}^t_{S^t \setminus S^{t-1}}$ (recall that S^t is obtained by hard thresholding), we have $\left\| \boldsymbol{x}^{t-1}_{S^{t-1} \setminus S^t} \right\| \leq \left\| \boldsymbol{z}^t_{S^t \setminus S^{t-1}} \right\|$ where we use the fact that $\left| S^t \setminus S^t \right| = \left| S^t \setminus S^{t-1} \right|$. Therefore, ζ_2 holds. Finally, we write $\boldsymbol{z}^t_{S^t \setminus S^{t-1}} = -\eta \nabla_{S^t \setminus S^{t-1}} F(\boldsymbol{x}^{t-1})$ and obtain ζ_3.

Since $\boldsymbol{x}^t$ is a minimizer of $F(\boldsymbol{x})$ over the support set S^t, it immediately follows that

$$F(\boldsymbol{x}^t) - F(\boldsymbol{x}^{t-1}) \leq F(\boldsymbol{z}^t_{S^t}) - F(\boldsymbol{x}^{t-1}) \leq -\eta(1-\eta\rho^+_{2s}) \left\|\nabla_{S^t\setminus S^{t-1}} F(\boldsymbol{x}^{t-1})\right\|^2.$$

Now we invoke Lemma 2.21 and pick $\eta \leq 1/\rho^+_{2s}$,

$$F(\boldsymbol{x}^t) - F(\boldsymbol{x}^{t-1}) \leq -2m\eta(1-\eta\rho^+_{2s}) \cdot \frac{\left|S^t\setminus S^{t-1}\right|}{|S^t\setminus S^{t-1}| + |S\setminus S^{t-1}|} \left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})\right),$$

which gives

$$F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) \leq \mu_t \left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})\right),$$

where $\mu_t = 1 - 2\eta\rho^-_{2s}(1-\eta\rho^+_{2s}) \cdot \frac{|S^t\setminus S^{t-1}|}{|S^t\setminus S^{t-1}|+|S\setminus S^{t-1}|}$. Now combining this with Prop. 2.19, we have

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \sqrt{2\kappa}\sqrt{\mu_1\mu_2\ldots\mu_t}\left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{3}{\rho^-_{2s}}\left\|\nabla_{2s}F(\bar{\boldsymbol{x}})\right\|.$$

Note that before the algorithm terminates, $1 \leq \left|S^t\setminus S^{t-1}\right| \leq r$. Hence,

$$\mu_t \leq 1 - \frac{2\eta\rho^-_{2s}(1-\eta\rho^+_{2s})}{1+s} =: \mu.$$

It then follows that

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \sqrt{2\kappa}(\sqrt{\mu})^t\left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{3}{\eta}\left\|\nabla_{2s}F(\bar{\boldsymbol{x}})\right\|. \tag{2.27}$$

Lemma 2.20 tells us

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \kappa\left\|\bar{\boldsymbol{x}}_{\overline{S^t}}\right\| + \frac{1}{\eta}\left\|\nabla_s F(\bar{\boldsymbol{x}})\right\|. \tag{2.28}$$

Hence, in light of Lemma 2.23 and Theorem 2.24, we obtain that PHT(r) recovers the support using at most

$$t_{\max} = \left(\frac{\log 2}{\log(1/\mu)} + \frac{\log(2\kappa)}{\log(1/\mu)} + \frac{2\log(\kappa/(1-\lambda))}{\log(1/\mu)} + 2\right)\|\bar{\boldsymbol{x}}\|_0$$

iterations. Note that picking $\eta = O(1/\rho^+_{2s})$, we have $\mu = O(1 - \frac{1}{\kappa})$ and $\log(1/\mu) = O(1/\kappa)$. This gives the $O(s\kappa \log \kappa)$ bound. □

2.C.2 Proof of Theorem 2.12

Proof. Let $\rho^- := \rho^-_{2s+r}$ and $\rho^+ := \rho^+_{2s+r}$. Let $\phi := \phi_{2s+r} = 1 - \eta\rho^-$ be the quantity given in Lemma 2.16. Using Lemma 2.27, we obtain

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \left(\sqrt{2}\phi\kappa + \nu(\kappa - 1)\right) \left\|\boldsymbol{x}^{t-1} - \bar{\boldsymbol{x}}\right\| + \frac{3\sqrt{s} + 4}{\rho^-} \left\|\nabla_{s+r} F(\bar{\boldsymbol{x}})\right\|,$$

where $\nu = \sqrt{1 + s/r}$. We need to ensure that the convergence coefficient is smaller than 1. Consider $\eta = \eta'/\rho^+$ with $\eta' \in (0, 1]$ for which $\phi = 1 - \eta'/\kappa$. It follows that

$$\sqrt{2}\phi\kappa + \nu(\kappa - 1) = \sqrt{2}(\kappa - \eta') + \nu(\kappa - 1) \leq (\sqrt{2} + \nu)(\kappa - \eta').$$

Hence, when we pick $1 - \frac{1}{\sqrt{2}+\nu} < \eta' \leq 1$, and the condition number satisfies

$$\kappa < \eta' + \frac{1}{\sqrt{2} + \nu},$$

the sequence of $\boldsymbol{x}^t - \bar{\boldsymbol{x}}$ contracts. On the other hand, using Lemma 2.20 we get

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \kappa \left\|\bar{\boldsymbol{x}}_{\overline{S^t}}\right\| + \frac{1}{\rho^-} \left\|\nabla_s F(\bar{\boldsymbol{x}})\right\|.$$

Hence, applying Lemma 2.23 and Theorem 2.24 we obtain the result. □

Lemma 2.27. *Consider the PHT(r) algorithm with $k = s$. Suppose that $F(\boldsymbol{x})$ is ρ^-_{2s+r}-RSC and ρ^+_{2s+r}-RSS. Further suppose that $\kappa < 2$. Let the step size $\eta \leq 1/\rho^+_{2s+r}$. Then it holds that*

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \left(\sqrt{2}\phi\kappa + \nu(\kappa - 1)\right) \left\|\boldsymbol{x}^{t-1} - \bar{\boldsymbol{x}}\right\| + \frac{3\sqrt{s} + 4}{\rho^-_{2s+r}} \left\|\nabla_{s+r} F(\bar{\boldsymbol{x}})\right\|,$$

where $\phi = 1 - \eta\rho^-_{2s+r}$ and $\nu = \sqrt{1 + s/r}$.

Proof. Consider the vector $\boldsymbol{z}^t_{J^t}$. It is easy to see that $J^t \backslash S^t$ contains the r smallest elements of $\boldsymbol{z}^t_{J^t}$.

Hence, for any subset $T \subset J^t$ such that $|T| \geq r$, we have

$$\left\| \boldsymbol{z}^t_{J^t \backslash S^t} \right\| \leq \left\| \boldsymbol{z}^t_T \right\|.$$

In particular, we choose $T = J^t \backslash S$ and obtain

$$\left\| \boldsymbol{z}^t_{J^t \backslash S^t} \right\| \leq \left\| \boldsymbol{z}^t_{J^t \backslash S} \right\|.$$

Eliminating the common contribution from $J^t \backslash (S^t \cup S)$ gives

$$\left\| \boldsymbol{z}^t_{J^t \cap S \backslash S^t} \right\| \leq \left\| \boldsymbol{z}^t_{J^t \cap S^t \backslash S} \right\|. \tag{2.29}$$

The LHS of (2.29) reads as

$$\begin{aligned} \left\| \boldsymbol{z}^t_{J^t \cap S \backslash S^t} \right\| &= \left\| (\boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}) - \bar{\boldsymbol{x}})_{J^t \cap S \backslash S^t} + \bar{\boldsymbol{x}}_{J^t \backslash S^t} \right\| \\ &\geq \left\| \bar{\boldsymbol{x}}_{J^t \backslash S^t} \right\| - \left\| (\boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}) - \bar{\boldsymbol{x}})_{J^t \cap S \backslash S^t} \right\|, \end{aligned}$$

while the RHS (2.29) is given by

$$\left\| \boldsymbol{z}^t_{J^t \cap S^t \backslash S} \right\| = \left\| (\boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}) - \bar{\boldsymbol{x}})_{J^t \cap S^t \backslash S} \right\|.$$

Hence, we have

$$\begin{aligned} \left\| \bar{\boldsymbol{x}}_{J^t \backslash S^t} \right\| &\leq \left\| (\boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}) - \bar{\boldsymbol{x}})_{J^t \cap S \backslash S^t} \right\| + \left\| (\boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}) - \bar{\boldsymbol{x}})_{J^t \cap S^t \backslash S} \right\| \\ &\leq \sqrt{2} \left\| (\boldsymbol{x}^{t-1} - \eta \nabla F(\boldsymbol{x}^{t-1}) - \bar{\boldsymbol{x}})_{J^t} \right\| \\ &\leq \sqrt{2} \phi_{2s+r} \left\| \boldsymbol{x}^{t-1} - \bar{\boldsymbol{x}} \right\| + \sqrt{2} \eta \left\| \nabla_{k+r} F(\bar{\boldsymbol{x}}) \right\|, \end{aligned}$$

where we use Lemma 2.17 for the last inequality and $\phi_{2s+r} = 1 - \eta \rho^-_{2s+r}$ for $\eta \leq 1/\rho^+_{2s+r}$. On the other hand, Lemma 2.22 shows that

$$\left\| \bar{\boldsymbol{x}}_{\overline{J^t}} \right\| \leq \nu \left(1 - \frac{1}{\kappa} \right) \left\| \boldsymbol{x}^{t-1} - \bar{\boldsymbol{x}} \right\| + \frac{\nu}{\rho^-_{2s+r}} \left\| \nabla_{s+r} F(\bar{\boldsymbol{x}}) \right\|,$$

where $\nu = \sqrt{1 + s/r}$. The fact $\overline{S^t} = (J^t \backslash S^t) \cup \overline{J^t}$ implies

$$\begin{aligned}\left\|\bar{\boldsymbol{x}}_{\overline{S^t}}\right\| &\leq \left\|\bar{\boldsymbol{x}}_{J^t \backslash S^t}\right\| + \left\|\bar{\boldsymbol{x}}_{\overline{J^t}}\right\| \\ &\leq \left(\sqrt{2}\phi_{2s+r} + \nu\left(1 - \frac{1}{\kappa}\right)\right)\left\|\boldsymbol{x}^{t-1} - \bar{\boldsymbol{x}}\right\| + \left(\sqrt{2}\eta + \frac{\nu}{\rho^-_{2s+r}}\right)\left\|\nabla_{k+r} F(\bar{\boldsymbol{x}})\right\|.\end{aligned}$$

Next, we invoke Lemma 2.20 to get

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \kappa \left\|\bar{\boldsymbol{x}}_{\overline{S^t}}\right\| + \frac{1}{\rho^-_{2s+r}}\left\|\nabla_k F(\bar{\boldsymbol{x}})\right\|.$$

Therefore,

$$\begin{aligned}\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| &\leq \left(\sqrt{2}\phi_{2s+r}\kappa + \nu(\kappa - 1)\right)\left\|\boldsymbol{x}^{t-1} - \bar{\boldsymbol{x}}\right\| + \left(\sqrt{2}\eta\kappa + \frac{\nu\kappa}{\rho^-_{2s+r}} + \frac{1}{\rho^-_{2s+r}}\right)\left\|\nabla_{s+r} F(\bar{\boldsymbol{x}})\right\| \\ &\leq \left(\sqrt{2}\phi_{2s+r}\kappa + \nu(\kappa - 1)\right)\left\|\boldsymbol{x}^{t-1} - \bar{\boldsymbol{x}}\right\| + \frac{2\sqrt{1+s} + 4}{\rho^-_{2s+r}}\left\|\nabla_{s+r} F(\bar{\boldsymbol{x}})\right\|,\end{aligned}$$

where we use the assumption that $\kappa < 2$, the fact $\nu \leq \sqrt{1+s}$ and $\eta \leq 1/\rho^+_{2s+r} < 1/\rho^-_{2s+r}$ for the last inequality. The result follows by noting $2\sqrt{1+s} < 3\sqrt{s}$. □

2.C.3 Proof of Theorem 2.13

Proof. Using Lemma 2.28, we have

$$F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) \leq \mu\left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})\right),$$

where

$$\mu = 1 - \frac{\eta\rho^-_{2k}(1 - \eta\rho^+_{2k})}{2}.$$

Now Prop. 2.19 suggests that

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \sqrt{2\kappa}\left(\sqrt{\mu}\right)^t \left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{3}{\rho^-_{2k}}\left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\|,$$

and Lemma 2.20 implies

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \kappa \left\|\bar{\boldsymbol{x}}_{\overline{S^t}}\right\| + \frac{1}{\rho_{2k}^-} \left\|\nabla_k F(\bar{\boldsymbol{x}})\right\|.$$

Combining these with Lemma 2.23 and Theorem 2.24 we complete the proof. □

Lemma 2.28. *Consider the PHT(r) algorithm. Suppose that $F(\boldsymbol{x})$ is ρ_{2k}^--RSC and ρ_{2k}^+-RSS, and let $\kappa = \rho_{2k}^+/\rho_{2k}^-$ be the condition number. Picking the step size $0 < \eta < 1/\rho_{2k}^+$ and the sparsity parameter $k \geq s + \left(1 + \frac{4}{\eta^2(\rho_{2k}^-)^2}\right)\min\{r, s\}$, then we have*

$$F(\boldsymbol{x}^t) - F(\boldsymbol{x}^{t-1}) \leq -\frac{\eta\rho_{2k}^-(1 - \eta\rho_{2k}^+)}{2}\left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})\right).$$

Proof. Using Lemma 2.29 we obtain

$$F(\boldsymbol{x}^t) - F(\boldsymbol{x}^{t-1}) \leq -\frac{1 - \eta\rho_{2k}^+}{2\eta}\left\|\boldsymbol{z}_{S^t}^t - \boldsymbol{x}^{t-1}\right\|^2.$$

Note that for the right-hand side, we may expand it as follows:

$$\begin{aligned}
\left\|\boldsymbol{z}_{S^t}^t - \boldsymbol{x}^{t-1}\right\|^2 &= \left\|\boldsymbol{x}_{S^t}^{t-1} - \boldsymbol{x}^{t-1} - \eta\nabla_{S^t}F(\boldsymbol{x}^{t-1})\right\|^2 \\
&= \left\|-\boldsymbol{x}_{S^{t-1}\setminus S^t}^{t-1} - \eta\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2 \\
&= \left\|\boldsymbol{x}_{S^{t-1}\setminus S^t}^{t-1}\right\|^2 + \eta^2\left\|\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2,
\end{aligned}$$

where we use the fact that $\boldsymbol{x}^{t-1}$ is supported on S^{t-1} and $\nabla F(\boldsymbol{x}^{t-1})$ is support on $\overline{S^{t-1}}$ for the second equality, and the third one follows in that the support sets are disjoint. It then follows quickly that

$$F(\boldsymbol{x}^t) - F(\boldsymbol{x}^{t-1}) \leq -\frac{(1 - \eta\rho_{2k}^+)\eta}{2}\left\|\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2.$$

It remains to lower bound the right-hand side in terms of $F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})$. In fact, in the following,

we show that

$$\left\|\nabla_{S^t\backslash S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2 \geq \rho_{2k}^{-}\left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})\right). \tag{2.30}$$

This suggests

$$F(\boldsymbol{x}^t) - F(\boldsymbol{x}^{t-1}) \leq -\frac{\eta\rho_{2k}^{-}(1-\eta\rho_{2k}^{+})}{2}\left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})\right)$$

which completes the proof. In the sequel, we prove the inequality (2.30) by discussing the size of the support set $S^t\backslash S^{t-1}$.

First, we consider $r \geq s$. Then it is possible that $\left|S^t\backslash S^{t-1}\right| \geq s$.

Case 1. $\left|S^t\backslash S^{t-1}\right| \geq s$. Using the RSC property, we have

$$\begin{aligned}
&\frac{\rho_{2k}^{-}}{2}\left\|\bar{\boldsymbol{x}} - \boldsymbol{x}^{t-1}\right\|^2 \\
&\leq F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^{t-1}) - \left\langle\nabla F(\boldsymbol{x}^{t-1}), \bar{\boldsymbol{x}} - \boldsymbol{x}^{t-1}\right\rangle \\
&\leq F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^{t-1}) + \frac{\rho_{2k}^{-}}{2}\left\|\bar{\boldsymbol{x}} - \boldsymbol{x}^{t-1}\right\|^2 + \frac{1}{2\rho_{2k}^{-}}\left\|\nabla_{S\cup S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2 \\
&= F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^{t-1}) + \frac{\rho_{2k}^{-}}{2}\left\|\bar{\boldsymbol{x}} - \boldsymbol{x}^{t-1}\right\|^2 + \frac{1}{2\rho_{2k}^{-}}\left\|\nabla_{S\backslash S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2.
\end{aligned}$$

Therefore, we get

$$\left\|\nabla_{S\backslash S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2 \geq 2\rho_{2k}^{-}\left(F(\boldsymbol{x}^{t-1}) - F(\bar{\boldsymbol{x}})\right).$$

Recall that $S^t\backslash S^{t-1}$ contains the largest elements of $\boldsymbol{z}^t_{\overline{S^{t-1}}}$. Hence, for any support set $T \subset \overline{S^{t-1}}$ with $|T| \leq \left|S^t\backslash S^{t-1}\right|$, we have

$$\left\|\boldsymbol{z}^t_T\right\| \leq \left\|\boldsymbol{z}^t_{S^t\backslash S^{t-1}}\right\|.$$

In particular, we can choose $T = S\backslash S^{t-1}$ as we assumed that $\left|S^t\backslash S^{t-1}\right| \geq s \geq |T|$. Then it holds

that

$$\left\|\boldsymbol{z}^t_{S^t\backslash S^{t-1}}\right\|^2 \geq \left\|\boldsymbol{z}^t_{S\backslash S^{t-1}}\right\|^2.$$

Note that for the left-hand side, $\boldsymbol{z}^t_{S^t\backslash S^{t-1}} = -\eta\nabla_{S^t\backslash S^{t-1}}F(\boldsymbol{x}^{t-1})$ while for the right-hand side, it is exactly equal to $-\eta\nabla_{S\backslash S^{t-1}}F(\boldsymbol{x}^{t-1})$. This completes the proof of the first case.

Case 2. $\left|S^t\backslash S^{t-1}\right| < s \leq r$. The proof of this part is more involved. We still begin with the RSC property, which gives

$$\begin{aligned}
\frac{\bar{\rho}_{2k}}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2 &\leq F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^{t-1}) - \left\langle\nabla F(\boldsymbol{x}^{t-1}), \bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\rangle \\
&\leq F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^{t-1}) + \frac{\bar{\rho}_{2k}}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2 + \frac{1}{\bar{\rho}_{2k}}\left\|\nabla_{S\cup S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2 \\
&= F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^{t-1}) + \frac{\bar{\rho}_{2k}}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2 + \frac{1}{\bar{\rho}_{2k}}\left\|\nabla_{S\backslash S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2 \\
&= F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^{t-1}) + \frac{\bar{\rho}_{2k}}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2 + \frac{1}{\bar{\rho}_{2k}}\left\|\nabla_{S\backslash(S^t\cup S^{t-1})}F(\boldsymbol{x}^{t-1})\right\|^2 \\
&\quad + \frac{1}{\bar{\rho}_{2k}}\left\|\nabla_{(S^t\backslash S^{t-1})\cap S}F(\boldsymbol{x}^{t-1})\right\|^2 \\
&\leq F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^{t-1}) + \frac{\bar{\rho}_{2k}}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2 + \frac{1}{\bar{\rho}_{2k}}\left\|\nabla_{S\backslash(S^t\cup S^{t-1})}F(\boldsymbol{x}^{t-1})\right\|^2 \\
&\quad + \frac{1}{\bar{\rho}_{2k}}\left\|\nabla_{S^t\backslash S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2. \qquad (2.31)
\end{aligned}$$

Note that the last term is retained for deduction. What we need to show is a proper bound of the term $\left\|\nabla_{S\backslash(S^t\cup S^{t-1})}F(\boldsymbol{x}^{t-1})\right\|^2$ above. First, we observe that

$$\boldsymbol{z}^t_{S\backslash(S^t\cup S^{t-1})} = -\eta\nabla_{S\backslash(S^t\cup S^{t-1})}F(\boldsymbol{x}^{t-1}).$$

Next, we compare the elements of $S\backslash(S^t\cup S^{t-1})$ to those in $(S^t\cap S^{t-1})\backslash S$. For convenience, we denote $T = J^t\backslash(S^{t-1}\cup S^t)$. Since S^t contains the k largest elements of $\boldsymbol{z}^t_{J^t}$, those of $(S^t\cap S^{t-1})\backslash S$ are larger than those in T. On the other hand, recall that elements in $J^t\backslash S^{t-1}$ are larger than those in $\overline{J^t}$ due to the partial hard thresholding. Since T is a subset of $J^t\backslash S^{t-1}$, we have that T is larger than $\overline{J^t}$. Consequently, elements in $(S^t\cap S^{t-1})\backslash S$ are larger than those in $T\cup\overline{J^t} = \overline{S^{t-1}\cup S^t}$.

This suggests that

$$\frac{\left\|\boldsymbol{z}^t_{S\setminus(S^t\cup S^{t-1})}\right\|^2}{|S\setminus(S^t\cup S^{t-1})|}\leq\frac{\left\|\boldsymbol{z}^t_{(S^t\cap S^{t-1})\setminus S}\right\|^2}{|(S^t\cap S^{t-1})\setminus S|}.$$

Note that $\left|S^t\setminus S^{t-1}\right|<s$ implies $\left|(S^t\cap S^{t-1})\setminus S\right|\geq k-2s$. Therefore,

$$\begin{aligned}\eta^2\left\|\nabla_{S\setminus(S^t\cup S^{t-1})}F(\boldsymbol{x}^{t-1})\right\|^2&\leq\frac{s}{k-2s}\left\|\boldsymbol{x}^{t-1}_{(S^t\cap S^{t-1})\setminus S}-\eta\nabla_{(S^t\cap S^{t-1})\setminus S}F(\boldsymbol{x}^{t-1})\right\|^2\\&=\frac{s}{k-2s}\left\|\boldsymbol{x}^{t-1}_{(S^t\cap S^{t-1})\setminus S}\right\|^2\\&=\frac{s}{k-2s}\left\|(\boldsymbol{x}^{t-1}-\bar{\boldsymbol{x}})_{(S^t\cap S^{t-1})\setminus S}\right\|^2\\&\leq\frac{s}{k-2s}\left\|\boldsymbol{x}^{t-1}-\bar{\boldsymbol{x}}\right\|^2.\end{aligned}$$

Plugging the above into (2.31), we obtain

$$\begin{aligned}\frac{\rho^-_{2k}}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2\leq F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-_{2k}}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{s}{(k-2s)\eta^2\rho^-_{2k}}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2\\+\frac{1}{\rho^-_{2k}}\left\|\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2.\end{aligned}$$

Picking $k\geq 2s+\frac{4s}{\eta^2(\rho^-_{2k})^2}$ gives

$$\frac{\rho^-_{2k}}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2\leq F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho^-_{2k}}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{1}{\rho^-_{2k}}\left\|\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2,$$

which is exactly the claim (2.30).

Now we consider the parameter setting $r<s$. In this case, $\left|S^t\setminus S^{t-1}\right|$ cannot be greater than s. In fact, like we have done for Case 2, we can show that

$$\eta^2\left\|\nabla_{S\setminus(S^t\cup S^{t-1})}F(\boldsymbol{x}^{t-1})\right\|^2\leq\frac{r}{k-r-s}\left\|\boldsymbol{x}^{t-1}-\bar{\boldsymbol{x}}\right\|^2.$$

Plugging the above into (2.31), we obtain

$$\frac{\rho_{2k}^-}{2}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2 \le F(\bar{\boldsymbol{x}})-F(\boldsymbol{x}^{t-1})+\frac{\rho_{2k}^-}{4}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2+\frac{r}{(k-r-s)\eta^2\rho_{2k}^-}\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{t-1}\right\|^2 + \frac{1}{\rho_{2k}^-}\left\|\nabla_{S^t\setminus S^{t-1}}F(\boldsymbol{x}^{t-1})\right\|^2.$$

Using $k \ge s+r+\frac{4r}{\eta^2(\rho_{2k}^-)^2}$ we prove (2.30).

Overall, we find that picking $k \ge s+\left(1+\frac{4}{\eta^2(\rho_{2k}^-)^2}\right)\min\{r,s\}$ always guarantees the result.

□

Lemma 2.29. *Consider the PHT(r) algorithm. Suppose that $F(\boldsymbol{x})$ is ρ_{2k}^+-RSS. We have*

$$F(\boldsymbol{x}^t)-F(\boldsymbol{x}^{t-1}) \le -\frac{1-\eta\rho_{2k}^+}{2\eta}\left\|\boldsymbol{z}_{S^t}^t-\boldsymbol{x}^{t-1}\right\|^2.$$

Proof. We partition $\boldsymbol{z}^t$ into four disjoint parts: $S^{t-1}\setminus S^t$, $S^{t-1}\cap S^t$, $S^t\setminus S^{t-1}$ and $\overline{J^t}$. It then follows that

$$\begin{aligned}\left\|\boldsymbol{z}_{S^t}^t-\boldsymbol{z}^t\right\|^2 &= \left\|\boldsymbol{z}_{S^{t-1}\setminus S^t}^t\right\|^2+\left\|\boldsymbol{z}_{\overline{J^t}}^t\right\|^2\\ &\le \left\|\boldsymbol{z}_{S^t\setminus S^{t-1}}^t\right\|^2+\left\|\boldsymbol{z}_{\overline{J^t}}^t\right\|^2\\ &= \left\|\boldsymbol{z}_{\overline{S^{t-1}}}^t\right\|^2\\ &= \eta^2\left\|\nabla F(\boldsymbol{x}^{t-1})\right\|^2.\end{aligned}$$

On the other hand, the LHS reads as

$$\begin{aligned}\left\|\boldsymbol{z}_{S^t}^t-\boldsymbol{z}^t\right\|^2 &= \left\|\boldsymbol{z}_{S^t}^t-\boldsymbol{x}^{t-1}+\eta\nabla F(\boldsymbol{x}^{t-1})\right\|^2\\ &= \left\|\boldsymbol{z}_{S^t}^t-\boldsymbol{x}^{t-1}\right\|^2+\eta^2\left\|\nabla F(\boldsymbol{x}^{t-1})\right\|^2+2\eta\left\langle\nabla F(\boldsymbol{x}^{t-1}),\boldsymbol{z}_{S^t}^t-\boldsymbol{x}^{t-1}\right\rangle.\end{aligned}$$

Hence,

$$\left\langle\nabla F(\boldsymbol{x}^{t-1}),\boldsymbol{z}_{S^t}^t-\boldsymbol{x}^{t-1}\right\rangle \le -\frac{1}{2\eta}\left\|\boldsymbol{z}_{S^t}^t-\boldsymbol{x}^{t-1}\right\|^2.$$

Using the RSS property, we have

$$\begin{aligned} F(\boldsymbol{x}^t) - F(\boldsymbol{x}^{t-1}) &\leq F(\boldsymbol{y}^t) - F(\boldsymbol{x}^{t-1}) \\ &= F(\boldsymbol{z}_{S^t}^t) - F(\boldsymbol{x}^{t-1}) \\ &\leq \left\langle \nabla F(\boldsymbol{x}^{t-1}), \boldsymbol{z}_{S^t}^t - \boldsymbol{x}^{t-1} \right\rangle + \frac{\rho_{2k}^+}{2} \left\| \boldsymbol{z}_{S^t}^t - \boldsymbol{x}^{t-1} \right\|^2 \\ &\leq -\frac{1 - \eta\rho_{2k}^+}{2\eta} \left\| \boldsymbol{z}_{S^t}^t - \boldsymbol{x}^{t-1} \right\|^2 . \end{aligned}$$

This completes the proof. □

Chapter 3

Learning Sparse Models with Stochastic Optimization

3.1 Background

In this chapter, we are interested in the hard thresholding (HT) operator underlying a large body of the developed algorithms in compressed sensing (e.g., IHT, CoSaMP, SP), machine learning [160], and statistics [97]. Our motivation is two-fold. From a high level, compared to the convex programs, these HT-based algorithms are always orders of magnitude computationally more efficient, hence more practical for large-scale problems [143]. Nevertheless, they usually require a more stringent condition to guarantee the success. This naturally raises an interesting question of whether we can derive milder conditions for HT-based algorithms to achieve the best of the two worlds. For practitioners, to address the huge volume of data, a popular strategy in machine learning is to appeal to stochastic algorithms that sequentially update the solution. However, as many researchers observed [85, 54, 151], it is hard for the ℓ_1-based stochastic algorithms to preserve the sparse structure of the solution as the batch solvers do. This immediately poses the question of whether we are able to apply the principal idea of hard thresholding to stochastic algorithms while still ensuring a fast convergence.

To elaborate the problem more precisely, let us first turn to some basic properties of hard thresholding along with simple yet illustrative cases. For a general vector $\boldsymbol{b} \in \mathbb{R}^d$, the hard thresholded

signal $\mathcal{H}_k(\boldsymbol{b})$ is formed by setting all but the largest (in magnitude) k elements of $\boldsymbol{b}$ to zero. Ties are broken lexicographically. Hence, the hard thresholded signal $\mathcal{H}_k(\boldsymbol{b})$ is always k-sparse, i.e., the number of non-zero components does not exceed k. Moreover, the resultant signal $\mathcal{H}_k(\boldsymbol{b})$ is a best k-sparse approximation to $\boldsymbol{b}$ in terms of any ℓ_p norm ($p \geq 1$). That is, for any k-sparse vector $\boldsymbol{x}$

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b}\|_p \leq \|\boldsymbol{x} - \boldsymbol{b}\|_p.$$

In view of the above inequality, a broadly used bound in the literature for the deviation of the thresholded signal is as follows:

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\| \leq 2\|\boldsymbol{b} - \boldsymbol{x}\|. \tag{3.1}$$

To gain intuition on the utility of (3.1) and to spell out the importance of offering a tight bound for it, let us consider the compressed sensing problem as an example for which we aim to recover the true sparse signal $\boldsymbol{x}$ from its linear measurements. Here, $\boldsymbol{b}$ is a good but dense approximation to $\boldsymbol{x}$ obtained by, e.g., full gradient descent. Then (3.1) justifies that in order to obtain a structured (i.e., sparse) approximation by hard thresholding, the distance of the iterate to the true signal $\boldsymbol{x}$ is upper bounded by a multiple of 2 to the one before. For comparison, it is worth mentioning that ℓ_1-based convex algorithms usually utilize the soft thresholding operator which enjoys the non-expansiveness property [48], i.e., the iterate becomes closer to the optimum after projection. This salient feature might partially attribute to the wide range of applications of the ℓ_1-regularized formulations. Hence, to derive comparable performance guarantee, tightening the bound (3.1) is crucial in that it controls how much deviation the hard thresholding operator induces. This turns out to be more demanding for stochastic gradient methods, where the proxy $\boldsymbol{b}$ itself is affected by the randomness of sample realization. In other words, since $\boldsymbol{b}$ does not minimize the objective function (it only optimizes the objective in expectation), the deviation (3.1) makes it more challenging to analyze the convergence behavior. As an example, [110] proposed a stochastic solver for general sparsity-constrained programs but suffered a non-vanishing optimization error due to randomness. This indicates that to mitigate the randomness barrier, we have to seek a better bound to control the precision of the thresholded solution and the variance.

3.1.1 Summary of Contributions

In this work, we make three contributions:

1. We examine the tightness of (3.1) that has been used for a decade in the literature and show that the equality therein will never be attained. We then improve this bound and quantitatively characterize that the deviation is inversely proportional to the value of $\sqrt{k}$. Our bound is tight, in the sense that the equality we build can be attained for specific signals, hence cannot be improved if no additional information is available. Our bound is universal in the sense that it holds for all choices of k-sparse signals $\boldsymbol{x}$ and for general signals $\boldsymbol{b}$.

2. Owing to the tight estimate, we demonstrate how the RIP (or RIP-like) condition assumed by a wide range of hard thresholding based algorithms can be relaxed. In the context of compressed sensing, it means that in essence, many more kinds of sensing matrices or fewer measurements can be utilized for data acquisition. For machine learning, it suggests that existing algorithms are capable of handling more difficult statistical models.

3. Finally, we present an computationally efficient algorithm that applies hard thresholding in large-scale setting and we prove its linear convergence to a global optimum up to the statistical precision of the problem. We also prove that with sufficient samples, our algorithm identifies the true parameter for prevalent statistical models. Returning to (3.1), our analysis shows that only when the deviation is controlled below the multiple of 1.15 can such an algorithm succeed. This immediately implies that the conventional bound (3.1) is not applicable in the challenging scenario.

3.1.2 Notation

For an integer $d > 0$, suppose that Ω is a subset of $\{1,\ 2,\ \ldots,\ d\}$. Then for a general vector $\boldsymbol{v} \in \mathbb{R}^d$, we define $\mathcal{P}_\Omega(\cdot)$ as the orthogonal projection onto the support set Ω which retains elements contained in Ω and sets others to zero. That is,

$$\left(\mathcal{P}_\Omega(\boldsymbol{v})\right)_i = \begin{cases} v_i, & \text{if } i \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$$

In particular, let Γ be the support set indexing the k largest absolute components of $\boldsymbol{v}$. In this way, the hard thresholding operator is given by

$$\mathcal{H}_k(\boldsymbol{v}) = \mathcal{P}_\Gamma(\boldsymbol{v}).$$

We will also use the orthogonal projection of a vector $\boldsymbol{v}$ onto an ℓ_2-ball with radius ω. That is,

$$\Pi_\omega(\boldsymbol{v}) = \frac{\boldsymbol{v}}{\max\{1, \|\boldsymbol{v}\|/\omega\}}.$$

3.2 The Key Bound

We argue that the conventional bound (3.1) is not tight, in the sense that the equality therein can hardly be attained. To see this, recall how the bound was derived for a k-sparse signal $\boldsymbol{x}$ and a general one $\boldsymbol{b}$:

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\| = \|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b} + \boldsymbol{b} - \boldsymbol{x}\| \overset{\xi}{\leq} \|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b}\| + \|\boldsymbol{b} - \boldsymbol{x}\| \leq 2\|\boldsymbol{b} - \boldsymbol{x}\|,$$

where the last inequality holds because $\mathcal{H}_k(\boldsymbol{b})$ is a best k-sparse approximation to $\boldsymbol{b}$. The major issue occurs in ξ. Though it is the well-known triangle inequality and the equality could be attained if there is no restriction on the signals $\boldsymbol{x}$ and $\boldsymbol{b}$, we remind here that the signal $\boldsymbol{x}$ does have a specific structure – it is k-sparse. Note that in order to fulfill the equality in ξ, we must have $\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b} = \gamma(\boldsymbol{b} - \boldsymbol{x})$ for some $\gamma \geq 0$, that is,

$$\mathcal{H}_k(\boldsymbol{b}) = (\gamma + 1)\boldsymbol{b} - \gamma\boldsymbol{x}. \tag{3.2}$$

One may verify that the above equality holds *if and only if*

$$\boldsymbol{x} = \boldsymbol{b} = \mathcal{H}_k(\boldsymbol{b}). \tag{3.3}$$

To see this, let Ω be the support set of $\mathcal{H}_k(\boldsymbol{b})$ and $\overline{\Omega}$ be the complement. Let $\boldsymbol{b}_1 = \mathcal{P}_\Omega(\boldsymbol{b}) = \mathcal{H}_k(\boldsymbol{b})$ and $\boldsymbol{b}_2 = \mathcal{P}_{\overline{\Omega}}(\boldsymbol{b})$. Likewise, we define $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ as the components of $\boldsymbol{x}$ supported on Ω and $\overline{\Omega}$ respectively. Hence, (3.2) indicates $\boldsymbol{x}_1 = \boldsymbol{b}_1$ and $\boldsymbol{x}_2 = (1 + \gamma^{-1})\boldsymbol{b}_2$ where we assume $\gamma > 0$ since

$\gamma = 0$ immediately implies $\mathcal{H}_k(\boldsymbol{b}) = \boldsymbol{b}$ and hence the equality of (3.1) does not hold. If $\|\boldsymbol{b}_1\|_0 < k$, then we have $\boldsymbol{x}_2 = \boldsymbol{b}_2 = \mathbf{0}$ since $\boldsymbol{b}_1$ contains the k largest absolute elements of $\boldsymbol{b}$. Otherwise, the fact that $\|\boldsymbol{x}\|_0 \leq k$ and $\boldsymbol{x}_1 = \boldsymbol{b}_1$ implies $\boldsymbol{x}_2 = \mathbf{0}$, and hence $\boldsymbol{b}_2$. Therefore, we obtain (3.3).

When (3.3) happens, however, we in reality have $\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\| = \|\boldsymbol{b} - \boldsymbol{x}\| = 0$. In other words, the factor of 2 in (3.1) can essentially be replaced with an *arbitrary constant*! In this sense, we conclude that the bound (3.1) is not tight. Our new estimate for hard thresholding is as follows:

Theorem 3.1 (Tight Bound for Hard Thresholding)**.** *Let $\boldsymbol{b} \in \mathbb{R}^d$ be an arbitrary vector and $\boldsymbol{x} \in \mathbb{R}^d$ be any K-sparse signal. For any $k \geq K$, we have the following bound:*

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\| \leq \sqrt{\nu}\,\|\boldsymbol{b} - \boldsymbol{x}\|, \quad \nu = 1 + \frac{\rho + \sqrt{(4+\rho)\,\rho}}{2}, \quad \rho = \frac{\min\{K, d-k\}}{k - K + \min\{K, d-k\}}.$$

In particular, our bound is tight in the sense that there exist specific vectors of $\boldsymbol{b}$ and $\boldsymbol{x}$ such that the equality holds.

Remark 1 (Maximum of ν)**.** In contrast to the constant bound (3.1), our result asserts that the deviation resulting from hard thresholding is inversely proportional to $\sqrt{k}$ (when $K \leq d - k$) in a universal manner. When k tends to d, ρ is given by $(d-k)/(d-K)$ which is still decreasing with respect to k. Thus, the maximum value of ρ equals one. Even in this case, we find that $\sqrt{\nu_{\max}} = \sqrt{1 + \frac{\sqrt{5}+1}{2}} = \frac{\sqrt{5}+1}{2} \approx 1.618$.

Remark 2. Though for some batch algorithms such as IHT and CoSaMP, the constant bound (3.1) suffices to establish the convergence due to specific conditions, we show in Section 3.4 that it cannot ensure the global convergence for stochastic algorithms.

Remark 3. When $\boldsymbol{x}$ is not exactly K-sparse, we still can bound the error by $\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\| \leq \|\mathcal{H}_k(\boldsymbol{b}) - \mathcal{H}_k(\boldsymbol{x})\| + \|\mathcal{H}_k(\boldsymbol{x}) - \boldsymbol{x}\|$. Thus, without loss of generality, we assumed that the signal $\boldsymbol{x}$ is K-sparse.

Proof. (Sketch) Our bound follows from fully exploring the sparsity pattern of the signals and from fundamental arguments in optimization. Denote

$$\boldsymbol{w} := \mathcal{H}_k(\boldsymbol{b}).$$

Let Ω be the support set of $\boldsymbol{w}$ and let $\overline{\Omega}$ be its complement. We immediately have $\mathcal{P}_{\Omega}(\boldsymbol{b}) = \boldsymbol{w}$. Let Ω' be the support set of $\boldsymbol{x}$. Define

$$\boldsymbol{b}_1 = \mathcal{P}_{\Omega\setminus\Omega'}(\boldsymbol{b}), \quad \boldsymbol{b}_2 = \mathcal{P}_{\Omega\cap\Omega'}(\boldsymbol{b}), \quad \boldsymbol{b}_3 = \mathcal{P}_{\overline{\Omega}\setminus\Omega'}(\boldsymbol{b}), \quad \boldsymbol{b}_4 = \mathcal{P}_{\overline{\Omega}\cap\Omega'}(\boldsymbol{b}).$$

Likewise, we define $\boldsymbol{x}_i$ and $\boldsymbol{w}_i$ for $1 \leq i \leq 4$. Due to the construction, we have $\boldsymbol{w}_1 = \boldsymbol{b}_1, \boldsymbol{w}_2 = \boldsymbol{b}_2, \boldsymbol{w}_3 = \boldsymbol{w}_4 = \boldsymbol{x}_1 = \boldsymbol{x}_3 = \boldsymbol{0}$. Our goal is to estimate the maximum value of $\|\boldsymbol{w} - \boldsymbol{x}\|^2 / \|\boldsymbol{b} - \boldsymbol{x}\|^2$. It is easy to show that when attaining the maximum, $\|\boldsymbol{b}_3\|$ must be zero. Denote

$$\gamma := \frac{\|\boldsymbol{w} - \boldsymbol{x}\|^2}{\|\boldsymbol{b} - \boldsymbol{x}\|^2} = \frac{\|\boldsymbol{b}_1\|^2 + \|\boldsymbol{b}_2 - \boldsymbol{x}_2\|^2 + \|\boldsymbol{x}_4\|^2}{\|\boldsymbol{b}_1\|^2 + \|\boldsymbol{b}_2 - \boldsymbol{x}_2\|^2 + \|\boldsymbol{b}_4 - \boldsymbol{x}_4\|^2}. \tag{3.4}$$

Note that the variables here only involve $\boldsymbol{x}$ and $\boldsymbol{b}$. Arranging the equation we obtain

$$(\gamma - 1)\|\boldsymbol{b}_2 - \boldsymbol{x}_2\|^2 + \gamma\|\boldsymbol{b}_4 - \boldsymbol{x}_4\|^2 - \|\boldsymbol{x}_4\|^2 + (\gamma - 1)\|\boldsymbol{b}_1\|^2 = 0. \tag{3.5}$$

It is evident that for specific choices of $\boldsymbol{b}$ and $\boldsymbol{x}$, we have $\gamma = 1$. Since we are interested in the maximum of γ, we assume $\gamma > 1$ below. Fixing $\boldsymbol{b}$, we can view the left-hand side of the above equation as a function of $\boldsymbol{x}$. One may verify that the function has a positive definite Hessian matrix and thus it attains the minimum at stationary point given by

$$\boldsymbol{x}_2^* = \boldsymbol{b}_2, \quad \boldsymbol{x}_4^* = \frac{\gamma}{\gamma - 1}\boldsymbol{b}_4. \tag{3.6}$$

On the other hand, (3.5) implies that the minimum function value should not be greater than zero. Plugging the stationary point back gives

$$\|\boldsymbol{b}_1\|^2\gamma^2 - (2\|\boldsymbol{b}_1\|^2 + \|\boldsymbol{b}_4\|^2)\gamma + \|\boldsymbol{b}_1\|^2 \leq 0.$$

Solving the above inequality with respect to γ, we obtain

$$\gamma \leq 1 + \left(2\|\boldsymbol{b}_1\|^2\right)^{-1}\left(\|\boldsymbol{b}_4\|^2 + \sqrt{\left(4\|\boldsymbol{b}_1\|^2 + \|\boldsymbol{b}_4\|^2\right)\|\boldsymbol{b}_4\|^2}\right). \tag{3.7}$$

To derive an upper bound that is uniform over the choice of $\boldsymbol{b}$, we recall that $\boldsymbol{b}_1$ contains the largest

absolute elements of $\boldsymbol{b}$ while $\boldsymbol{b}_4$ has smaller values. In particular, the average in $\boldsymbol{b}_1$ is larger than that in $\boldsymbol{b}_4$, which gives

$$\|\boldsymbol{b}_4\|^2/\|\boldsymbol{b}_4\|_0 \leq \|\boldsymbol{b}_1\|^2/\|\boldsymbol{b}_1\|_0.$$

Note that $\|\boldsymbol{b}_1\|_0 = k - \|\boldsymbol{b}_2\|_0 = k - (K - \|\boldsymbol{b}_4\|_0)$. Hence, combining with the fact that $0 \leq \|\boldsymbol{b}_4\|_0 \leq \min\{K, d-k\}$ and optimizing over $\|\boldsymbol{b}_4\|_0$ in the above inequality gives

$$\|\boldsymbol{b}_4\|^2 \leq \frac{\min\{K, d-k\}}{k - K + \min\{K, d-k\}} \|\boldsymbol{b}_1\|^2. \tag{3.8}$$

Finally, we arrive at a uniform upper bound

$$\gamma \leq 1 + \frac{\rho + \sqrt{(4+\rho)\,\rho}}{2}, \quad \rho = \frac{\min\{K, d-k\}}{k - K + \min\{K, d-k\}}.$$

See Appendix 3.B for the full proof. □

Remark 4 (Tightness)**.** We construct proper vectors $\boldsymbol{b}$ and $\boldsymbol{x}$ to establish the tightness of our bound by a backward induction. Note that γ equals ν if and only if $\|\boldsymbol{b}_4\|^2 = \rho\,\|\boldsymbol{b}_1\|^2$. Hence, we pick

$$\|\boldsymbol{b}_4\|^2 = \rho\,\|\boldsymbol{b}_1\|^2, \quad \boldsymbol{x}_2 = \boldsymbol{b}_2, \quad \boldsymbol{x}_4 = \frac{\nu}{\nu - 1}\boldsymbol{b}_4, \tag{3.9}$$

where $\boldsymbol{x}_2$ and $\boldsymbol{x}_4$ are actually chosen as the stationary point as in (3.6). We note that the quantity of ν only depends on d, k and K, not on the components of $\boldsymbol{b}$ or $\boldsymbol{x}$. Plugging the above back to (3.4) justifies $\gamma = \nu$.

It remains to show that our choices in (3.9) do not violate the definition of $\boldsymbol{b}_i$'s, i.e., we need to ensure that the elements in $\boldsymbol{b}_1$ or $\boldsymbol{b}_2$ are equal to or greater than those in $\boldsymbol{b}_3$ or $\boldsymbol{b}_4$. Note that there is no such constraint for the K-sparse vector $\boldsymbol{x}$. Let us consider the case $K < d - k$ and $\|\boldsymbol{b}_4\|_0 = K$, so that $\|\boldsymbol{b}_1\|_0 = k$ and $\rho = K/k$. Thus, the first equality of (3.9) holds as soon as all the entries of $\boldsymbol{b}$ have same magnitude. The fact $\|\boldsymbol{b}_4\|_0 = K$ also implies Ω' is a subset of $\overline{\Omega}$ due to the definition of $\boldsymbol{b}_4$ and the sparsity of $\boldsymbol{x}$, hence we have $\boldsymbol{x}_2 = \boldsymbol{0} = \boldsymbol{b}_2$. Finally, picking $\boldsymbol{x}_4$ as we did in (3.9) completes the reasoning since it does not violate the sparsity constraint on $\boldsymbol{x}$.

As we pointed out and just verified, the bound given by Theorem 3.1 is tight. However, if there is additional information for the signals, a better bound can be established. For instance, let us

further assume that the signal $\boldsymbol{b}$ is r-sparse. If $r \leq k$, then $\boldsymbol{b}_4$ is a zero vector and (3.7) reads as $\gamma \leq 1$. Otherwise, we have $\|\boldsymbol{b}_4\|_0 \leq \min\{K, r-k\}$ and (3.8) is improved to

$$\|\boldsymbol{b}_4\|^2 \leq \frac{\min\{K, r-k\}}{k - K + \min\{K, r-k\}} \|\boldsymbol{b}_1\|^2 .$$

Henceforth, we can show that the parameter ρ is given by

$$\rho = \frac{\min\{K, r-k\}}{k - K + \min\{K, r-k\}}.$$

Note that the fact $r \leq d$ implies that the above is a tighter bound than the one in Theorem 3.1.

We would also like to mention that in Lemma 1 of [75], a closely related bound was established:

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b}\| \leq \sqrt{\frac{d-k}{d-K}} \|\boldsymbol{b} - \boldsymbol{x}\| . \tag{3.10}$$

One may use this nice result to show that

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\| \leq \|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b}\| + \|\boldsymbol{b} - \boldsymbol{x}\| \leq \left(1 + \sqrt{\frac{d-k}{d-K}}\right) \|\boldsymbol{b} - \boldsymbol{x}\| , \tag{3.11}$$

which also improves on (3.1) provided $k > K$. However, one shortcoming of (3.11) is that the factor depends on the dimension. For comparison, we recall that in the regime $K \leq d - k$, our bound is free of the dimension. This turns out to be a salient feature to integrate hard thresholding into stochastic methods, and we will comment on it more in Section 3.4.

3.3 Implications to Compressed Sensing

In this section, we investigate the implications of Theorem 3.1 for compressed sensing and signal processing. Since most of the HT-based algorithms utilize the deviation bound (3.1) to derive the convergence condition, they can be improved by our new bound. We exemplify the power of our theorem on two popular algorithms: IHT [20] and CoSaMP [103]. We note that our analysis also applies to their extensions such as [10]. To be clear, the purpose of this section is not dedicated to improving the best RIP condition for which recovery is possible by any methods (either convex or non-convex). Rather, we focus on two broadly used greedy algorithms and illustrate how our bound

improves on previous results.

We proceed with a brief review of the problem setting in compressed sensing. Compressed sensing algorithms aim to recover the true K-sparse signal $\boldsymbol{x}^* \in \mathbb{R}^d$ from a set of its (perhaps noisy) measurements

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{e}, \tag{3.12}$$

where $\boldsymbol{e} \in \mathbb{R}^d$ is some observation noise and $\boldsymbol{A}$ is a known $n \times d$ sensing matrix with $n \ll d$, hence the name compressive sampling. In general, the model is not identifiable since it is an underdetermined system. Yet, the prior knowledge that $\boldsymbol{x}^*$ is sparse radically changes the premise. That is, if the geometry of the sparse signal is preserved under the action of the sampling matrix $\boldsymbol{A}$ for a restricted set of directions, then it is possible to invert the sampling process. Such a novel idea was quantified as the kth restricted isometry property of $\boldsymbol{A}$ by [37], which requires that there exists a constant $\delta \geq 0$, such that for all k-sparse signals $\boldsymbol{x}$

$$(1-\delta)\left\|\boldsymbol{x}\right\|^2 \leq \left\|\boldsymbol{A}\boldsymbol{x}\right\|^2 \leq (1+\delta)\left\|\boldsymbol{x}\right\|^2. \tag{3.13}$$

The kth restricted isometry constant (RIC) δ_k is then defined as the smallest one that satisfies the above inequalities. Note that $\delta_{2k} < 1$ is the minimum requirement for distinguishing all k-sparse signals from the measurements. This is because for two arbitrary k-sparse vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and their respective measurements $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, the RIP condition reads as

$$(1-\delta_{2k})\left\|\boldsymbol{x}_1-\boldsymbol{x}_2\right\|^2 \leq \left\|\boldsymbol{y}_1-\boldsymbol{y}_2\right\|^2 \leq (1+\delta_{2k})\left\|\boldsymbol{x}_1-\boldsymbol{x}_2\right\|^2,$$

for which $\delta_{2k} < 1$ guarantees that $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ implies $\boldsymbol{y}_1 \neq \boldsymbol{y}_2$. To date, there are three quintessential examples known to exhibit a profound restricted isometry behavior as long as the number of measurements is large enough: Gaussian matrices (optimal RIP, i.e., very small δ_k), partial Fourier matrices (fast computation) and Bernoulli ensembles (low memory footprint). Notably, it was shown in recent work that random matrices with a heavy-tailed distribution also satisfy the RIP with overwhelming probability [1, 89].

Equipped with the standard RIP condition, many efficient algorithms have been developed. A partial list includes ℓ_1-norm based convex programs, IHT, CoSaMP, SP and regularized OMP [104],

along with much interesting work devoted to improving or sharpening the RIP condition [149, 102, 30, 101]. To see why relaxing RIP is of central interest, note that the standard result [12] asserts that the RIP condition $\delta_k \leq \delta$ holds with high probability over the draw of $\boldsymbol{A}$ provided

$$n \geq \mathrm{C}_0 \delta^{-2} k \log(d/k). \tag{3.14}$$

Hence, a slight relaxation of the condition $\delta_k \leq \delta$ may dramatically decrease the number of measurements. That being said, since the constant C_0 above is unknown, in general one cannot tell the precise sample size for greedy algorithms. Estimating the constant is actually the theme of phase transition [53, 52]. While precise phase transition for ℓ_1-based convex programs has been well understood [146], an analogous result for greedy algorithms remains an open problem. Notably, in [18], phase transition for IHT/CoSaMP was derived using the constant bound (3.1). We believe that our tight bound shall sharpen these results and we leave it as our future work. In the present chapter, we focus on the ubiquitous RIP condition. In the language of RIP, we establish improved results.

3.3.1 Iterative Hard Thresholding

The IHT algorithm recovers the underlying K-sparse signal $\boldsymbol{x}^*$ by iteratively performing a full gradient descent on the least-squares loss followed by a hard thresholding step. That is, IHT starts with an arbitrary point $\boldsymbol{x}^0$ and at the t-th iteration, it updates the new solution as follows:

$$\boldsymbol{x}^t = \mathcal{H}_k \left(\boldsymbol{x}^{t-1} + \boldsymbol{A}^\top (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{t-1}) \right). \tag{3.15}$$

Note that [20] used the parameter $k = K$. However, in practice one may only know to an upper bound on the true sparsity K. Thus, we consider the projection sparsity k as a parameter that depends on K. To establish the global convergence with a geometric rate of 0.5, [20] applied the bound (3.1) and assumed the RIP condition

$$\delta_{2k+K} \leq 0.18. \tag{3.16}$$

As we have shown, (3.1) is actually not tight and hence, their results, especially the RIP condition can be improved by Theorem 3.1.

Theorem 3.2. *Consider the model* (3.12) *and the IHT algorithm* (3.15). *Pick* $k \geq K$ *and let* $\{\boldsymbol{x}^t\}_{t\geq 1}$ *be the iterates produced by IHT. Then, under the RIP condition* $\delta_{2k+K} \leq 1/\sqrt{8\nu}$*, for all* $t \geq 1$

$$\left\|\boldsymbol{x}^t - \boldsymbol{x}^*\right\| \leq 0.5^t \left\|\boldsymbol{x}^0 - \boldsymbol{x}^*\right\| + \mathrm{C}\left\|\boldsymbol{e}\right\|,$$

where ν *is given by Theorem 3.1.*

Let us first study the vanilla case $k = K$. [20] required $\delta_{3K} \leq 0.18$ whereas our analysis shows $\delta_{3K} \leq 0.22$ suffices. Note that even a little relaxation on RIP is challenging and may require several pages of mathematical induction [33, 29, 61]. In contrast, our improvement comes from a direct application of Theorem 3.1 which only modifies several lines of the original proof in [20]. See Appendix 3.C for details. In view of (3.14), we find that the necessary number of measurements for IHT is dramatically reduced with a factor of 0.67 by our new theorem in that the minimum requirement of n is inversely proportional to the square of δ_{2k+K}.

Another important consequence of the theorem is a characterization on the RIP condition and the sparsity parameter, which, to the best of our knowledge, has not been studied in the literature. In [20], when gradually tuning k larger than K, it always requires $\delta_{2k+K} \leq 0.18$. Note that due to the monotonicity of RIC, i.e., $\delta_r \leq \delta_{r'}$ if $r \leq r'$, the condition turns out to be more and more stringent. Compared to their result, since ν is inversely proportional to $\sqrt{k}$, Theorem 3.2 is powerful especially when k becomes larger. For example, suppose $k = 20K$. In this case, Theorem 3.2 justifies that IHT admits the linear convergence as soon as $\delta_{41K} \leq 0.32$ whereas [20] requires $\delta_{41K} \leq 0.18$. Such a property is appealing in practice, in that among various real-world applications, the true sparsity is indeed unknown and we would like to estimate a conservative upper bound on it.

On the other hand, for a given sensing matrix, there does exist a fundamental limit for the maximum choice of k. To be more precise, the condition in Theorem 3.2 together with the probabilistic argument (3.14) require

$$1/\sqrt{8\nu} \geq \delta_{2k+K}, \quad \mathrm{C}_1\nu(2k+K)\log\left(d/(2k+K)\right) \leq n.$$

Although it could be very interesting to derive a quantitative characterization for the maximum value of k, we argue that it is perhaps intractable owing to two aspects: First, it is known that one has to enumerate all the combinations of the $2k + K$ columns of $\boldsymbol{A}$ to compute the restricted isometry constant δ_{2k+K} [8, 9]. This suggests that it is NP-hard to estimate the largest admissible value of k. Also, there is no analytic solution of the stationary point for the left-hand side of the second inequality.

3.3.2 Compressive Sampling Matching Pursuit

The CoSaMP algorithm proposed by [103] is one of the most efficient algorithms for sparse recovery. Let $F(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$. CoSaMP starts from an arbitrary initial point $\boldsymbol{x}^0$ and proceeds as follows:

$$\begin{aligned}
\Omega^t &= \operatorname{supp}\left(\nabla F(\boldsymbol{x}^{t-1}),\ k\right) \cup \operatorname{supp}\left(\boldsymbol{x}^{t-1}\right), \\
\boldsymbol{b}^t &= \arg\min_{\boldsymbol{x}}\ F(\boldsymbol{x}),\ \text{s.t. } \operatorname{supp}\left(\boldsymbol{x}\right) \subset \Omega^t, \\
\boldsymbol{x}^t &= \mathcal{H}_k\left(\boldsymbol{b}^t\right).
\end{aligned}$$

Compared to IHT which performs hard thresholding after gradient update, CoSaMP prunes the gradient at the beginning of each iteration, followed by solving a least-squares program restricted on a small support set. In particular, in the last step, CoSaMP applies hard thresholding to form a k-sparse iterate for future updates. The analysis of CoSaMP consists of bounding the estimation error in each step. Owing to Theorem 3.1, we advance the theoretical result of CoSaMP by improving the error bound for its last step, and hence the RIP condition.

Theorem 3.3. *Consider the model* (3.12) *and the CoSaMP algorithm. Pick $k \geq K$ and let $\{\boldsymbol{x}^t\}_{t\geq 1}$ be the iterates produced by CoSaMP. Then, under the RIP condition*

$$\delta_{3k+K} \leq \frac{\left(\sqrt{32\nu + 49} - 9\right)^{1/2}}{4\sqrt{\nu - 1}},$$

it holds that for all $t \geq 1$

$$\left\|\boldsymbol{x}^t - \boldsymbol{x}^*\right\| \leq 0.5^t \left\|\boldsymbol{x}^0 - \boldsymbol{x}^*\right\| + \mathrm{C}\,\|\boldsymbol{e}\|,$$

where ν is given by Theorem 3.1.

Roughly speaking, the bound is still inversely proportional to $\sqrt{\nu}$. Hence, it is monotonically increasing with respect to k, indicating our theorem is more effective for a large quantity of k. In fact, for the CoSaMP algorithm, our bound above is superior to the best known result even when $k = K$. To see this, we have the RIP condition $\delta_{4K} \leq 0.31$. In comparison, [103] derived a bound $\delta_{4K} \leq 0.1$ and [62, Theorem 6.27] improved it to $\delta_{4K} < 0.29$ for a geometric rate of 0.5. We notice that for binary sparse vectors, [75] presented a different proof technique and obtained the RIP condition $\delta_{4K} \leq 0.35$ for CoSaMP.

3.4 Hard Thresholding in Large-Scale Optimization

Now we move on to the machine learning setting where our focus is pursuing an optimal sparse solution that minimizes a given objective function based on a set of training samples $Z_1^n := \{Z_i\}_{i=1}^n$. Different from compressed sensing, we usually have sufficient samples which means n can be very large. Therefore, the computational complexity is of primary interest. Formally, we are interested in optimizing the following program:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}; Z_1^n) = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}; Z_i), \quad \text{s.t. } \|\boldsymbol{x}\|_0 \leq K,\ \|\boldsymbol{x}\| \leq \omega. \tag{3.17}$$

The global optimum of the above problem is denoted by $\boldsymbol{x}_{\text{opt}}$. We note that the objective function is presumed to be decomposable with respect to the samples. This is quite a mild condition and most of the popular machine learning models fulfill it. Typical examples include (but not limited to) the sparse linear regression and sparse logistic regression:

- Sparse Linear Regression: For all $1 \leq i \leq n$, we have $Z_i = (\boldsymbol{a}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ and the loss function $F(\boldsymbol{x}; Z_1^n) = \frac{1}{2n}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2$ is the least-squares and can be explained by $f(\boldsymbol{x}; Z_i) = \frac{1}{2}\|\boldsymbol{a}_i \cdot \boldsymbol{x} - y_i\|^2$.

- Sparse Logistic Regression: For all $1 \leq i \leq n$, we have $Z_i = (\boldsymbol{a}_i, y_i) \in \mathbb{R}^d \times \{+1, -1\}$ and the negative log-likelihood is penalized, i.e., $F(\boldsymbol{x}; Z_1^n) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \boldsymbol{a}_i \cdot \boldsymbol{x}\right)\right)$ for which $f(\boldsymbol{x}; Z_i) = \log\left(1 + \exp\left(-y_i \boldsymbol{a}_i \cdot \boldsymbol{x}\right)\right)$.

To ease notation, we will often write $F(\boldsymbol{x}; Z_1^n)$ as $F(\boldsymbol{x})$ and $f(\boldsymbol{x}; Z_i)$ as $f_i(\boldsymbol{x})$ for $i = 1, 2, \cdots, n$. It is worth mentioning that the objective function $F(\boldsymbol{x})$ is allowed to be non-convex. Hence, in order to ensure the existence of a global optimum, a natural option is to impose an ℓ_p-norm ($p \geq 1$) constraint [94, 95]. Here we choose the ℓ_2-norm constraint owing to its fast projection. Previous work, e.g., [2] prefers the computationally less efficient ℓ_1-norm to promote sparsity and to guarantee the existence of optimum. In our problem, yet, we already have imposed the hard sparsity constraint so the ℓ_2-norm constraint is a better fit.

The major contribution of this section is a computationally efficient algorithm termed hard thresholded stochastic variance reduced gradient method (HT-SVRG) to optimize (3.17), tackling one of the most important problems in large-scale machine learning: producing sparse solutions by stochastic methods. We emphasize that the formulation (3.17) is in stark contrast to the ℓ_1-regularized programs considered by previous stochastic solvers such as Prox-SVRG [152] and SAGA [48]. We target here a stochastic algorithm for the *non-convex* problem that is less exploited in the literature. From a theoretical perspective, (3.17) is more difficult to analyze but it always produces sparse solutions, whereas performance guarantees for convex programs are fruitful but one cannot characterize the sparsity of the obtained solution (usually the solution is not sparse). When we appeal to stochastic algorithms to solve the convex programs, the ℓ_1-norm formulation becomes much less effective in terms of sparsification, naturally owing to the randomness. See [85, 151, 54] for more detailed discussion on the issue. We also remark that existing work such as [159, 10, 75] investigated the sparsity-constrained problem (3.17) in a batch scenario, which is not practical for large-scale learning problems. The perhaps most related work to our new algorithm is [110]. Nonetheless, the optimization error therein does not vanish for noisy statistical models.

Our main result shows that for prevalent statistical models, our algorithm is able to recover the true parameter with a linear rate. Readers should distinguish the optimal solution $\boldsymbol{x}_{\rm opt}$ and the true parameter. For instance, consider the model (3.12). Minimizing (3.17) does not amount to recovering $\boldsymbol{x}^*$ if there is observation noise. In fact, the convergence to $\boldsymbol{x}_{\rm opt}$ is only guaranteed to an accuracy reflected by the *statistical precision* of the problem, i.e., $\|\boldsymbol{x}^* - \boldsymbol{x}_{\rm opt}\|$, which is the best one can hope for any statistical model [2]. We find that the global convergence is attributed to both the tight bound and the variance reduction technique to be introduced below, and examining the necessity of them is an interesting future work.

Algorithm 1 Hard Thresholded Stochastic Variance Reduced Gradient Method (HT-SVRG)

Require: Training samples $\{Z_i\}_{i=1}^n$, maximum stage count S, sparsity parameter k, update frequency m, learning rate η, radius ω, initial solution $\tilde{\boldsymbol{x}}^0$.
Ensure: Optimal solution $\tilde{\boldsymbol{x}}^S$.

1: **for** $s = 1$ to S **do**
2: Set $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}^{s-1},\ \tilde{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{\boldsymbol{x}}),\ \boldsymbol{x}^0 = \tilde{\boldsymbol{x}}$.
3: **for** $t = 1$ to m **do**
4: Uniformly pick $i_t \in \{1, 2, \cdots, n\}$ and update the solution

$$\begin{aligned}\boldsymbol{b}^t &= \boldsymbol{x}^{t-1} - \eta\left(\nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\tilde{\boldsymbol{x}}) + \tilde{\boldsymbol{\mu}}\right),\\ \boldsymbol{r}^t &= \mathcal{H}_k\left(\boldsymbol{b}^t\right),\\ \boldsymbol{x}^t &= \Pi_\omega(\boldsymbol{r}^t).\end{aligned}$$

5: **end for**
6: Uniformly choose $j^s \in \{0, 1, \cdots, m-1\}$ and set $\tilde{\boldsymbol{x}}^s = \boldsymbol{x}^{j^s}$.
7: **end for**

3.4.1 Algorithm

Our algorithm (Algorithm 1) applies the framework of [78], where the primary idea is to leverage past gradients for the current update for the sake of variance reduction – a technique that has a long history in statistics [114]. To guarantee that each iterate is k-sparse, it then invokes the hard thresholding operation. Note that the orthogonal projection for $\boldsymbol{r}^t$ will not change the support set, and hence $\boldsymbol{x}^t$ is still k-sparse. Also note that our sparsity constraint in (3.17) reads as $\|\boldsymbol{x}\|_0 \leq K$. What we will show below is that when the parameter k is properly chosen (which depends on K), we obtain a globally convergent sequence of iterates.

The most challenging part on establishing the global convergence comes from the hard thresholding operation $\mathcal{H}_k\left(\boldsymbol{r}^t\right)$. Note that it is $\boldsymbol{b}^t$ that reduces the objective value in expectation. If $\boldsymbol{b}^t$ is not k-sparse (usually it is dense), $\boldsymbol{x}^t$ is not equal to $\boldsymbol{b}^t$ so it does not decrease the objective function. In addition, compared with the convex proximal operator [48] which enjoys the non-expansiveness of the distance to the optimum, the hard thresholding step can enlarge the distance up to a multiple of 2 if using the bound (3.1). What makes it a more serious issue is that these inaccurate iterates $\boldsymbol{x}^t$ will be used for future updates, and hence the error might be progressively propagated at an exponential rate.

Our key idea is to first bound the curvature of the function from below and above to establish RIP-like condition, which, combined with Theorem 3.1, downscales the deviation resulting from

hard thresholding. Note that ν is always greater than one (see Theorem 3.1), hence the curvature bound is necessary. Due to variance reduction, we show that the optimization error vanishes when restricted on a small set of directions as soon as we have sufficient samples. Moreover, with hard thresholding we are able to control the error per iteration and to obtain near-optimal sample complexity.

3.4.2 Deterministic Analysis

We will first establish a general theorem that characterizes the progress of HT-SVRG for approximating an arbitrary K-sparse signal $\hat{\boldsymbol{x}}$. Then we will discuss how to properly choose the hyper-parameters of the algorithm. Finally we move on to specify $\hat{\boldsymbol{x}}$ to develop convergence results for a global optimum of (3.17) and for a true parameter (e.g., $\boldsymbol{x}^*$ of the compressed sensing problem).

Assumptions

Recall Definition 2.1 and Definition 2.2. We assume the following:

$(A1)$ $F(\boldsymbol{x})$ satisfies the RSC condition with parameter α_{k+K}.

$(A2)$ For all $1 \leq i \leq n$, $f_i(\boldsymbol{x})$ satisfies the RSS condition with parameter L_{3k+K}.

Here, we recall that K was first introduced in (3.17) and the parameter k was used in our algorithm. Compared to the convex algorithms such as SAG [124], SVRG [78] and SAGA [48] that assume strong convexity and smoothness everywhere, we only assume these in a restricted sense. This is more practical especially in the high dimensional regime where the Hessian matrix could be degenerate [2]. We also stress that the RSS condition is imposed on each $f_i(\boldsymbol{x})$, whereas prior work requires it for $F(\boldsymbol{x})$ which is milder than ours [105].

Upper Bound of Progress

For brevity, let us denote

$$L := L_{3k+K}, \quad \alpha := \alpha_{k+K}, \quad c := L/\alpha,$$

where we call the quantity c as the condition number of the problem. It is also crucial to measure the ℓ_2-norm of the gradient restricted on sparse directions, and we write

$$\|\nabla_{3k+K}F(\boldsymbol{x})\| := \max_{\Omega}\big\{\, \|\mathcal{P}_{\Omega}\left(\nabla F(\boldsymbol{x})\right)\| :\ |\Omega| \leq 3k+K\big\}.$$

Note that for convex programs, the above evaluated at a global optimum is zero. As will be clear, $\|\nabla_{3k+K}F(\boldsymbol{x})\|$ reflects how close the iterates returned by HT-SVRG can be to the point $\boldsymbol{x}$. For prevalent statistical models, it vanishes when there are sufficient samples. Related to this quantity, our analysis also involves

$$Q(\boldsymbol{x}) := \left(16\nu\eta^2 L\omega m + \frac{2\omega}{\alpha}\right)\|\nabla_{3k+K}F(\boldsymbol{x})\| + 4\nu\eta^2 m\,\|\nabla_{3k+K}F(\boldsymbol{x})\|^2,$$

where we recall that ν is the expansiveness factor given by Theorem 3.1, η and m are used in the algorithm and ω is a universal constant that upper bounds the ℓ_2-norm of the signal we hope to estimate. Virtually, with an appropriate parameter setting, $Q(\boldsymbol{x})$ scales as $\|\nabla_{3k+K}F(\boldsymbol{x})\|$ which will be clarified. For a particular stage s, we denote $\mathcal{I}^s := \{i_1, i_2, \cdots, i_m\}$, i.e., the samples randomly chosen for updating the solution.

Theorem 3.4. *Consider Algorithm 1 and a K-sparse signal $\hat{\boldsymbol{x}}$ of interest. Assume $(A1)$ and $(A2)$. Pick the step size $0 < \eta < 1/(4L)$. If $\nu < 4L/(4L-\alpha)$, then it holds that*

$$\mathbb{E}\big[F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}})\big] \leq \beta^s\big[F(\tilde{\boldsymbol{x}}^0) - F(\hat{\boldsymbol{x}})\big] + \tau(\hat{\boldsymbol{x}}),$$

where the expectation is taken over $\{\mathcal{I}^1, j^1, \mathcal{I}^2, j^2, \cdots, \mathcal{I}^s, j^s\}$ and $0 < \beta < 1$ provided that m is large enough. In particular, for $1/(1-\eta\alpha) < \nu < 4L/(4L-\alpha)$, we have

$$\begin{aligned}\beta = \beta_1 &:= \frac{1}{(2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1)\,m} + \frac{2\nu\eta^2\alpha L}{2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1},\\ \tau(\hat{\boldsymbol{x}}) = \tau_1(\hat{\boldsymbol{x}}) &:= \frac{\alpha Q(\hat{\boldsymbol{x}})}{2(2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1)(1-\beta_1)m}.\end{aligned}$$

For $\nu \leq 1/(1-\eta\alpha)$, we have

$$\beta = \beta_2 := \frac{1}{\nu\eta\alpha(1-2\eta L)m} + \frac{2\eta L}{1-2\eta L}, \quad \tau(\hat{\boldsymbol{x}}) = \tau_2(\hat{\boldsymbol{x}}) := \frac{Q(\hat{\boldsymbol{x}})}{2\nu\eta\alpha(1-2\eta L)(1-\beta_2)m}.$$

The proof can be found in Appendix 3.D.1.

Remark 5. For the theorem to hold, $\sqrt{\nu} < \sqrt{4L/(4L-\alpha)} \leq \sqrt{4/3} \approx 1.15$ due to $L \geq \alpha$. Hence, the conventional bound (3.1) is not applicable. In contrast, Theorem 3.1 asserts that this condition can be fulfilled by tuning k slightly larger than K.

Remark 6. With the conditions on η and ν, the coefficient β is always less than one provided that m is sufficiently large.

Remark 7. The theorem does *not* assert convergence to an arbitrary sparse vector $\hat{\boldsymbol{x}}$. This is because $F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}})$ might be less than zero. However, specifying $\hat{\boldsymbol{x}}$ does give convergence results, as to be elaborated later.

Hyper-Parameter Setting

Before moving on to the convergence guarantee, let us discuss the minimum requirement on the hyper-parameters k, m and η, and determine how to choose them to simplify Theorem 3.4.

For the sake of success of HT-SVRG, we require $\nu < 4c/(4c-1)$, which implies $\rho < 1/(16c^2 - 4c)$. Recall that ρ is given in Theorem 3.1. In general, we are interested in the regime $K \leq k \ll d$. Hence, we have $\rho = K/k$ and the minimum requirement for the sparsity parameter is

$$k > (16c^2 - 4c)K. \tag{3.18}$$

To our knowledge, the idea of relaxed sparsity was first introduced in [163] for OMP and in [75] for projected gradient descent. However, the relaxed sparsity here emerges in a different way in that HT-SVRG is a stochastic algorithm, and their proof technique cannot be used.

We also contrast our tight bound to the inequality (3.11) that is obtained by combining the triangle inequality and Lemma 1 of [75]. Following our proof pipeline, (3.11) gives

$$k \geq \left(1 - \left(\sqrt{4c(4c-1)^{-1}} - 1\right)^2\right) d + \left(\sqrt{4c(4c-1)^{-1}} - 1\right)^2 K$$

which grows with the dimension d, whereas using Theorem 3.1 the sparsity parameter k depends only on the desired sparsity K. In this regard, we conclude that for the stochastic case, our bound is vital.

Another component of the algorithm is the update frequency m. Intuitively, HT-SVRG performs m number of stochastic gradient update followed by a full gradient evaluation, in order to mitigate the variance. In this light, m should not be too small. Otherwise, the algorithm reduces to the full gradient method which is not computationally efficient. On the other spectrum, a large m leads to a slow convergence that is reflected in the convergence coefficient β. To quantitatively analyze how m should be selected, let us consider the case $\nu \leq 1/(1-\eta\alpha)$ for example. The case $1/(1-\eta\alpha) < \nu < 4L/(4L-\alpha)$ follows in a similar way. In order to ensure $\beta_2 < 1$, we must have $m > 1/\left(\nu\eta\alpha(1-4\eta L)\right)$. In particular, picking

$$\eta = \frac{\eta'}{L}, \quad \eta' \in (0, 1/4), \tag{3.19}$$

we find that the update frequency m has to satisfy

$$m > \frac{c}{\nu\eta'(1-\eta')}, \tag{3.20}$$

which is of the same order as in the convex case [78] when $\eta' = \Theta(1)$. Note that the way we choose the learning rate $\eta = \eta'/L$ is also a common practice in convex optimization [107].

With (3.18), (3.19) and (3.20) in mind, we provide detailed choices of the hyper-parameters. Due to $0 < \eta < 1/(4L)$, β_1 is monotonically increasing with respect to ν. By Theorem 3.1, we know that ν is decreasing with respect to k. Thus, a larger quantity of k results in a smaller value of β_1, and hence a faster rate. Interestingly, for β_2 we discover that the smaller the k is, the faster the algorithm concentrates. Hence, we have the following consequence:

Proposition 3.5. *Fix η and m. Then the optimal choice of ν in Theorem 3.4 is $\nu = 1/(1-\eta\alpha)$ in the sense that the convergence coefficient β attains the minimum.*

In light of the proposition, in the sections to follow, we will only consider the setting $\nu = 1/(1-\eta\alpha)$. But we emphasize that our analysis and results essentially apply to any $\nu \leq 4L/(4L-\alpha)$.

Now let

$$\eta = \frac{1}{8L}, \quad m = 4(8c-1), \quad k = 8c(8c-1)K. \tag{3.21}$$

This gives

$$\beta = \frac{2}{3}, \quad \tau(\hat{\boldsymbol{x}}) = \frac{5\omega}{\alpha} \left\| \nabla_{3k+K} F(\hat{\boldsymbol{x}}) \right\| + \frac{1}{\alpha L} \left\| \nabla_{3k+K} F(\hat{\boldsymbol{x}}) \right\|^2 . \tag{3.22}$$

Global Linear Convergence

We are in the position to state the global linear convergence to an optimum of the sparsity-constrained optimization program (3.17).

Corollary 3.6. *Assume* $(A1)$ *and* $(A2)$. *Consider the HT-SVRG algorithm with hyper-parameters given in* (3.21). *Then the sequence* $\{\tilde{\boldsymbol{x}}^s\}_{s\geq 1}$ *converges linearly to a global optimum* $\boldsymbol{x}_{\rm opt}$ *of* (3.17)

$$\begin{aligned} \mathbb{E}\big[F(\tilde{\boldsymbol{x}}^s) - F(\boldsymbol{x}_{\rm opt})\big] \leq & \left(\frac{2}{3}\right)^s \big[F(\tilde{\boldsymbol{x}}^0) - F(\boldsymbol{x}_{\rm opt})\big] \\ & + \frac{5\omega}{\alpha} \left\| \nabla_{3k+K} F(\boldsymbol{x}_{\rm opt}) \right\| + \frac{1}{\alpha L} \left\| \nabla_{3k+K} F(\boldsymbol{x}_{\rm opt}) \right\|^2 . \end{aligned}$$

Proof. This is a direct consequence of Theorem 3.4. □

Whenever $\nabla_{3k+K} F(\boldsymbol{x}_{\rm opt}) = \boldsymbol{0}$, the corollary reads as

$$\mathbb{E}\big[F(\tilde{\boldsymbol{x}}^s) - F(\boldsymbol{x}_{\rm opt})\big] \leq \left(\frac{2}{3}\right)^s \big[F(\tilde{\boldsymbol{x}}^0) - F(\boldsymbol{x}_{\rm opt})\big].$$

It implies that if one is solving a convex problem without the sparsity constraint but the optimal solution happens to be sparse, it is safe to perform hard thresholding without loss of optimality. In the noiseless compressed sensing setting where $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^*$, the corollary guarantees that HT-SVRG exactly recovers the underlying true signal $\boldsymbol{x}^*$ when $F(\boldsymbol{x})$ is chosen as the least-squares loss in that $\boldsymbol{x}_{\rm opt} = \boldsymbol{x}^*$ and $\nabla F(\boldsymbol{x}^*) = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{y}) = \boldsymbol{0}$.

On the other side, the RSC property implies that

$$\|\tilde{\boldsymbol{x}}^s - \hat{\boldsymbol{x}}\| \leq \sqrt{\frac{2\max\{F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}}), 0\}}{\alpha}} + \frac{2\left\|\nabla_{k+K} F(\hat{\boldsymbol{x}})\right\|}{\alpha}.$$

The proof is straightforward and can be found in Lemma 14 of [129]. Now we specify $\hat{\boldsymbol{x}}$ as the true parameter of some statistical model, for instance, $\boldsymbol{x}^*$ in (3.12). It is hence possible to establish recovery guarantee of $\boldsymbol{x}^*$, which is known as the problem of parameter estimation.

Corollary 3.7. *Assume* $(A1)$ *and* $(A2)$. *Let* L' *be the RSS parameter of* $F(\boldsymbol{x})$ *at the sparsity level* $3k+K$. *Consider the HT-SVRG algorithm with hyper-parameters given in* (3.21). *Then the sequence* $\{\tilde{\boldsymbol{x}}^s\}_{s\geq 1}$ *recovers a* K*-sparse signal* $\boldsymbol{x}^*$ *with a geometric rate*

$$\mathbb{E}\big[\,\|\tilde{\boldsymbol{x}}^s-\boldsymbol{x}^*\|\,\big]\leq\sqrt{\frac{2L'}{\alpha}}\cdot\left(\frac{2}{3}\right)^{\frac{s}{2}}\big\|\tilde{\boldsymbol{x}}^0-\boldsymbol{x}^*\big\|+\sqrt{\frac{10\omega}{\alpha^2}\,\|\nabla_{3k+K}F(\boldsymbol{x}^*)\|}$$
$$+\left(\sqrt{\frac{2}{\alpha^3}}+\frac{3}{\alpha}\right)\|\nabla_{3k+K}F(\boldsymbol{x}^*)\|\,.$$

The proof can be found in Appendix 3.D.2.

Remark 8. The RSS parameter L' of $F(\boldsymbol{x})$ always ranges in $[\alpha, L]$, which is simply by definition.

Computational Complexity

We compare the computational complexity of HT-SVRG to that of projected gradient descent (PGD) studied in [75], which is a batch counterpart to HT-SVRG. First, we remark that the analysis of PGD is based on the smoothness parameter L' of $F(\boldsymbol{x})$ at sparsity level $2k+K$. We write $c'=L'/\alpha$. To achieve a given accuracy $\epsilon>0$, PGD requires $\mathcal{O}\left(c'\log(1/\epsilon)\right)$ iterations. Hence the total computational complexity is $\mathcal{O}\left(nc'd\log(1/\epsilon)\right)$. For HT-SVRG, in view of Corollary 3.6, the convergence coefficient is a constant. Hence, HT-SVRG needs $\mathcal{O}\left(\log(1/\epsilon)\right)$ iterations where we note that the error term $\|\nabla_{3k+K}F(\boldsymbol{x}^*)\|$ can be made as small as ϵ with sufficient samples (to be clarified in the sequel). In each stage, HT-SVRG computes a full gradient $\tilde{\boldsymbol{\mu}}$ followed by m times stochastic updates. Therefore, the total complexity of HT-SVRG is given by $\mathcal{O}\left((n+c)d\log(1/\epsilon)\right)$ by noting the fact $m=\mathcal{O}\left(c\right)$. In the scenario $c<n(c'-1)$, HT-SVRG significantly improves on PGD in terms of time cost.

3.4.3 Statistical Results

The last ingredient of our theorem is the term $\tau(\hat{\boldsymbol{x}})$ which measures how close the iterates could be to a given sparse signal $\hat{\boldsymbol{x}}$. With appropriate hyper-parameter settings, the quantity relies exclusively on $\|\nabla_{3k+K}F(\hat{\boldsymbol{x}})\|$, as suggested by (3.22). Thereby, this section is dedicated to characterizing $\|\nabla_{3k+K}F(\hat{\boldsymbol{x}})\|$. We will also give examples for which HT-SVRG is computationally more efficient than PGD. For the purpose of a concrete result, we study two problems: sparse linear regression and

sparse logistic regression. These are two of the most popular statistical models in the literature and have found a variety of applications in machine learning and statistics [119]. Notably, it is known that similar statistical results can be built for low-rank matrix regression, sparse precision matrix estimation, as suggested in [105, 2].

Sparse Linear Regression

For sparse linear regression, the observation model is given by

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{e}, \quad \|\boldsymbol{x}^*\|_0 \leq K,\ \|\boldsymbol{x}^*\| \leq \omega, \tag{3.23}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ is the design matrix, $\boldsymbol{y} \in \mathbb{R}^n$ is the response, $\boldsymbol{e} \in \mathbb{R}^n$ is some noise, and $\boldsymbol{x}^*$ is the K-sparse true parameter we hope to estimate from the knowledge of $\boldsymbol{A}$ and $\boldsymbol{y}$. Note that when we have the additional constraint $n \ll d$, the model above is exactly that of compressed sensing (3.12).

In order to (approximately) estimate the parameter, a natural approach is to optimize the following non-convex program:

$$\min_{\boldsymbol{x}}\ F(\boldsymbol{x}) := \frac{1}{2n}\sum_{i=1}^{n} \|y_i - \boldsymbol{a}_i \cdot \boldsymbol{x}\|^2, \quad \text{s.t. } \|\boldsymbol{x}\|_0 \leq K,\ \|\boldsymbol{x}\| \leq \omega. \tag{3.24}$$

For our analysis, we assume the following on the design matrix and the noise:

$(A3)$ $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n$ are independent and identically distributed (i.i.d.) Gaussian random vectors $N(\boldsymbol{0}, \boldsymbol{\Sigma})$. All the diagonal elements of $\boldsymbol{\Sigma}$ satisfy $\Sigma_{jj} \leq 1$. The noise $\boldsymbol{e}$ is independent of $\boldsymbol{A}$ and its entries are i.i.d. Gaussian random variables $N(0, \sigma^2)$.

Proposition 3.8. *Consider the sparse linear regression model* (3.23) *and the program* (3.24). *Assume* $(A3)$. *Then for a sparsity level* r,

- *with probability at least* $1 - \exp(-\mathrm{C}_0 n)$,

$$\alpha_r = \lambda_{\min}(\boldsymbol{\Sigma}) - \mathrm{C}_1\frac{r\log d}{n}, \quad L_r' = \lambda_{\max}(\boldsymbol{\Sigma}) + \mathrm{C}_2\frac{r\log d}{n};$$

- *with probability at least* $1 - \mathrm{C}_3 r/d$

$$L_r = \mathrm{C}_4 r \log d;$$

- *and with probability at least* $1 - \mathrm{C}_5/d$

$$\|\nabla_r F(\boldsymbol{x}^*)\| \leq \mathrm{C}_6 \sigma \sqrt{\frac{r \log d}{n}}, \quad \|\nabla_r F(\boldsymbol{x}_{\mathrm{opt}})\| \leq L_r' \|\boldsymbol{x}_{\mathrm{opt}} - \boldsymbol{x}^*\| + \mathrm{C}_6 \sigma \sqrt{\frac{r \log d}{n}}.$$

Above, $\lambda_{\min}(\boldsymbol{\Sigma})$ *and* $\lambda_{\max}(\boldsymbol{\Sigma})$ *are the minimum and maximum singular values of* $\boldsymbol{\Sigma}$ *respectively.*

We recall that α_r and L_r are involved in our assumptions $(A1)$ and $(A2)$, and L_r' is the RSS parameter of $F(\boldsymbol{x})$. The estimation for α_r, L_r' and $\|\nabla_r F(\boldsymbol{x}^*)\|$ follows from standard results in the literature [119], while that for L_r follows from Proposition E.1 in [13] by noting the fact that bounding L_r amounts to estimating $\max_i \|\mathcal{H}_r(\boldsymbol{a}_i)\|^2$. In order to estimate $\|\nabla_r F(\boldsymbol{x}_{\mathrm{opt}})\|$, notice that

$$\begin{aligned}
\|\nabla_r F(\boldsymbol{x}_{\mathrm{opt}})\| &\leq \|\nabla_r F(\boldsymbol{x}_{\mathrm{opt}}) - \nabla_r F(\boldsymbol{x}^*)\| + \|\nabla_r F(\boldsymbol{x}^*)\| \\
&\leq \|\nabla F(\boldsymbol{x}_{\mathrm{opt}}) - \nabla F(\boldsymbol{x}^*)\| + \|\nabla_r F(\boldsymbol{x}^*)\| \\
&\leq L_r' \|\boldsymbol{x}_{\mathrm{opt}} - \boldsymbol{x}^*\| + \|\nabla_r F(\boldsymbol{x}^*)\|,
\end{aligned}$$

where we use the definition of RSS in the last inequality.

Now we let $r = 3k + K = \mathrm{const} \cdot c^2 K$ and get $\alpha = \lambda_{\min}(\boldsymbol{\Sigma}) - \mathrm{C}_1 \frac{c^2 K \log d}{n}$, $L = \mathrm{C}_4 c^2 K \log d$. Suppose that $\lambda_{\min}(\boldsymbol{\Sigma}) = 2\mathrm{C}_4 (K \log d)^2$ and $n = q \cdot \frac{\mathrm{C}_1}{\mathrm{C}_4} K \log d$ with $q \geq 1$. Then our assumptions $(A1)$ and $(A2)$ are met with high probability with

$$\alpha = \mathrm{C}_4 (K \log d)^2, \ L = \mathrm{C}_4 (K \log d)^3, \ \text{and } c = K \log d.$$

For Corollary 3.6, as far as

$$s \geq \mathrm{C}_7 \log \left(\frac{F(\tilde{\boldsymbol{x}}^0) - F(\boldsymbol{x}_{\mathrm{opt}})}{\epsilon} \right), \ n = \mathrm{C}_7 \left(\omega \sigma\right)^2 \epsilon^{-2} K \log d,$$

we have

$$\mathbb{E}\big[F(\tilde{\boldsymbol{x}}^s) - F(\boldsymbol{x}_{\rm opt})\big] \le \epsilon + \frac{\lambda_{\max}(\boldsymbol{\Sigma})}{\lambda_{\min}(\boldsymbol{\Sigma})}\,\|\boldsymbol{x}_{\rm opt} - \boldsymbol{x}^*\| + \left(\frac{\lambda_{\max}(\boldsymbol{\Sigma})}{\lambda_{\min}(\boldsymbol{\Sigma})}\,\|\boldsymbol{x}_{\rm opt} - \boldsymbol{x}^*\|\right)^2$$

for some accuracy parameter $\epsilon > 0$. This suggests that it is possible for HT-SVRG to approximate a global optimum of (3.17) up to $\|\boldsymbol{x}_{\rm opt} - \boldsymbol{x}^*\|$, namely the statistical precision of the problem.

Returning to Corollary 3.7, to guarantee that

$$\mathbb{E}\big[\,\|\tilde{\boldsymbol{x}}^s - \boldsymbol{x}^*\|\,\big] \le \epsilon,$$

it suffices to pick

$$s \ge \mathrm{C}_8 \log(\omega\sqrt{c'}/\epsilon), \quad n = \mathrm{C}_8(\omega\sigma)^2\epsilon^{-4}K\log d.$$

Finally, we compare the computational cost to PGD. It is not hard to see that under the same situation $\lambda_{\min}(\boldsymbol{\Sigma}) = 2\mathrm{C}_4(K\log d)^2$ and $n = \frac{\mathrm{C}_1}{\mathrm{C}_4}K\log d$,

$$L' = \mathrm{C}_4(K\log d)^3,\ c' = K\log d,\ \text{provided that } \lambda_{\max}(\boldsymbol{\Sigma}) = \mathrm{C}_4(K\log d)^3 - \frac{\mathrm{C}_2\mathrm{C}_4}{\mathrm{C}_1}(K\log d)^2.$$

Thus $c < n(c'-1)$, i.e., HT-SVRG is more efficient than PGD. It is also possible to consider other regimes of the covariance matrix and the sample size, though we do not pursue it here.

Sparse Logistic Regression

For sparse logistic regression, the observation model is given by

$$\Pr(y_i \mid \boldsymbol{a}_i;\ \boldsymbol{x}^*) = \frac{1}{1+\exp(-y_i\boldsymbol{a}_i\cdot\boldsymbol{x}^*)}, \quad \|\boldsymbol{x}^*\|_0 \le K,\ \|\boldsymbol{x}\| \le \omega,\ \forall\, 1 \le i \le n, \tag{3.25}$$

where y_i is either 0 or 1. It then learns the parameter by minimizing the negative log-likelihood:

$$\min_{\boldsymbol{x}}\ F(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n}\log\left(1+\exp(-y_i\boldsymbol{a}_i\cdot\boldsymbol{x})\right), \quad \text{s.t. } \|\boldsymbol{x}\|_0 \le K,\ \|\boldsymbol{x}\| \le \omega. \tag{3.26}$$

There is a large body of work showing that the statistical property is rather analogous to that of linear regression. See, for example, [105]. In fact, the statistical results apply to generalized linear

models as well.

3.5 Experiments

In this section, we present a comprehensive empirical study for the proposed HT-SVRG algorithm on two tasks: sparse recovery (compressed sensing) and image classification. The experiments on sparse recovery is dedicated to verifying the theoretical results we presented, and we visualize the classification models learned by HT-SVRG to demonstrate the practical efficacy.

3.5.1 Sparse Recovery

To understand the practical behavior of our algorithm as well as to justify the theoretical analysis, we perform experiments on synthetic data. The experimental settings are as follows:

- **Data Generation.** The data dimension d is fixed as 256 and we generate an $n \times d$ Gaussian random sensing matrix $\boldsymbol{A}$ whose entries are i.i.d. with zero mean and variance $1/n$. Then 1000 K-sparse signals $\boldsymbol{x}^*$ are independently generated, where the support of each signal is uniformly chosen. That is, we run our algorithm and the baselines for 1000 trials. The measurements $\boldsymbol{y}$ for each signal $\boldsymbol{x}^*$ is obtained by $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^*$ which is noise free. In this way, we are able to study the convergence rate by plotting the logarithm of the objective value since the optimal objective value is known to be zero.

- **Baselines.** We mainly compare with two closely related algorithms: IHT and PGD. Both of them compute the full gradient of the least-squares loss followed by hard thresholding. Yet, PGD is more general, in the sense that it allows the sparsity parameter k to be larger than the true sparsity K ($k = K$ for IHT) and also considers a flexible step size η ($\eta = 1$ for IHT). Hence, PGD can be viewed as a batch counterpart to our method HT-SVRG.

- **Evaluation Metric.** We say a signal $\boldsymbol{x}^*$ is successfully recovered by a solution $\boldsymbol{x}$ if

$$\frac{\|\boldsymbol{x} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}^*\|} < 10^{-3}.$$

  In this way, we can compute the percentage of success over the 1000 trials for each algorithm.

- **Hyper-Parameters.** If not specified, we use $m = 3n$, $k = 9K$, and $S = 10000$ for HT-SVRG. We also use the heuristic step size $\eta = 2/\mathrm{svds}(\boldsymbol{A}\boldsymbol{A}^\top)$ for HT-SVRG and PGD, where $\mathrm{svds}(\boldsymbol{A}\boldsymbol{A}^\top)$ returns the largest singular value of the matrix $\boldsymbol{A}\boldsymbol{A}^\top$. Since for each stage, HT-SVRG computes the full gradient for $(2m/n + 1)$ times, we run the IHT and PGD for $(2m/n + 1)S$ iterations for fair comparison, i.e., all of the algorithms have the same number of full gradient evaluations.

Phase Transition

Our first simulation aims at offering a big picture on the recovery performance. To this end, we vary the number of measurements n from 1 to 256, roughly with a step size 8. We also study the performance with respect to the true sparsity parameter K, which ranges from 1 to 26, roughly with step size 2. The results are illustrated in Figure 3.1, where a brighter block means a higher percentage of success and the brightest ones indicate exact sparse recovery. It is apparent that PGD and HT-SVRG require fewer measurements for an accurate recovery than IHT, possibly due to the flexibility in choosing the sparsity parameter and the step size. We also observe that as a stochastic algorithm, HT-SVRG performs comparably to PGD. This suggests that HT-SVRG is an appealing solution to large-scale sparse learning problems in that HT-SVRG is computationally more efficient.

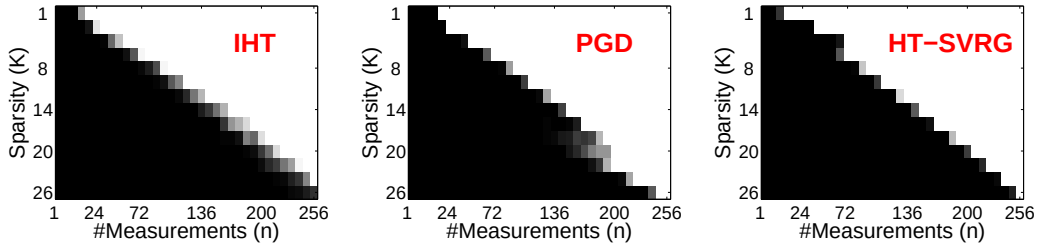


Figure 3.1: **Percentage of successful recovery under various sparsity and sample size.** The values range from 0 to 100, where a brighter color means a higher percentage of success (the brightest blocks correspond to the value of 100). PGD admits a higher percentage of recovery compared to IHT because it flexibly chooses the step size and sparsity parameter. As a stochastic variant, HT-SVRG performs comparably to the batch counterpart PGD.

In Figure 3.2, we exemplify some of the results obtained from HT-SVRG by plotting two kinds of curves: the success of percentage against the sample size n and that against the signal sparsity K. In this way, one can examine the detailed values and can determine the minimum sample size for a

particular sparsity. For instance, the left panel tells that to ensure that 80% percents of the 16-sparse signals are recovered, we have to collect 175 measurements. We can also learn from the right panel that using 232 measurements, any signal whose sparsity is 22 or less can be reliably recovered.

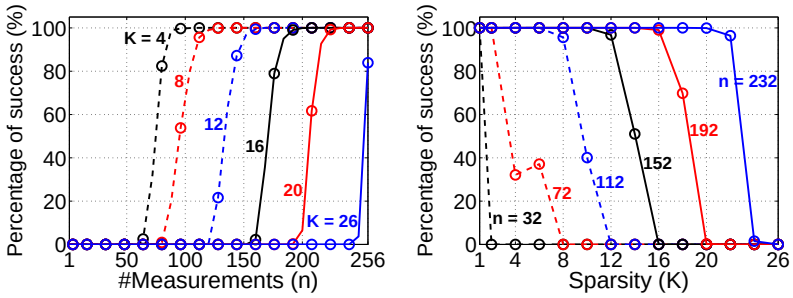


Figure 3.2: **Percentage of success of HT-SVRG against the number of measurements (left) and the sparsity (right).**

Based on the results in Figure 3.1 and Figure 3.2, we have an approximate estimation on the minimum requirement of the sample size which ensures accurate (or exact) recovery. Now we are to investigate how many measurements are needed to guarantee a success percentage of 95% and 99%. To this end, for each signal sparsity K, we look for the number of measurements n_0 from Figure 3.1 where 90 percents of success are achieved. Then we carefully enlarge n_0 with step size 1 and run the algorithms. The empirical results are recorded in Figure 3.3, where the circle markers represent the empirical results with different colors indicating different algorithms, e.g., red circle for empirical observation of HT-SVRG. Then we fit these empirical results by linear regression, which are plotted as solid or dashed lines. For example, the green line is a fitted model for IHT. We find that n is almost linear with K. Especially, the curve of HT-SVRG is nearly on top of that of PGD, which again verifies HT-SVRG is an attractive alternative to the batch method.

Influence of Hyper-Parameters

Next, we turn to investigate the influence of the hyper-parameters, i.e., the sparsity parameter k, update frequency m and step size η on the convergence behavior of HT-SVRG. We set the true sparsity $K = 4$ and collect 100 measurements for each groundtruth signal, i.e., $n = 100$. Note that the standard setting we employed is $k = 9K = 36$, $m = 3n = 300$ and $\eta = 2/\mathrm{svds}(\boldsymbol{A}\boldsymbol{A}^\top) \approx 0.3$. Each time we vary one of these parameters while fixing the other two, and the results are plotted in Figure 3.4. We point out that although the convergence result (Theorem 3.4) is deterministic, the

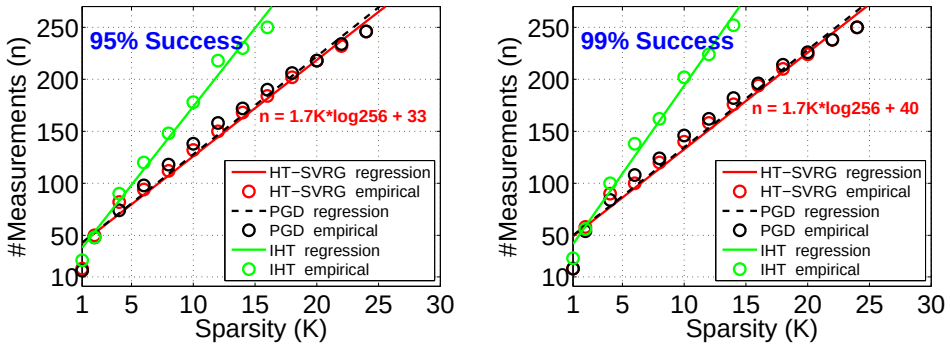


Figure 3.3: **Minimum number of measurements to achieve** 95% **and** 99% **percentage of success.** Red equation indicates the linear regression of HT-SVRG. The markers and curves for HT-SVRG are almost on top of PGD, which again justifies that HT-SVRG is an appealing stochastic alternative to the batch method PGD.

vanishing optimization error (Proposition 3.8) is guaranteed under a probabilistic argument. Hence, it is possible that for a specific configuration of parameters, 97% of the signals are exactly recovered but HT-SVRG fails on the remaining, as we have observed in, e.g., Figure 3.2. Clearly, we are not supposed to average all the results to examine the convergence rate. For our purpose, we set a threshold 95%, that is, we average over the success trials if more than 95% percents of the signals are exactly recovered. Otherwise, we say that the set of parameters cannot ensure convergence and we average over these failure signals which will give an illustration of divergence.

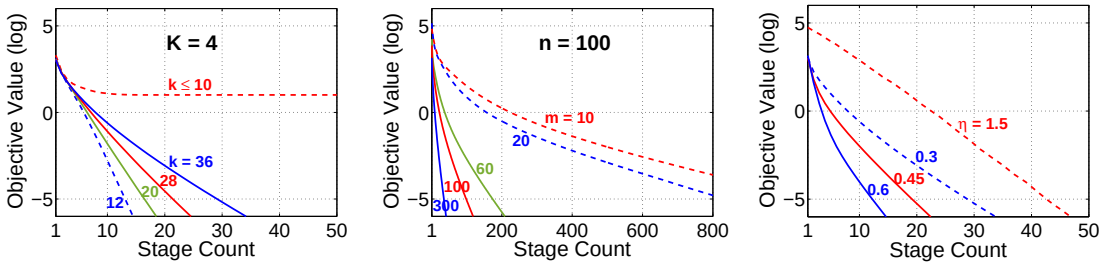


Figure 3.4: **Convergence of HT-SVRG with different parameters.** We have 100 measurements for the 256-dimensional signal where only 4 elements are non-zero. The standard setting is $k = 36$, $m = 300$ and $\eta = 0.3$. **Left:** If the sparsity parameter k is not large enough, HT-SVRG will not recover the signal. **Middle:** A small m leads to a frequent full gradient evaluation and hence slow convergence. **Right:** We observe divergence when $\eta \geq 3$.

The left panel of Figure 3.4 verifies the condition that k has to be larger than K, while the second panel shows the update frequency m can be reasonably small in the price of a slow convergence rate. Finally, the empirical study demonstrates that our heuristic choice $\eta = 0.3$ works well, and when

$\eta > 3$, the objective value exceeds 10^{120} within 3 stages (which cannot be depicted in the figure). For very small step sizes, we plot the convergence curve by gradually enlarging the update frequency m in Figure 3.5. The empirical results agree with Theorem 3.4 that for any $0 < \eta < 1/(4L)$, HT-SVRG converges as soon as m is large enough.

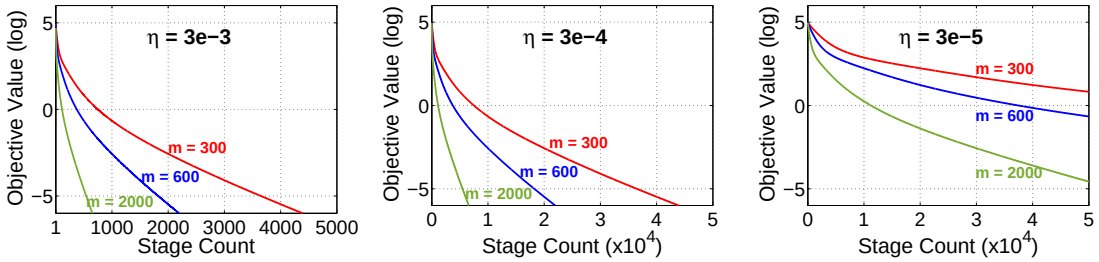


Figure 3.5: **Convergence behavior under small step size.** We observe that as long as we pick a sufficiently large value for m, HT-SVRG always converges. This is not surprising since our theorem guarantees for any $\eta < 1/(4L)$, HT-SVRG will converge if m is large enough. Also note that the geometric convergence rate is observed after certain iterations, e.g., for $\eta = 3 \times 10^{-5}$, the log(error) decreases linearly after 20 thousands iterations.

3.5.2 Classification

In addition to the application of sparse recovery, we illustrated that HT-SVRG can deal with binary classification by minimizing the sparse logistic regression problem (3.26). Here, we study the performance on a realistic image dataset MNIST[1], consisting of 60 thousands training samples and 10 thousands samples for testing. There is one digit on each image of size 28-by-28, hence totally 10 classes. Some of the images are shown in Figure 3.6.

Figure 3.6: **Sample images in the MNIST database.**

[1] `http://yann.lecun.com/exdb/mnist/`

The update frequency m is fixed as $m = 3n$. We compute the heuristic step size η as in the previous section, i.e., $\eta = 2/\mathrm{svds}(\boldsymbol{A}\boldsymbol{A}^\top) \approx 10^{-3}$. Since for the real-world dataset, the true sparsity is actually unknown, we tune the sparsity parameter k and study the performance of the algorithm.

First, we visualize five pair-wise models learned by HT-SVRG in Figure 3.7, where each row is associated with a binary classification task indicated by the two digits at the leading of the row, and the subsequent red-blue figures are used to illustrate the learned models under different sparsity parameter. For example, the third colorful figure depicted on the second row corresponds to recognizing a digit is "1" or "7" with the sparsity $k = 30$. In particular, for each pair, we label the small digit as positive and the large one as negative, and the blue and red pixels are the weights with positive and negative values respectively. Apparently, the models we learned are discriminative.

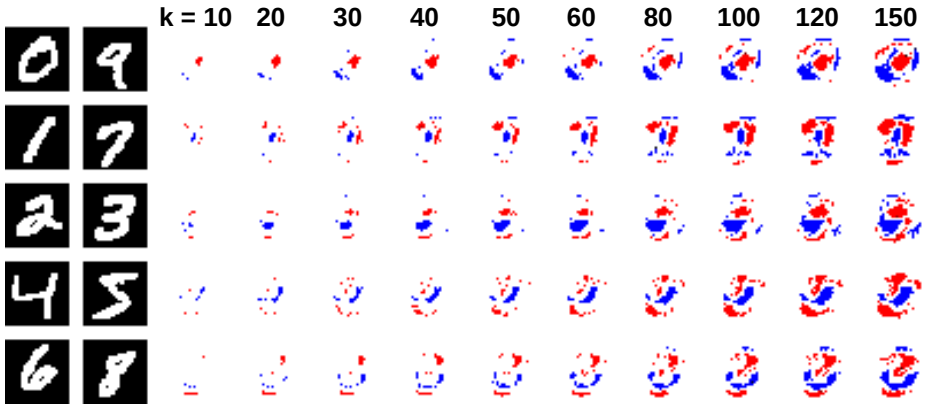


Figure 3.7: **Visualization of the models.** We visualize 5 models learned by HT-SVRG under different choices of sparsity shown on the top of each column. Note that the feature dimension is 784. From the top row to the bottom row, we illustrate the models of "0 vs 9", "1 vs 7", "2 vs 3", "4 vs 5" and "6 vs 8", where for each pair, we label the small digit as positive and the large one as negative. The red color represents negative weights while the blue pixels correspond with positive weights.

We also quantitatively show the convergence and prediction accuracy curves in Figure 3.8. Note that here, the y-axis is the objective value $F(\tilde{\boldsymbol{x}}^s)$ rather than $\log(F(\tilde{\boldsymbol{x}}^s) - F(\boldsymbol{x}_{\mathrm{opt}}))$, due to the fact that computing the exact optimum of (3.26) is NP-hard. Generally speaking, HT-SVRG converges quite fast and usually attains the minimum of objective value within 20 stages. It is not surprising to see that choosing a large quantity for the sparsity leads to a better (lower) objective value. However, in practice a small assignment for the sparsity, e.g., $k = 70$ facilitates an efficient computation while still suffices to ensure fast convergence and accurate prediction.

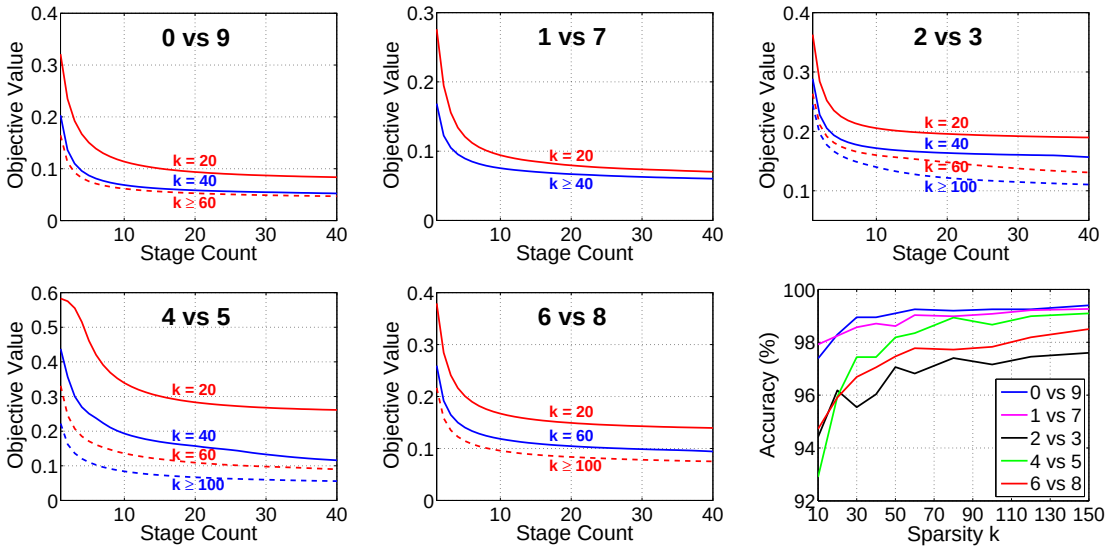


Figure 3.8: **Quantitative results on convergence and accuracy.** The first 5 figures demonstrate the convergence behavior of HT-SVRG for each binary classification task, where curves with different colors represent the objective value against number of stages under different sparsity k. Generally speaking, HT-SVRG converges within 20 stages which is a very fast rate. The last figure reflects the classification accuracy against the sparsity for all 5 classification tasks, where we find that for a moderate choice, e.g., $k = 70$, it already guarantees an accurate prediction (we recall the dimension is 784).

3.6 Conclusion and Open Problems

In this chapter, we have provided a tight bound on the deviation resulting from the hard thresholding operator, which underlies a vast volume of algorithms developed for sparsity-constrained problems. Our derived bound is universal over all choices of parameters and we have proved that it cannot be improved without further information on the signals. We have discussed the implications of our result to the community of compressed sensing and machine learning, and have demonstrated that the theoretical results of a number of popular algorithms in the literature can be advanced. In addition, we have devised a novel algorithm which tackles the problem of sparse learning in large-scale setting. We have elaborated that our algorithm is guaranteed to produce global optimal solution for prevalent statistical models only when it is equipped with the tight bound, hence justifying that the conventional bound is not applicable in the challenging scenario.

There are several interesting open problems. The first question to ask is whether one can establish sharp RIP condition or sharp phase transition for hard thresholding based algorithms such as IHT and CoSaMP with the tight bound. Moreover, compared to the hard thresholded SGD

method [110], HT-SVRG admits a vanishing optimization error. This poses a question of whether we are able to provably show the necessity of variance reduction for such a sparsity-constrained problem.

3.A Technical Lemmas

We present some useful lemmas that will be invoked by subsequent analysis.

Lemma 3.9. *Consider the HT-SVRG algorithm for a fixed stage s. Let $\hat{\boldsymbol{x}}$ be the target sparse vector. Let Ω be a support set such that* $\operatorname{supp}\left(\boldsymbol{x}^{t-1}\right) \cup \operatorname{supp}\left(\tilde{\boldsymbol{x}}\right) \cup \operatorname{supp}\left(\hat{\boldsymbol{x}}\right) \subseteq \Omega$. *Put $r = |\Omega|$. Assume $(A2)$. For all $1 \leq t \leq m$, denote $\boldsymbol{v}^t = \nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\tilde{\boldsymbol{x}}) + \tilde{\boldsymbol{\mu}}$. Then we have the following:*

$$\begin{aligned}\mathbb{E}_{i_t|\boldsymbol{x}^{t-1}}\left[\left\|\mathcal{P}_{\Omega}\left(\boldsymbol{v}^t\right)\right\|^2\right] \leq\ & 4L_r\left[F(\boldsymbol{x}^{t-1}) - F(\hat{\boldsymbol{x}})\right] + 4L_r\left[F(\tilde{\boldsymbol{x}}) - F(\hat{\boldsymbol{x}})\right] \\ & - 4L_r\left\langle \nabla F(\hat{\boldsymbol{x}}), \boldsymbol{x}^{t-1} + \tilde{\boldsymbol{x}} - 2\hat{\boldsymbol{x}}\right\rangle + 4\left\|\mathcal{P}_{\Omega}\left(\nabla F(\hat{\boldsymbol{x}})\right)\right\|^2.\end{aligned}$$

Proof. We have

$$\begin{aligned}\left\|\mathcal{P}_{\Omega}\left(\boldsymbol{v}^t\right)\right\|^2 =\ & \left\|\mathcal{P}_{\Omega}\left(\nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\tilde{\boldsymbol{x}}) + \tilde{\boldsymbol{\mu}}\right)\right\|^2 \\ \leq\ & 2\left\|\mathcal{P}_{\Omega}\left(\nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\hat{\boldsymbol{x}})\right)\right\|^2 + 2\left\|\mathcal{P}_{\Omega}\left(\nabla f_{i_t}(\tilde{\boldsymbol{x}}) - \nabla f_{i_t}(\hat{\boldsymbol{x}}) - \tilde{\boldsymbol{\mu}}\right)\right\|^2 \\ =\ & 2\left\|\mathcal{P}_{\Omega}\left(\nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\hat{\boldsymbol{x}})\right)\right\|^2 + 2\left\|\mathcal{P}_{\Omega}\left(\nabla f_{i_t}(\tilde{\boldsymbol{x}}) - \nabla f_{i_t}(\hat{\boldsymbol{x}})\right)\right\|^2 \\ & + 2\left\|\mathcal{P}_{\Omega}\left(\tilde{\boldsymbol{\mu}}\right)\right\|^2 - 4\left\langle \mathcal{P}_{\Omega}\left(\nabla f_{i_t}(\tilde{\boldsymbol{x}}) - \nabla f_{i_t}(\hat{\boldsymbol{x}})\right), \mathcal{P}_{\Omega}\left(\tilde{\boldsymbol{\mu}}\right)\right\rangle \\ \overset{\xi_1}{=}\ & 2\left\|\mathcal{P}_{\Omega}\left(\nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\hat{\boldsymbol{x}})\right)\right\|^2 + 2\left\|\mathcal{P}_{\Omega}\left(\nabla f_{i_t}(\tilde{\boldsymbol{x}}) - \nabla f_{i_t}(\hat{\boldsymbol{x}})\right)\right\|^2 \\ & + 2\left\|\mathcal{P}_{\Omega}\left(\tilde{\boldsymbol{\mu}}\right)\right\|^2 - 4\left\langle \nabla f_{i_t}(\tilde{\boldsymbol{x}}) - \nabla f_{i_t}(\hat{\boldsymbol{x}}), \mathcal{P}_{\Omega}\left(\tilde{\boldsymbol{\mu}}\right)\right\rangle \\ \overset{\xi_2}{\leq}\ & 4L_r\left[f_{i_t}(\boldsymbol{x}^{t-1}) - f_{i_t}(\hat{\boldsymbol{x}}) - \left\langle \nabla f_{i_t}(\hat{\boldsymbol{x}}), \boldsymbol{x}^{t-1} - \hat{\boldsymbol{x}}\right\rangle\right] \\ & + 4L_r\left[f_{i_t}(\tilde{\boldsymbol{x}}) - f_{i_t}(\hat{\boldsymbol{x}}) - \left\langle \nabla f_{i_t}(\hat{\boldsymbol{x}}), \tilde{\boldsymbol{x}} - \hat{\boldsymbol{x}}\right\rangle\right] \\ & + 2\left\|\mathcal{P}_{\Omega}\left(\tilde{\boldsymbol{\mu}}\right)\right\|^2 - 4\left\langle \nabla f_{i_t}(\tilde{\boldsymbol{x}}) - \nabla f_{i_t}(\hat{\boldsymbol{x}}), \mathcal{P}_{\Omega}\left(\tilde{\boldsymbol{\mu}}\right)\right\rangle,\end{aligned}$$

where ξ_1 is by algebra, ξ_2 applies Lemma 2.14 and the fact that $|\Omega| = r$.

Taking the conditional expectation, we obtain the following:

$$\begin{aligned}
&\mathbb{E}_{i_t|\boldsymbol{x}^{t-1}}\left[\left\|\mathcal{P}_{\Omega}\left(\boldsymbol{v}^{t}\right)\right\|^{2}\right] \\
&\leq 4L_r\left[F(\boldsymbol{x}^{t-1})-F(\hat{\boldsymbol{x}})\right]+4L_r\left[F(\tilde{\boldsymbol{x}})-F(\hat{\boldsymbol{x}})\right] \\
&\quad -4L_r\left\langle\nabla F(\hat{\boldsymbol{x}}),\boldsymbol{x}^{t-1}+\tilde{\boldsymbol{x}}-2\hat{\boldsymbol{x}}\right\rangle+2\left\langle 2\mathcal{P}_{\Omega}\left(\nabla F(\hat{\boldsymbol{x}})\right)-\mathcal{P}_{\Omega}\left(\tilde{\boldsymbol{\mu}}\right),\mathcal{P}_{\Omega}\left(\tilde{\boldsymbol{\mu}}\right)\right\rangle \\
&=4L_r\left[F(\boldsymbol{x}^{t-1})-F(\hat{\boldsymbol{x}})\right]+4L_r\left[F(\tilde{\boldsymbol{x}})-F(\hat{\boldsymbol{x}})\right] \\
&\quad -4L_r\left\langle\nabla F(\hat{\boldsymbol{x}}),\boldsymbol{x}^{t-1}+\tilde{\boldsymbol{x}}-2\hat{\boldsymbol{x}}\right\rangle+\left\|2\mathcal{P}_{\Omega}\left(\nabla F(\hat{\boldsymbol{x}})\right)\right\|^{2} \\
&\quad -\left\|2\mathcal{P}_{\Omega}\left(\nabla F(\hat{\boldsymbol{x}})\right)-\mathcal{P}_{\Omega}\left(\tilde{\boldsymbol{\mu}}\right)\right\|^{2}-\left\|\mathcal{P}_{\Omega}\left(\tilde{\boldsymbol{\mu}}\right)\right\|^{2} \\
&\leq 4L_r\left[F(\boldsymbol{x}^{t-1})-F(\hat{\boldsymbol{x}})\right]+4L_r\left[F(\tilde{\boldsymbol{x}})-F(\hat{\boldsymbol{x}})\right] \\
&\quad -4L_r\left\langle\nabla F(\hat{\boldsymbol{x}}),\boldsymbol{x}^{t-1}+\tilde{\boldsymbol{x}}-2\hat{\boldsymbol{x}}\right\rangle+4\left\|\mathcal{P}_{\Omega}\left(\nabla F(\hat{\boldsymbol{x}})\right)\right\|^{2}.
\end{aligned}$$

The proof is complete. □

Corollary 3.10. *Assume the same conditions as in Lemma 3.9. If* $\nabla F(\hat{\boldsymbol{x}})=0$*, we have*

$$\mathbb{E}_{i_t|\boldsymbol{x}^{t-1}}\left[\left\|\mathcal{P}_{\Omega}\left(\boldsymbol{v}^{t}\right)\right\|^{2}\right]\leq 4L_r\left[F(\boldsymbol{x}^{t-1})+F(\tilde{\boldsymbol{x}})-2F(\hat{\boldsymbol{x}})\right].$$

3.B Proofs for Section 3.2

3.B.1 Proof of Theorem 3.1

Proof. The result is true for the trivial case that $\boldsymbol{b}$ is a zero vector. In the following, we assume that $\boldsymbol{b}$ is not a zero vector. Denote

$$\boldsymbol{w}:=\mathcal{H}_k\left(\boldsymbol{b}\right).$$

Let Ω be the support set of $\boldsymbol{w}$ and let $\overline{\Omega}$ be its complement. We immediately have $\mathcal{P}_{\Omega}\left(\boldsymbol{b}\right)=\boldsymbol{w}$.

Let Ω' be the support set of $\boldsymbol{x}$. For the sake of simplicity, let us split the vector $\boldsymbol{b}$ as follows:

$$\begin{aligned}
\boldsymbol{b}_1&=\mathcal{P}_{\Omega\setminus\Omega'}\left(\boldsymbol{b}\right), \quad \boldsymbol{b}_2=\mathcal{P}_{\Omega\cap\Omega'}\left(\boldsymbol{b}\right), \\
\boldsymbol{b}_3&=\mathcal{P}_{\overline{\Omega}\setminus\Omega'}\left(\boldsymbol{b}\right), \quad \boldsymbol{b}_4=\mathcal{P}_{\overline{\Omega}\cap\Omega'}\left(\boldsymbol{b}\right).
\end{aligned}$$

Likewise, we denote

$$\boldsymbol{w}_1 = \mathcal{P}_{\Omega\setminus\Omega'}(\boldsymbol{w}), \quad \boldsymbol{w}_2 = \mathcal{P}_{\Omega\cap\Omega'}(\boldsymbol{w}), \quad \boldsymbol{w}_3 = \mathcal{P}_{\overline{\Omega}\setminus\Omega'}(\boldsymbol{w}) = \boldsymbol{0}, \quad \boldsymbol{w}_4 = \mathcal{P}_{\overline{\Omega}\cap\Omega'}(\boldsymbol{w}) = \boldsymbol{0},$$
$$\boldsymbol{x}_1 = \mathcal{P}_{\Omega\setminus\Omega'}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{x}_2 = \mathcal{P}_{\Omega\cap\Omega'}(\boldsymbol{x}), \quad \boldsymbol{x}_3 = \mathcal{P}_{\overline{\Omega}\setminus\Omega'}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{x}_4 = \mathcal{P}_{\overline{\Omega}\cap\Omega'}(\boldsymbol{x}).$$

Due to the hard thresholding, we have

$$\boldsymbol{w}_1 = \boldsymbol{b}_1, \quad \boldsymbol{w}_2 = \boldsymbol{b}_2.$$

In this way, by simple algebra we have

$$\|\boldsymbol{w}-\boldsymbol{x}\|^2 = \|\boldsymbol{b}_1\|^2 + \|\boldsymbol{b}_2-\boldsymbol{x}_2\|^2 + \|\boldsymbol{x}_4\|^2,$$
$$\|\boldsymbol{b}-\boldsymbol{x}\|^2 = \|\boldsymbol{b}_1\|^2 + \|\boldsymbol{b}_2-\boldsymbol{x}_2\|^2 + \|\boldsymbol{b}_3\|^2 + \|\boldsymbol{b}_4-\boldsymbol{x}_4\|^2.$$

Our goal is to estimate the maximum of $\|\boldsymbol{w}-\boldsymbol{x}\|^2 / \|\boldsymbol{b}-\boldsymbol{x}\|^2$. It is easy to show that when attaining the maximum value, $\|\boldsymbol{b}_3\|$ must be zero since otherwise one may decrease this term to make the objective larger. Hence, maximizing $\|\boldsymbol{w}-\boldsymbol{x}\|^2 / \|\boldsymbol{b}-\boldsymbol{x}\|^2$ amounts to estimating the upper bound of the following over all choices of $\boldsymbol{x}$ and $\boldsymbol{b}$:

$$\gamma := \frac{\|\boldsymbol{b}_1\|^2 + \|\boldsymbol{b}_2-\boldsymbol{x}_2\|^2 + \|\boldsymbol{x}_4\|^2}{\|\boldsymbol{b}_1\|^2 + \|\boldsymbol{b}_2-\boldsymbol{x}_2\|^2 + \|\boldsymbol{b}_4-\boldsymbol{x}_4\|^2}. \tag{3.27}$$

Firstly, we consider the case of $\|\boldsymbol{b}_1\| = 0$, which means $\Omega = \Omega'$ implying $\gamma = 1$. In the following, we consider $\|\boldsymbol{b}_1\| \neq 0$. In particular, we consider $\gamma > 1$ since we are interested in the maximum value of γ.

Arranging (3.27) we obtain

$$(\gamma-1)\|\boldsymbol{b}_2-\boldsymbol{x}_2\|^2 + \gamma\|\boldsymbol{b}_4-\boldsymbol{x}_4\|^2 - \|\boldsymbol{x}_4\|^2 + (\gamma-1)\|\boldsymbol{b}_1\|^2 = 0. \tag{3.28}$$

Let us fix $\boldsymbol{b}$ and define the function

$$G(\boldsymbol{x}_2,\boldsymbol{x}_4) = (\gamma-1)\|\boldsymbol{b}_2-\boldsymbol{x}_2\|^2 + \gamma\|\boldsymbol{b}_4-\boldsymbol{x}_4\|^2 - \|\boldsymbol{x}_4\|^2 + (\gamma-1)\|\boldsymbol{b}_1\|^2.$$

Thus, (3.28) indicates that $G(\boldsymbol{x}_2, \boldsymbol{x}_4)$ can attain the objective value of zero. Note that $G(\boldsymbol{x}_2, \boldsymbol{x}_4)$ is a quadratic function and its gradient and Hessian matrix can be computed as follows:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{x}_2} G(\boldsymbol{x}_2, \boldsymbol{x}_4) &= 2(\gamma - 1)(\boldsymbol{x}_2 - \boldsymbol{b}_2), \\
\frac{\partial}{\partial \boldsymbol{x}_4} G(\boldsymbol{x}_2, \boldsymbol{x}_4) &= 2\gamma(\boldsymbol{x}_4 - \boldsymbol{b}_4) - 2\boldsymbol{x}_4, \\
\nabla^2 G(\boldsymbol{x}_2, \boldsymbol{x}_4) &= 2(\gamma - 1)\boldsymbol{I},
\end{aligned}$$

where $\boldsymbol{I}$ is the identity matrix. Since the Hessian matrix is positive definite, $G(\boldsymbol{x}_2, \boldsymbol{x}_4)$ attains the global minimum at the stationary point, which is given by

$$\boldsymbol{x}_2^* = \boldsymbol{b}_2, \quad \boldsymbol{x}_4^* = \frac{\gamma}{\gamma - 1}\boldsymbol{b}_4,$$

resulting in the minimum objective value

$$G(\boldsymbol{x}_2^*, \boldsymbol{x}_4^*) = \frac{\gamma}{1 - \gamma} \left\| \boldsymbol{b}_4 \right\|^2 + (\gamma - 1) \left\| \boldsymbol{b}_1 \right\|^2.$$

In order to guarantee the feasible set of (3.28) is non-empty, we require that

$$G(\boldsymbol{x}_2^*, \boldsymbol{x}_4^*) \leq 0,$$

implying

$$\left\| \boldsymbol{b}_1 \right\|^2 \gamma^2 - (2 \left\| \boldsymbol{b}_1 \right\|^2 + \left\| \boldsymbol{b}_4 \right\|^2)\gamma + \left\| \boldsymbol{b}_1 \right\|^2 \leq 0.$$

Solving the above inequality with respect to γ, we obtain

$$\gamma \leq 1 + \frac{\left\| \boldsymbol{b}_4 \right\|^2 + \sqrt{\left(4 \left\| \boldsymbol{b}_1 \right\|^2 + \left\| \boldsymbol{b}_4 \right\|^2 \right) \left\| \boldsymbol{b}_4 \right\|^2}}{2 \left\| \boldsymbol{b}_1 \right\|^2}. \tag{3.29}$$

To derive an upper bound that is uniform over the choice of $\boldsymbol{b}$, we recall that $\boldsymbol{b}_1$ contains the largest absolute elements of $\boldsymbol{b}$ while $\boldsymbol{b}_4$ has smaller values. In particular, the averaged value of $\boldsymbol{b}_4$ is no

greater than that of $\boldsymbol{b}_1$ in magnitude, i.e.,

$$\frac{\|\boldsymbol{b}_4\|^2}{\|\boldsymbol{b}_4\|_0} \leq \frac{\|\boldsymbol{b}_1\|^2}{\|\boldsymbol{b}_1\|_0}.$$

Note that $\|\boldsymbol{b}_1\|_0 = k - \|\boldsymbol{b}_2\|_0 = k - (K - \|\boldsymbol{b}_4\|_0)$. Hence, combining with the fact that $0 \leq \|\boldsymbol{b}_4\|_0 \leq \min\{K, d-k\}$ and optimizing over $\|\boldsymbol{b}_4\|_0$ gives

$$\|\boldsymbol{b}_4\|^2 \leq \frac{\min\{K, d-k\}}{k - K + \min\{K, d-k\}} \|\boldsymbol{b}_1\|^2 .$$

Plugging back to (3.29), we finally obtain

$$\gamma \leq 1 + \frac{\rho + \sqrt{(4+\rho)\,\rho}}{2}, \quad \rho = \frac{\min\{K, d-k\}}{k - K + \min\{K, d-k\}}.$$

The proof is complete. □

3.C Proofs for Section 3.3

3.C.1 Proof of Theorem 3.2

We follow the proof pipeline of [20] and only remark the difference of our proof and theirs, i.e., where Theorem 3.1 applies. In case of possible confusion due to notation, we follow the symbols in Blumensath and Davies. One may refer to that article for a complete proof.

The first difference occurs in Eq. (22) of [20], where they reached

$$\text{(Old)} \quad \left\| \boldsymbol{x}^s - \boldsymbol{x}^{[n+1]} \right\| \leq 2 \left\| \boldsymbol{x}^s_{B^{n+1}} - \boldsymbol{a}^{[n+1]}_{B^{n+1}} \right\|,$$

while Theorem 3.1 gives

$$\text{(New)} \quad \left\| \boldsymbol{x}^s - \boldsymbol{x}^{[n+1]} \right\| \leq \sqrt{\nu} \left\| \boldsymbol{x}^s_{B^{n+1}} - \boldsymbol{a}^{[n+1]}_{B^{n+1}} \right\|.$$

Combining this new inequality and Eq. (23) therein, we obtain

$$\left\| \boldsymbol{x}^s - \boldsymbol{x}^{[n+1]} \right\| \leq \sqrt{\nu} \left\| (\boldsymbol{I} - \boldsymbol{\Phi}^\top_{B^{n+1}} \boldsymbol{\Phi}_{B^{n+1}}) \boldsymbol{r}^{[n]}_{B^{n+1}} \right\| + \sqrt{\nu} \left\| (\boldsymbol{\Phi}^\top_{B^{n+1}} \boldsymbol{\Phi}_{B^{n+1} \setminus B^{n+1}}) \boldsymbol{r}^{[n]}_{B^{n+1} \setminus B^{n+1}} \right\|.$$

By noting the fact that $\left|B^n \cup B^{n+1}\right| \leq 2s + s^*$ where s^* denotes the sparsity of the global optimum and following their reasoning of Eq. (24) and (25), we have a new bound for Eq. (26):

$$\text{(New)} \quad \left\|\boldsymbol{r}^{[n+1]}\right\| \leq \sqrt{2\nu}\delta_{2s+s^*} \left\|\boldsymbol{r}^{[n]}\right\| + \sqrt{(1+\delta_{s+s^*})\nu}\,\|\boldsymbol{e}\|\,.$$

Now our result follows by setting the coefficient of $\left\|\boldsymbol{r}^{[n]}\right\|$ to 0.5. Note that specifying $\nu = 4$ gives the result of [20].

3.C.2 Proof of Theorem 3.3

We follow the proof technique of Theorem 6.27 in [62] which gives the best known RIP condition for the CoSaMP algorithm to date. Since most of the reasoning is similar, we only point out the difference of our proof and theirs, i.e., where Theorem 3.1 applies. In case of confusion by notation, we follow the symbols used in [62]. The reader may refer to that book for a complete proof.

The first difference is in Eq. (6.49) of [62]. Note that to derive this inequality, Foucart and Rauhut invoked the conventional bound (3.1), which gives

$$\text{(Old)} \quad \left\|\boldsymbol{x}_S - \boldsymbol{x}^{n+1}\right\|^2 \leq \left\|(\boldsymbol{x}_S - \boldsymbol{u}^{n+1})_{\overline{U^{n+1}}}\right\|^2 + 4\left\|(\boldsymbol{x}_S - \boldsymbol{u}^{n+1})_{U^{n+1}}\right\|^2,$$

while utilizing Theorem 3.1 gives

$$\text{(New)} \quad \left\|\boldsymbol{x}_S - \boldsymbol{x}^{n+1}\right\|^2 \leq \left\|(\boldsymbol{x}_S - \boldsymbol{u}^{n+1})_{\overline{U^{n+1}}}\right\|^2 + \nu\left\|(\boldsymbol{x}_S - \boldsymbol{u}^{n+1})_{U^{n+1}}\right\|^2.$$

Combining this new inequality with Eq. (6.50) and Eq. (6.51) therein, we obtain

$$\begin{aligned}\left\|\boldsymbol{x}_S - \boldsymbol{x}^{n+1}\right\| \leq{}& \sqrt{2}\delta_{3s+s^*}\sqrt{\frac{1+(\nu-1)\delta_{3s+s^*}^2}{1-\delta_{3s+s^*}^2}}\,\|\boldsymbol{x}^n - \boldsymbol{x}_S\| \\ &+ \sqrt{2}\delta_{3s+s^*}\sqrt{\frac{1+(\nu-1)\delta_{3s+s^*}^2}{1-\delta_{3s+s^*}^2}}\,\left\|(\boldsymbol{A}^*\boldsymbol{e}')_{(S\cup S^n)\Delta T^{n+1}}\right\| \\ &+ \frac{2}{1-\delta_{3s+s^*}}\left\|(\boldsymbol{A}^*\boldsymbol{e}')_{U^{n+1}}\right\|,\end{aligned}$$

where s^* denotes the sparsity of the optimum. Our new bound follows by setting the coefficient of $\|\boldsymbol{x}^n - \boldsymbol{x}_S\|$ to 0.5 and solving the resultant equation. Note that setting $\nu = 4$ gives the old bound

of Foucart and Rauhut.

3.D Proofs for Section 3.4

3.D.1 Proof of Theorem 3.4

Proof. Fix a stage s. Let us denote

$$\boldsymbol{v}^t = \nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\tilde{\boldsymbol{x}}) + \tilde{\boldsymbol{\mu}},$$

so that

$$\boldsymbol{b}^t = \boldsymbol{x}^{t-1} - \eta \boldsymbol{v}^t.$$

By specifying $\Omega = \operatorname{supp}\left(\boldsymbol{x}^{t-1}\right) \cup \operatorname{supp}\left(\boldsymbol{x}^t\right) \cup \operatorname{supp}\left(\tilde{\boldsymbol{x}}\right) \cup \operatorname{supp}\left(\hat{\boldsymbol{x}}\right)$, it follows that

$$\boldsymbol{r}^t = \mathcal{H}_k\left(\boldsymbol{b}^t\right) = \mathcal{H}_k\left(\mathcal{P}_\Omega\left(\boldsymbol{b}^t\right)\right).$$

Thus, the Euclidean distance of $\boldsymbol{x}^t$ and $\hat{\boldsymbol{x}}$ can be bounded as follows:

$$\left\|\boldsymbol{x}^t - \hat{\boldsymbol{x}}\right\|^2 \leq \left\|\boldsymbol{r}^t - \hat{\boldsymbol{x}}\right\|^2 = \left\|\mathcal{H}_k\left(\mathcal{P}_\Omega\left(\boldsymbol{b}^t\right)\right) - \hat{\boldsymbol{x}}\right\|^2 \leq \nu \left\|\mathcal{P}_\Omega\left(\boldsymbol{b}^t\right) - \hat{\boldsymbol{x}}\right\|^2, \tag{3.30}$$

where the first inequality holds because $\boldsymbol{x}^t = \Pi_\omega(\boldsymbol{r}^t)$ and $\|\hat{\boldsymbol{x}}\| \leq \omega$. We also have

$$\begin{aligned}\left\|\mathcal{P}_\Omega\left(\boldsymbol{b}^t\right) - \hat{\boldsymbol{x}}\right\|^2 &= \left\|\boldsymbol{x}^{t-1} - \hat{\boldsymbol{x}} - \eta \mathcal{P}_\Omega\left(\boldsymbol{v}^t\right)\right\|^2 \\ &= \left\|\boldsymbol{x}^{t-1} - \hat{\boldsymbol{x}}\right\|^2 + \eta^2 \left\|\mathcal{P}_\Omega\left(\boldsymbol{v}^t\right)\right\|^2 - 2\eta \left\langle \boldsymbol{x}^{t-1} - \hat{\boldsymbol{x}}, \boldsymbol{v}^t \right\rangle,\end{aligned}$$

where the second equality uses the fact that $\left\langle \boldsymbol{x}^{t-1} - \hat{\boldsymbol{x}}, \mathcal{P}_\Omega\left(\boldsymbol{v}^t\right)\right\rangle = \left\langle \boldsymbol{x}^{t-1} - \hat{\boldsymbol{x}}, \boldsymbol{v}^t \right\rangle$. The first term will be preserved for mathematical induction. The third term is easy to manipulate thanks to the unbiasedness of $\boldsymbol{v}^t$. For the second term, we use Lemma 3.9 to bound it. Put them together,

conditioning on $\boldsymbol{x}^{t-1}$ and taking the expectation over i_t for (3.30), we have

$$
\begin{aligned}
&\mathbb{E}_{i_t|\boldsymbol{x}^{t-1}}\Big[\left\|\boldsymbol{x}^t-\hat{\boldsymbol{x}}\right\|^2\Big] \\
\overset{\xi_1}{\leq}\ &\nu\left\|\boldsymbol{x}^{t-1}-\hat{\boldsymbol{x}}\right\|^2+4\nu\eta^2L\left[F(\boldsymbol{x}^{t-1})-F(\hat{\boldsymbol{x}})+F(\tilde{\boldsymbol{x}})-F(\hat{\boldsymbol{x}})\right]-2\nu\eta\left\langle\boldsymbol{x}^{t-1}-\hat{\boldsymbol{x}},\nabla F(\boldsymbol{x}^{t-1})\right\rangle \\
&-4\nu\eta^2L\left\langle\nabla F(\hat{\boldsymbol{x}}),\boldsymbol{x}^{t-1}+\tilde{\boldsymbol{x}}-2\hat{\boldsymbol{x}}\right\rangle+4\nu\eta^2\left\|\mathcal{P}_\Omega\left(\nabla F(\hat{\boldsymbol{x}})\right)\right\|^2 \\
\overset{\xi_2}{\leq}\ &\nu(1-\eta\alpha)\left\|\boldsymbol{x}^{t-1}-\hat{\boldsymbol{x}}\right\|^2-2\nu\eta(1-2\eta L)\left[F(\boldsymbol{x}^{t-1})-F(\hat{\boldsymbol{x}})\right]+4\nu\eta^2L\left[F(\tilde{\boldsymbol{x}})-F(\hat{\boldsymbol{x}})\right] \\
&+4\nu\eta^2L\left\|\mathcal{P}_\Omega\left(\nabla F(\hat{\boldsymbol{x}})\right)\right\|\cdot\left\|\boldsymbol{x}^{t-1}+\tilde{\boldsymbol{x}}-2\hat{\boldsymbol{x}}\right\|+4\nu\eta^2\left\|\mathcal{P}_\Omega\left(\nabla F(\hat{\boldsymbol{x}})\right)\right\|^2 \\
\leq\ &\nu(1-\eta\alpha)\left\|\boldsymbol{x}^{t-1}-\hat{\boldsymbol{x}}\right\|^2-2\nu\eta(1-2\eta L)\left[F(\boldsymbol{x}^{t-1})-F(\hat{\boldsymbol{x}})\right] \\
&+4\nu\eta^2L\left[F(\tilde{\boldsymbol{x}})-F(\hat{\boldsymbol{x}})\right]+4\nu\eta^2Q'(4L\omega+Q')
\end{aligned}
$$

where ξ_1 applies Lemma 3.9, ξ_2 applies Assumption $(A1)$ and we write $Q':=\|\nabla_{3k+K}F(\hat{\boldsymbol{x}})\|$ for brevity.

Now summing over the inequalities over $t=1,2,\cdots,m$, conditioning on $\tilde{\boldsymbol{x}}$ and taking the expectation with respect to $\mathcal{I}^s=\{i_1,i_2,\cdots,i_m\}$, we have

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{I}^s|\tilde{\boldsymbol{x}}}\Big[\left\|\boldsymbol{x}^m-\hat{\boldsymbol{x}}\right\|^2\Big] \\
\leq\ &\left[\nu(1-\eta\alpha)-1\right]\mathbb{E}_{\mathcal{I}^s|\tilde{\boldsymbol{x}}}\sum_{t=1}^m\left\|\boldsymbol{x}^{t-1}-\hat{\boldsymbol{x}}\right\|^2+\left\|\boldsymbol{x}^0-\hat{\boldsymbol{x}}\right\|^2+4\nu\eta^2Q'(4L\omega+Q')m \\
&-2\nu\eta(1-2\eta L)\mathbb{E}_{\mathcal{I}^s|\tilde{\boldsymbol{x}}}\sum_{t=1}^m\left[F(\boldsymbol{x}^{t-1})-F(\hat{\boldsymbol{x}})\right]+4\nu\eta^2Lm\left[F(\tilde{\boldsymbol{x}})-F(\hat{\boldsymbol{x}})\right] \\
=\ &\left[\nu(1-\eta\alpha)-1\right]m\mathbb{E}_{\mathcal{I}^s,j^s|\tilde{\boldsymbol{x}}}\left\|\tilde{\boldsymbol{x}}^s-\hat{\boldsymbol{x}}\right\|^2+\left\|\tilde{\boldsymbol{x}}-\hat{\boldsymbol{x}}\right\|^2+4\nu\eta^2Q'(4L\omega+Q')m \\
&-2\nu\eta(1-2\eta L)m\mathbb{E}_{\mathcal{I}^s,j^s|\tilde{\boldsymbol{x}}}\left[F(\tilde{\boldsymbol{x}}^s)-F(\hat{\boldsymbol{x}})\right]+4\nu\eta^2Lm\left[F(\tilde{\boldsymbol{x}})-F(\hat{\boldsymbol{x}})\right] \\
\leq\ &\left[\nu(1-\eta\alpha)-1\right]m\mathbb{E}_{\mathcal{I}^s,j^s|\tilde{\boldsymbol{x}}}\left\|\tilde{\boldsymbol{x}}^s-\hat{\boldsymbol{x}}\right\|^2+\left(\frac{2}{\alpha}+4\nu\eta^2Lm\right)\left[F(\tilde{\boldsymbol{x}})-F(\hat{\boldsymbol{x}})\right] \\
&-2\nu\eta(1-2\eta L)m\mathbb{E}_{\mathcal{I}^s,j^s|\tilde{\boldsymbol{x}}}\left[F(\tilde{\boldsymbol{x}}^s)-F(\hat{\boldsymbol{x}})\right]+4\nu\eta^2Q'(4L\omega+Q')m+2Q'\omega/\alpha, \qquad (3.31)
\end{aligned}
$$

where we recall that j^s is the randomly chosen index used to determine $\tilde{\boldsymbol{x}}^s$ (see Algorithm 1). The last inequality holds due to the RSC condition and $\left\|\boldsymbol{x}^t\right\|\leq\omega$. For brevity, we write

$$
Q:=4\nu\eta^2Q'(4L\omega+Q')m+2Q'\omega/\alpha,\quad Q'=\left\|\nabla_{3k+K}F(\hat{\boldsymbol{x}})\right\|.
$$

Based on (3.31), we discuss two cases to examine the convergence of the algorithm.

Case 1. $\nu(1-\eta\alpha) \leq 1$. This immediately results in

$$\begin{aligned}
&\mathbb{E}_{\mathcal{I}^s|\tilde{\boldsymbol{x}}}\left[\, \|\boldsymbol{x}^m - \hat{\boldsymbol{x}}\|^2 \,\right] \\
&\leq \left(\frac{2}{\alpha} + 4\nu\eta^2 Lm\right)[F(\tilde{\boldsymbol{x}}) - F(\hat{\boldsymbol{x}})] - 2\nu\eta(1-2\eta L)m\, \mathbb{E}_{\mathcal{I}^s,j^s|\tilde{\boldsymbol{x}}}\left[F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}})\right] + Q,
\end{aligned}$$

which implies

$$\nu\eta(1-2\eta L)m\mathbb{E}_{\mathcal{I}^s,j^s|\tilde{\boldsymbol{x}}}\left[F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}})\right] \leq \left(\frac{1}{\alpha} + 2\nu\eta^2 Lm\right)[F(\tilde{\boldsymbol{x}}) - F(\hat{\boldsymbol{x}})] + \frac{Q}{2}.$$

Pick η such that

$$1 - 2\eta L > 0, \tag{3.32}$$

we obtain

$$\mathbb{E}_{\mathcal{I}^s,j^s|\tilde{\boldsymbol{x}}}\left[F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}})\right] \leq \left(\frac{1}{\nu\eta\alpha(1-2\eta L)m} + \frac{2\eta L}{1-2\eta L}\right)[F(\tilde{\boldsymbol{x}}) - F(\hat{\boldsymbol{x}})] + \frac{Q}{2\nu\eta\alpha(1-2\eta L)m}.$$

To guarantee the convergence, we must impose

$$\frac{2\eta L}{1-2\eta L} < 1. \tag{3.33}$$

Putting (3.32), (3.33) and $\nu(1-\eta\alpha) \leq 1$ together gives

$$\eta < \frac{1}{4L}, \quad \nu \leq \frac{1}{1-\eta\alpha}. \tag{3.34}$$

The convergence coefficient here is

$$\beta = \frac{1}{\nu\eta\alpha(1-2\eta L)m} + \frac{2\eta L}{1-2\eta L}. \tag{3.35}$$

Thus, we have

$$\mathbb{E}\left[F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}})\right] \leq \beta^s \left[F(\tilde{\boldsymbol{x}}^0) - F(\hat{\boldsymbol{x}})\right] + \frac{Q}{2\nu\eta\alpha(1-2\eta L)(1-\beta)m},$$

where the expectation is taken over $\{\mathcal{I}^1, j^1, \mathcal{I}^2, j^2, \cdots, \mathcal{I}^s, j^s\}$.

Case 2. $\nu(1-\eta\alpha) > 1$. In this case, (3.31) implies

$$\begin{aligned}\mathbb{E}_{\mathcal{I}^s|\tilde{\boldsymbol{x}}}\left[\left\|\boldsymbol{x}^m - \hat{\boldsymbol{x}}\right\|^2\right] \leq & \left(\frac{2}{\alpha} + 4\nu\eta^2 Lm\right)\left[F(\tilde{\boldsymbol{x}}) - F(\hat{\boldsymbol{x}})\right] + Q \\ & + \left(\frac{2}{\alpha}\left[\nu(1-\eta\alpha) - 1\right]m - 2\nu\eta(1-2\eta L)m\right)\mathbb{E}_{\mathcal{I}^s, j^s|\tilde{\boldsymbol{x}}}\left[F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}})\right].\end{aligned}$$

Rearranging the terms gives

$$\left(2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1\right)m\,\mathbb{E}_{\mathcal{I}^s, j^s|\tilde{\boldsymbol{x}}}\left[F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}})\right] \leq \left(1 + 2\nu\eta^2\alpha Lm\right)\left[F(\tilde{\boldsymbol{x}}) - F(\hat{\boldsymbol{x}})\right] + \frac{\alpha Q}{2}.$$

To ensure the convergence, the minimum requirements are

$$\begin{aligned}2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1 > 0, \\ 2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1 > 2\nu\eta^2\alpha L.\end{aligned}$$

That is,

$$4\nu\alpha L\eta^2 - 2\nu\alpha\eta + \nu - 1 < 0.$$

We need to guarantee the feasible set of the above inequality is non-empty for the positive variable η. Thus, we require

$$4\nu^2\alpha^2 - 4 \times 4\nu\alpha L(\nu - 1) > 0,$$

which is equivalent to

$$\nu < \frac{4L}{4L - \alpha}.$$

Combining it with $\nu(1-\eta\alpha) > 1$ gives

$$\frac{1}{1-\eta\alpha} < \nu < \frac{4L}{4L-\alpha}.$$

To ensure the above feasible set is non-empty, we impose

$$\frac{1}{1-\eta\alpha} < \frac{4L}{4L-\alpha},$$

so that

$$0 < \eta < \frac{1}{4L}, \quad \frac{1}{1-\eta\alpha} < \nu < \frac{4L}{4L-\alpha}. \tag{3.36}$$

The convergence coefficient for this case is

$$\beta = \frac{1}{(2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1)\, m} + \frac{2\nu\eta^2\alpha L}{2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1}. \tag{3.37}$$

Thus,

$$\mathbb{E}\left[F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}})\right] \leq \beta^s \left[F(\tilde{\boldsymbol{x}}^0) - F(\hat{\boldsymbol{x}})\right] + \frac{\alpha Q}{2(2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1)(1-\beta)m}.$$

By combining (3.34) and (3.36), the minimum requirement for η and ν is

$$0 < \eta < \frac{1}{4L}, \quad \nu < \frac{4L}{4L-\alpha}.$$

The proof is complete. □

3.D.2 Proof of Corollary 3.7

Proof. By noting the concavity of the square root function, we have

$$\begin{aligned}\mathbb{E}\Big[\sqrt{\max\{F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}}), 0\}}\Big] &\leq \sqrt{\mathbb{E}\big[\max\{F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}}), 0\}\big]} \\ &\leq \sqrt{(2/3)^s \max\{F(\tilde{\boldsymbol{x}}^0) - F(\hat{\boldsymbol{x}}), 0\} + \tau(\hat{\boldsymbol{x}})}.\end{aligned}$$

Suppose that $F(\boldsymbol{x})$ satisfies RSS with parameter $L' \in [\alpha, L]$. It follows that

$$F(\tilde{\boldsymbol{x}}^0) - F(\hat{\boldsymbol{x}}) \leq \left\langle \nabla F(\hat{\boldsymbol{x}}), \tilde{\boldsymbol{x}}^0 - \hat{\boldsymbol{x}} \right\rangle + \frac{L'}{2} \left\| \tilde{\boldsymbol{x}}^0 - \hat{\boldsymbol{x}} \right\|^2 \leq \frac{1}{2L'} \left\| \nabla_{k+K} F(\hat{\boldsymbol{x}}) \right\|^2 + L' \left\| \tilde{\boldsymbol{x}}^0 - \hat{\boldsymbol{x}} \right\|^2.$$

Recall that

$$\tau(\hat{\boldsymbol{x}}) = \frac{5\omega}{\alpha} \left\| \nabla_{3k+K} F(\hat{\boldsymbol{x}}) \right\| + \frac{1}{\alpha L} \left\| \nabla_{3k+K} F(\hat{\boldsymbol{x}}) \right\|^2.$$

Hence using $\sqrt{a+b+c+d} \leq \sqrt{a} + \sqrt{b} + \sqrt{c} + \sqrt{d}$ gives

$$\begin{aligned} \mathbb{E}\Big[\sqrt{\max\{F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}}), 0\}} \Big] \leq \sqrt{L'} \left(\frac{2}{3} \right)^{\frac{s}{2}} \left\| \tilde{\boldsymbol{x}}^0 - \hat{\boldsymbol{x}} \right\| + \sqrt{\frac{5\omega}{\alpha} \left\| \nabla_{3k+K} F(\hat{\boldsymbol{x}}) \right\|} \\ + \left(\frac{1}{\alpha} + \sqrt{\frac{1}{2\alpha}} \right) \left\| \nabla_{3k+K} F(\hat{\boldsymbol{x}}) \right\|. \end{aligned}$$

Finally, the RSC property immediately suggests that (see, e.g., Lemma 20 in [130])

$$\begin{aligned} \mathbb{E}\big[\left\| \tilde{\boldsymbol{x}}^s - \hat{\boldsymbol{x}} \right\| \big] &\leq \sqrt{\frac{2}{\alpha}} \mathbb{E}\Big[\sqrt{\max\{F(\tilde{\boldsymbol{x}}^s) - F(\hat{\boldsymbol{x}}), 0\}} \Big] + \frac{2 \left\| \nabla_{k+K} F(\hat{\boldsymbol{x}}) \right\|}{\alpha} \\ &\leq \sqrt{\frac{2L'}{\alpha}} \cdot \left(\frac{2}{3} \right)^{\frac{s}{2}} \left\| \tilde{\boldsymbol{x}}^0 - \hat{\boldsymbol{x}} \right\| + \sqrt{\frac{10\omega}{\alpha^2} \left\| \nabla_{3k+K} F(\hat{\boldsymbol{x}}) \right\|} \\ &\quad + \left(\sqrt{\frac{2}{\alpha^3}} + \frac{3}{\alpha} \right) \left\| \nabla_{3k+K} F(\hat{\boldsymbol{x}}) \right\|. \end{aligned}$$

The proof is complete. □

Chapter 4

Online Optimization for Low-Rank Matrix Recovery

4.1 Background

In the last decade, estimating low-rank matrices has attracted increasing attention in the machine learning community owing to its successful applications in a wide range of fields including subspace clustering [93], collaborative filtering [63] and robust dimensionality reduction [34]. Suppose that we are given an observed data matrix $\boldsymbol{Z}$ in $\mathbb{R}^{d\times n}$, i.e., n observations in d ambient dimensions, we aim to learn a prediction matrix $\boldsymbol{X}$ with a low-rank structure so as to approximate the observation. This problem, together with its many variants, typically involves minimizing a weighted combination of the residual error and a penalty for the matrix rank.

Generally speaking, it is intractable to optimize a matrix rank due to the discrete and non-convex nature [120]. To tackle this challenge, researchers suggested alternative convex relaxations to the matrix rank. The two most widely used convex surrogates are the nuclear norm [120] and the max-norm (a.k.a. γ_2-norm) [138]. The nuclear norm is defined as the sum of the matrix singular values. Like the ℓ_1 norm in the vector case that induces sparsity, the nuclear norm was proposed as a rank minimization heuristic and was able to be formulated as a semi-definite programming (SDP) problem [58]. By combining the SDP formulation and the matrix factorization technique, [138] showed that the collaborative filtering problem can be effectively solved by optimizing a soft-margin based

program. Another interesting work on the nuclear norm comes from the data compression community. In real-world applications, due to possible sensor failure and background clutter, the underlying data can easily be corrupted. In this case, estimates produced by Principal Component Analysis (PCA) may be deviated far from the true subspace [79]. To handle the (gross) corruption, in the seminal work of [34], Candès et al. proposed a new formulation termed Robust PCA (RPCA), and proved that under mild conditions, solving a convex optimization problem consisting of a nuclear norm regularization and a weighted ℓ_1 norm penalty can exactly recover the low-rank component of the underlying data even if a constant fraction of the entries are arbitrarily corrupted.

The max-norm variant was developed as another convex relaxation to the rank function [138], where Srebro et al. formulated the max-norm regularized problem as an SDP and empirically showed the superiority to the nuclear norm. The main theoretical study on the max-norm comes from [139], where Srebro and Shraibman considered collaborative filtering as an example and proved that the max-norm scheme enjoys a lower generalization error than the nuclear norm. Following these theoretical foundations, [76] improved the error bound for the clustering problem. Another important contribution from [76] is that they partially characterized the subgradient of the max-norm, which is a hard mathematical entity and cannot be fully understood to date. However, since SDP solver is not scalable, there is a large gap between the theoretical progress and the practical applicability of the max-norm. To bridge the gap, a number of follow-up work attempted to design efficient algorithms to solve max-norm regularized or constrained problems. For example, [121] devised a gradient-based optimization method and empirically showed promising results on large collaborative filtering data sets. [87] presented large-scale optimization methods for max-norm constrained and max-norm regularized problems and showed a convergence to stationary point.

Nevertheless, algorithms presented in prior work [138, 121, 87, 112] require to access all the data when the objective function involves a max-norm regularization. In the large-scale setting, the applicability of such batch optimization methods will be hindered by the memory bottleneck. In this chapter, henceforth, we propose an online algorithm to solve max-norm regularized problems. The main advantage of online algorithms is that the memory cost is independent of the sample size, which makes it a good fit for the big data era.

To be more detailed, we are interested in a general max-norm regularized matrix decomposition

(MRMD) problem. Suppose that the observed data matrix $\boldsymbol{Z}$ can be decomposed into a low-rank component $\boldsymbol{X}$ and some structured noise $\boldsymbol{E}$, we aim to simultaneously and accurately estimate the two components, by solving the following convex program:

$$\text{(MRMD)} \quad \min_{\boldsymbol{X},\boldsymbol{E}} \quad \frac{1}{2}\left\|\boldsymbol{Z}-\boldsymbol{X}-\boldsymbol{E}\right\|_F^2 + \frac{\lambda_1}{2}\left\|\boldsymbol{X}\right\|_{\max}^2 + \lambda_2 h(\boldsymbol{E}). \tag{4.1}$$

Here, $\|\cdot\|_F$ denotes the Frobenius norm which is a commonly used metric for evaluating the residual, $\|\cdot\|_{\max}$ is the max-norm (which promotes low-rankness), and λ_1 and λ_2 are two non-negative parameters. $h(\boldsymbol{E})$ is some (convex) regularizer that can be adapted to various kinds of noise. We require that it can be represented as a summation of column norms. Formally, there exists some regularizer $\tilde{h}(\cdot)$, such that

$$h(\boldsymbol{E}) = \sum_{i=1}^{n} \tilde{h}(\boldsymbol{e}_i), \tag{4.2}$$

where $\boldsymbol{e}_i$ is the ith column of $\boldsymbol{E}$. Classical examples include:

- $\|\boldsymbol{E}\|_1$. That is, the ℓ_1 norm of the matrix $\boldsymbol{E}$ seen as a long vector, which is used to handle sparse corruption. In this case, $\tilde{h}(\cdot)$ is the ℓ_1 vector norm. Note that when equipped with this norm, the above problem reduces to the well-known RPCA formulation [34], but with the nuclear norm being replaced by the max-norm.

- $\|\boldsymbol{E}\|_{2,1}$. This is defined as the summation of the ℓ_2 column norms, which is effective when a small fraction of the samples are contaminated (recall that each column of $\boldsymbol{Z}$ is a sample). The matrix $\ell_{2,1}$ norm is typically used to handle outliers and interestingly, the above program becomes Outlier PCA [154] in this case.

- $\|\boldsymbol{E}\|_F^2$ or $\boldsymbol{E} = \boldsymbol{0}$. The formulation of (4.1) works as a large-margin based program, with the hinge loss replaced by the squared loss [138].

Hence, (4.1) is quite general and our algorithmic and theoretical results hold for such a general form, uncovering important problems including max-norm regularized RPCA, max-norm regularized Outlier PCA and maximum margin matrix factorization. Furthermore, with a careful design, the above formulation (4.1) can be extended to address the matrix completion problem [36], as we will show in Section 4.5.

Considering the connection between max-norm and nuclear norm, one might be interested in an alternative formulation as follows:

$$\min_{\boldsymbol{X},\boldsymbol{E}} \quad \frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{X}-\boldsymbol{E}\|_F^2 + \frac{\lambda_1'}{2}\|\boldsymbol{X}\|_{\max} + \lambda_2 h(\boldsymbol{E}). \tag{4.3}$$

First, we would like to point out that the above formulation is equivalent to (4.1), in the sense that if we choose proper parameter λ_1' for (4.3) and some parameter λ_1 for (4.1), they produce same solutions. To see this, we note that (4.3) is equivalent to the following constrained program:

$$\min_{\boldsymbol{X},\boldsymbol{E}} \frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{X}-\boldsymbol{E}\|_F^2 + \lambda_2 h(\boldsymbol{E}), \quad \text{s.t. } \|\boldsymbol{X}\|_{\max} \leq \kappa,$$

for some parameter κ. Taking the square on both sides of the inequality constraint gives

$$\min_{\boldsymbol{X},\boldsymbol{E}} \frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{X}-\boldsymbol{E}\|_F^2 + \lambda_2 h(\boldsymbol{E}), \quad \text{s.t. } \|\boldsymbol{X}\|_{\max}^2 \leq \kappa^2.$$

Again, we know that for some proper choice of λ_1, the above program is equivalent to (4.1). The reason we choose (4.1) is for a convenient computation of the solution. We defer a more detailed discussion to Section 4.3.

4.1.1 Contributions

In summary, our main contributions is two-folds: **1)** We are the first to develop an online algorithm to solve a family of max-norm regularized problems (4.1), which admits a wide range of applications in machine learning. We also show that our approach can be used to solve other popular max-norm regularized problems such as matrix completion. **2)** We prove that the sequence of solutions produced by our algorithm converges to a stationary point of the expected loss function asymptotically (see Section 4.4).

4.1.2 Related Work

Here we discuss some relevant work in the literature. Most previous work on max-norm focused on showing that it is empirically superior to the nuclear norm in real-world problems, such as collaborative filtering [138], clustering [76] and hamming embedding [108]. Other work, for instance, [126],

studied the influence of data distribution with the max-norm regularization and observed good performance even when the data are sampled non-uniformly. There are also interesting work which investigated the connection between the max-norm and the nuclear norm. A comprehensive study on this problem, in the context of collaborative filtering, can be found in [139], which established and compared the generalization bound for the nuclear norm regularization and the max-norm, showing that the latter one results in a tighter bound. More recently, [63] attempted to unify them to gain insightful perspective.

Also in line with this work is matrix decomposition. As we mentioned, when we penalize the noise $\boldsymbol{E}$ with ℓ_1 matrix norm, it reverts to the well known RPCA formulation [34]. The only difference is that [34] analyzed the RPCA problem with the nuclear norm, while (4.1) employs the max-norm. Owing to the explicit form of the subgradient of the nuclear norm, [34] established a dual certificate for the success of their formulation, which facilitates their theoretical analysis. In contrast, the max-norm is a much harder mathematical entity (even its subgradient has not been fully characterized). Henceforth, it still remains challenging to understand the behavior of the max-norm regularizer in the general setting (4.1). Studying the conditions for the exact recovery of MRMD is out of the scope of this chapter. We leave this as a future work.

From a high level, the goal of this chapter is similar to that of [59]. Motivated by the celebrated RPCA problem [34, 154, 155], [59] developed an online implementation for the nuclear-norm regularized matrix decomposition. Yet, since the max-norm is a more complicated mathematical entity, new techniques and insights are needed in order to develop online methods for the max-norm regularization. For example, after converting the max-norm to its matrix factorization form, the data are still coupled and we propose to transform the problem to a constrained one for stochastic optimization.

The main technical contribution of this chapter is converting max-norm regularization to an appropriate matrix factorization problem that is amenable to online implementation. Compared to [99] which also studies online matrix factorization, our formulation contains an additional structured noise that brings the benefit of robustness to contamination. Some of our proof techniques are also different. For example, to prove the convergence of the dictionary and to well define their problem, [99] assumed that the magnitude of the learned dictionary is constrained. In contrast, we *prove* that the optimal basis is uniformly bounded, and hence our problem is naturally well-defined.

Our algorithm can be viewed as a majorization-minimization scheme, for which [98] derived a general analysis on the convergence behavior. However, we find that Algorithm 1 in [98] requires the knowledge of the Lipschitz constant to obtain a surrogate function. In our work, we use a suboptimal solution to derive the surrogate function (see Step 3 and Step 5 in our Algorithm 2 to be introduced). Due to such a different mechanism, it remains an open question whether one can apply their algorithm and theoretical analysis to the problem considered here. It is also worth mentioning that in order to establish their theoretical results, [98] *assumed* that the iterates and the empirical loss function are uniformly bounded (see Assumption (C) and Assumption (D) therein). For our problem, we can virtually prove this property (see Proposition 4.4 and Corollary 4.5 to follow). Finally, we note that our algorithm is different from block coordinate descent, see, e.g., [147]. In fact, block coordinate descent randomly and independently picks a mini-batch of samples and updates a block variable, whereas we in each iteration update only the variables associated with the revealed sample. Another key difference is that [147] considered a strongly convex objective function, while we are working with a non-convex case.

4.1.3 Notation

There are four matrix norms that will be heavily used: $\|\boldsymbol{M}\|_F$ for the Frobenius norm, $\|\boldsymbol{M}\|_1$ for the ℓ_1 matrix norm seen as a long vector, $\|\boldsymbol{M}\|_{\max}$ for the max-norm induced by the product of $\ell_{2,\infty}$-norm on the factors of $\boldsymbol{M}$. Here, the $\ell_{2,\infty}$-norm is defined as the maximum ℓ_2 row norm.

4.2 Problem Setup

We are interested in developing an online algorithm for the MRMD problem (4.1) so as to mitigate the memory issue. To this end, we utilize the following definition of the max-norm [138]:

$$\|\boldsymbol{X}\|_{\max} := \min_{\boldsymbol{U},\boldsymbol{V}} \Big\{ \|\boldsymbol{U}\|_{2,\infty} \cdot \|\boldsymbol{V}\|_{2,\infty} : \ \boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top, \boldsymbol{U} \in \mathbb{R}^{d\times r}, \boldsymbol{V} \in \mathbb{R}^{n\times r} \Big\}, \tag{4.4}$$

where r is an upper bound on the intrinsic dimension of the underlying data. Plugging the above back to (4.1), we obtain an equivalent form:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} \quad \frac{1}{2}\left\|\boldsymbol{Z}-\boldsymbol{U}\boldsymbol{V}^\top-\boldsymbol{E}\right\|_F^2+\frac{\lambda_1}{2}\|\boldsymbol{U}\|_{2,\infty}^2\|\boldsymbol{V}\|_{2,\infty}^2+\lambda_2 h(\boldsymbol{E}). \tag{4.5}$$

In this chapter, if not specified, "equivalent" means we do not change the optimal value of the objective function. Intuitively, the variable $\boldsymbol{U}$ serves as a (possibly overcomplete) basis for the clean data while correspondingly, the variable $\boldsymbol{V}$ works as a coefficients matrix with each row being the coefficients for each sample (recall that we organize the observed samples in a column-wise manner). In order to make the new formulation (4.5) equivalent to MRMD (4.1), the quantity of d should be sufficiently large due to (4.4).

At a first sight, the problem can only be optimized in a batch manner for which the memory cost is prohibitive. To see this, note that we are considering the regime of $r < d \ll n$ and the size of the coefficients $\boldsymbol{V}$ is proportional to n. In order to optimize the above program over the variable $\boldsymbol{V}$, we have to compute the gradient with respect to it. Recall that the $\ell_{2,\infty}$-norm counts the largest ℓ_2 row norm of $\boldsymbol{V}$, hence coupling all the samples (each row of $\boldsymbol{V}$ associates with a sample).

Fortunately, we have the following proposition that alleviates the inter-dependency among the rows of $\boldsymbol{V}$, hence facilitating an online algorithm where the rows of $\boldsymbol{V}$ can be optimized sequentially.

Proposition 4.1. *Problem* (4.5) *is equivalent to the following constrained program:*

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} \ \frac{1}{2}\left\|\boldsymbol{Z}-\boldsymbol{U}\boldsymbol{V}^\top-\boldsymbol{E}\right\|_F+\frac{\lambda_1}{2}\|\boldsymbol{U}\|_{2,\infty}^2+\lambda_2 h(\boldsymbol{E}), \quad \text{s.t. } \|\boldsymbol{V}\|_{2,\infty}^2\le 1. \tag{4.6}$$

Moreover, there exists an optimal solution $(\boldsymbol{U}^*,\boldsymbol{V}^*,\boldsymbol{E}^*)$ *attained at the boundary of the feasible set, i.e.,* $\|\boldsymbol{V}^*\|_{2,\infty}^2$ *is equal to the unit.*

Proposition 4.1 is crucial for the online implementation. It states that our primal MRMD problem (4.1) can be transformed to an equivalent constrained program (4.6) where the coefficients of *each individual* sample (i.e., a row of the matrix $\boldsymbol{V}$) is *uniformly and separately* constrained.

Consequently, we can, equipped with Proposition 4.1, rewrite the original problem in an online

fashion, with each sample being separately processed:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} \frac{1}{2}\sum_{i=1}^{n} \|\boldsymbol{z}_i - \boldsymbol{U}\boldsymbol{v}_i - \boldsymbol{e}_i\|^2 + \frac{\lambda_1}{2}\|\boldsymbol{U}\|_{2,\infty}^2 + \lambda_2 \sum_{i=1}^{n} \tilde{h}(\boldsymbol{e}_i),\ \text{s.t.}\ \|\boldsymbol{v}_i\|^2 \leq 1,\ \forall\, i \in [n], \quad (4.7)$$

where $\boldsymbol{z}_i$ is the ith observation, $\boldsymbol{v}_i$ is the ith row of $\boldsymbol{V}$ and $\boldsymbol{e}_i$ is some structured error penalized by the (convex) regularizer $\tilde{h}(\cdot)$ (recall that we require $h(\boldsymbol{E})$ can be decomposed column-wisely). Merging the first and third term above gives a compact form:

$$\min_{\boldsymbol{U}} \min_{\boldsymbol{V},\boldsymbol{E}} \sum_{i=1}^{n} \tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{U}, \boldsymbol{v}_i, \boldsymbol{e}_i) + \frac{\lambda_1}{2}\|\boldsymbol{U}\|_{2,\infty}^2, \quad \text{s.t.}\ \|\boldsymbol{v}_i\|^2 \leq 1,\ \forall i \in [n], \quad (4.8)$$

where

$$\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e}) := \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{U}\boldsymbol{v} - \boldsymbol{e}\|^2 + \lambda_2 \tilde{h}(\boldsymbol{e}). \quad (4.9)$$

This is indeed equivalent to optimizing (i.e., minimizing) the empirical loss function:

$$\min_{\boldsymbol{U}}\ f_n(\boldsymbol{U}), \quad (4.10)$$

where

$$f_n(\boldsymbol{U}) := \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{z}_i, \boldsymbol{U}) + \frac{\lambda_1}{2n}\|\boldsymbol{U}\|_{2,\infty}^2, \quad (4.11)$$

and

$$\ell(\boldsymbol{z}, \boldsymbol{U}) = \min_{\boldsymbol{v},\boldsymbol{e},\|\boldsymbol{v}\|^2 \leq 1} \tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e}). \quad (4.12)$$

Note that by Proposition 4.1, as long as the quantity of d is sufficiently large, the program (4.10) is equivalent to the primal formulation (4.1), in the sense that both of them could attain the same minimum. Compared to MRMD (4.1), which is solved in a batch manner by prior work, the formulation (4.10) paves a way for stochastic optimization procedure since all the samples are decoupled.

4.3 Algorithm

Based on the derivation in the preceding section, we are now ready to present our online algorithm to solve the MRMD problem (4.1). The implementation is outlined in Algorithm 2. Here we briefly explain the underlying intuition. We optimize the coefficients $\boldsymbol{v}$, the structured noise $\boldsymbol{e}$ and the basis

Algorithm 2 Online Max-Norm Regularized Matrix Decomposition

Require: $\boldsymbol{Z} \in \mathbb{R}^{d\times n}$ (observed samples), parameters $\lambda_1 > 0$ and $\lambda_2 > 0$, $\boldsymbol{U}_0 \in \mathbb{R}^{d\times r}$ (initial basis), zero matrices $\boldsymbol{A}_0 \in \mathbb{R}^{r\times r}$ and $\boldsymbol{B}_0 \in \mathbb{R}^{d\times r}$.

Ensure: Optimal basis $\boldsymbol{U}_n$.

1: **for** $t = 1$ to n **do**

2: Access the t-th sample $\boldsymbol{z}_t$.

3: Compute the coefficient and noise:

$$\{\boldsymbol{v}_t, \boldsymbol{e}_t\} = \underset{\boldsymbol{v},\boldsymbol{e},\|\boldsymbol{v}\|^2\leq 1}{\arg\min}\ \tilde{\ell}(\boldsymbol{z}_t, \boldsymbol{U}_{t-1}, \boldsymbol{v}, \boldsymbol{e}).$$

4: Compute the accumulation matrices $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$:

$$\begin{aligned}\boldsymbol{A}_t &\longleftarrow \boldsymbol{A}_{t-1} + \boldsymbol{v}_t\boldsymbol{v}_t^\top,\\ \boldsymbol{B}_t &\longleftarrow \boldsymbol{B}_{t-1} + (\boldsymbol{z}_t - \boldsymbol{e}_t)\,\boldsymbol{v}_t^\top.\end{aligned}$$

5: Compute the basis $\boldsymbol{U}_t$ by optimizing the surrogate function (4.13):

$$\begin{aligned}\boldsymbol{U}_t &= \underset{\boldsymbol{U}}{\arg\min}\,\frac{1}{t}\sum_{i=1}^{t}\tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{U}, \boldsymbol{v}_i, \boldsymbol{e}_i) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2\\ &= \underset{\boldsymbol{U}}{\arg\min}\,\frac{1}{t}\left(\frac{1}{2}\operatorname{Tr}\left(\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{A}_t\right) - \operatorname{Tr}\left(\boldsymbol{U}^\top\boldsymbol{B}_t\right)\right) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2.\end{aligned}$$

6: **end for**

$\boldsymbol{U}$ in an alternating manner, with only the basis $\boldsymbol{U}$ and two accumulation matrices being kept in memory. At the t-th iteration, given the basis $\boldsymbol{U}_{t-1}$ produced by the previous iteration, we can optimize (4.12) by examining the Karush-Kuhn-Tucker (KKT) conditions. To obtain a new iterate $\boldsymbol{U}_t$, we then minimize the following objective function:

$$g_t(\boldsymbol{U}) := \frac{1}{t}\sum_{i=1}^{t}\tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{U}, \boldsymbol{v}_i, \boldsymbol{e}_i) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2, \tag{4.13}$$

where $\{\boldsymbol{v}_i\}_{i=1}^t$ and $\{\boldsymbol{e}_i\}_{i=1}^t$ are already on hand. It can be verified that (4.13) is a surrogate function of the empirical loss $f_t(\boldsymbol{U})$ (4.11), since the obtained $\boldsymbol{v}_i$'s and $\boldsymbol{e}_i$'s are suboptimal. Interestingly, instead of recording all the past $\boldsymbol{v}_i$'s and $\boldsymbol{e}_i$'s, we only need to store two accumulation matrices whose sizes are independent of n, as shown in Algorithm 2. In the sequel, we elaborate each step.

4.3.1 Update the Coefficients and Noise

Given a sample $\boldsymbol{z}$ and a basis $\boldsymbol{U}$, we are able to estimate the optimal coefficients $\boldsymbol{v}$ and the noise $\boldsymbol{e}$ by minimizing $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e})$. That is, we are to solve the following program:

$$\min_{\boldsymbol{v},\boldsymbol{e}} \frac{1}{2} \|\boldsymbol{z} - \boldsymbol{U}\boldsymbol{v} - \boldsymbol{e}\|^2 + \lambda_2 \tilde{h}(\boldsymbol{e}), \quad \text{s.t. } \|\boldsymbol{v}\| \leq 1. \tag{4.14}$$

We notice that the constraint only involves the variable $\boldsymbol{v}$, and in order to optimize $\boldsymbol{v}$, we only need to consider the residual term in the objective function. This motivates us to employ a block coordinate descent algorithm. Namely, we alternatively optimize one variable with the other fixed, until some stopping criteria is fulfilled. In our implementation, when the difference between the current and the previous iterate is smaller than 10^{-6}, or the number of iterations exceeds 100, our algorithm will terminate and return the optimum.

Optimize the Coefficients $\boldsymbol{v}$

Now it remains to show how to compute a new iterate for one variable when the other one is fixed. According to [14], when the objective function is strongly convex with respect to (w.r.t.) each block variable, we are guaranteed that the block coordinate minimization algorithm converges. In our case, we observe that such a condition holds for $\boldsymbol{e}$ but not necessary for $\boldsymbol{v}$. In fact, the strong convexity w.r.t. $\boldsymbol{v}$ holds if and only if the basis $\boldsymbol{U}$ is with full rank. When $\boldsymbol{U}$ is not full rank, we may compute the Moore Penrose pseudo inverse to solve $\boldsymbol{v}$. However, for computational efficiency, we append a small jitter $\frac{\epsilon}{2} \|\boldsymbol{v}\|^2$ to the objective if necessary, so as to guarantee the convergence ($\epsilon = 0.01$ in our experiments). In this way, we obtain a *potentially* admissible iterate for $\boldsymbol{v}$ as follows:

$$\boldsymbol{v}_0 = (\boldsymbol{U}^\top \boldsymbol{U} + \epsilon \boldsymbol{I}_d)^{-1} \boldsymbol{U}^\top (\boldsymbol{z} - \boldsymbol{e}). \tag{4.15}$$

Here, ϵ is set to be zero if and only if $\boldsymbol{U}$ is full rank.

Next, we examine if $\boldsymbol{v}_0$ violates the inequality constraint in (4.14). If it happens to be a feasible solution, i.e., $\|\boldsymbol{v}_0\| \leq 1$, we have found the new iterate for $\boldsymbol{v}$. Otherwise, we conclude that the optimum of $\boldsymbol{v}$ must be attained on the boundary of the feasible set, i.e., $\|\boldsymbol{v}\| = 1$, for which the

minimizer can be computed by the method of Lagrangian multipliers:

$$\max_{\eta} \min_{\boldsymbol{v}} \frac{1}{2} \|\boldsymbol{z} - \boldsymbol{U}\boldsymbol{v} - \boldsymbol{e}\|^2 + \frac{\eta}{2} \left(\|\boldsymbol{v}\|^2 - 1 \right), \quad \text{s.t. } \eta > 0, \quad \|\boldsymbol{v}\| = 1. \tag{4.16}$$

By differentiating the objective function with respect to $\boldsymbol{v}$, we have

$$\boldsymbol{v} = \left(\boldsymbol{U}^\top \boldsymbol{U} + \eta \boldsymbol{I}_d \right)^{-1} \boldsymbol{U}^\top (\boldsymbol{z} - \boldsymbol{e}). \tag{4.17}$$

The following argument helps us to efficiently search the optimal solution.

Proposition 4.2. *Let $\boldsymbol{v}$ be given by* (4.17)*, where $\boldsymbol{U}$, $\boldsymbol{z}$ and $\boldsymbol{e}$ are assumed to be fixed. Then, the ℓ_2 norm of $\boldsymbol{v}$ is strictly monotonically decreasing with respect to the quantity of η.*

Proof. For simplicity, let us denote

$$\boldsymbol{v}(\eta) = \left(\boldsymbol{U}^\top \boldsymbol{U} + \eta \boldsymbol{I}_d \right)^{-1} \boldsymbol{b},$$

where $\boldsymbol{b} = \boldsymbol{U}^\top (\boldsymbol{z} - \boldsymbol{e})$ is a fixed vector. Suppose we have a full singular value decomposition (SVD) on $\boldsymbol{U} = \boldsymbol{L}\boldsymbol{S}\boldsymbol{R}^\top$, where the singular values $\{s_{11}, s_{22}, \cdots, s_{dd}\}$ (i.e., the diagonal elements in $\boldsymbol{S}$) are arranged in a decreasing order and at most r number of them are non-zero. Substituting $\boldsymbol{U}$ with its SVD, we obtain the squared ℓ_2 norm for $\boldsymbol{v}(\eta)$:

$$\|\boldsymbol{v}(\eta)\|^2 = \boldsymbol{b}^\top \left(\boldsymbol{R}\boldsymbol{S}^2\boldsymbol{R}^\top + \eta \boldsymbol{I}_d \right)^{-2} \boldsymbol{b} = \boldsymbol{b}^\top \boldsymbol{R}\boldsymbol{S}_\eta \boldsymbol{R}^\top \boldsymbol{b},$$

where $\boldsymbol{S}_\eta$ is a diagonal matrix whose ith diagonal element equals $(s_{ii}^2 + \eta)^{-2}$.

For any two entities $\eta_1 > \eta_2$, it is easy to see that the matrix $\boldsymbol{S}_{\eta_1} - \boldsymbol{S}_{\eta_2}$ is negative definite. Hence, it always holds that

$$\|\boldsymbol{v}(\eta_1)\|^2 - \|\boldsymbol{v}(\eta_2)\|^2 = \boldsymbol{b}^\top \boldsymbol{R}(\boldsymbol{S}_{\eta_1} - \boldsymbol{S}_{\eta_2})\boldsymbol{R}^\top \boldsymbol{b} < 0,$$

which concludes the proof. □

The above proposition offers an efficient computation scheme, i.e., bisection method, for searching the optimal $\boldsymbol{v}$ as well as the dual variable η. To be more detailed, we can maintain a lower bound

η_1 and an upper bound η_2, such that $\|\boldsymbol{v}(\eta_1)\| \geq 1$ and $\|\boldsymbol{v}(\eta_2)\| \leq 1$. According to the monotonic property shown in Proposition 4.2, the optimal η must fall into the interval $[\eta_1, \eta_2]$. By evaluating the value of $\|\boldsymbol{v}\|$ at the middle point $(\eta_1 + \eta_2)/2$, we can sequentially shrink the interval until $\|\boldsymbol{v}\|$ is close or equal to one. Note that we can initialize η_1 with zero (since $\|\boldsymbol{v}_0\| > 1$ implies the optimal $\eta^* > \epsilon \geq 0$). The bisection routine is summarized in Algorithm 3.

Algorithm 3 Bisection Method for Problem (4.16)

Require: $\boldsymbol{U} \in \mathbb{R}^{d\times r}$, $\boldsymbol{z} \in \mathbb{R}^d$, $\boldsymbol{e} \in \mathbb{R}^d$.
Ensure: Optimal primal and dual pair $(\boldsymbol{v}, \eta)$.

1: Initialize the lower bound $\eta_1 = 0$ and the upper bound η_2 large enough such that $\|\boldsymbol{v}(\eta_2)\| \leq 1$.
2: **repeat**
3: Compute the middle point:

$$\eta \leftarrow \frac{1}{2}(\eta_1 + \eta_2).$$

4: **if** $\|\boldsymbol{v}(\eta)\| < 1$ **then**
5: Update η_2:

$$\eta_2 \leftarrow \eta.$$

6: **else**
7: Update η_1:

$$\eta_1 \leftarrow \eta.$$

8: **end if**
9: **until** $\|\boldsymbol{v}\| = 1$

Optimize the Noise $\boldsymbol{e}$

We have clarified the technique used for solving $\boldsymbol{v}$ in Problem (4.14) when $\boldsymbol{e}$ is fixed. Now let us turn to the phase where $\boldsymbol{v}$ is fixed and we want to find the optimal $\boldsymbol{e}$. Since $\boldsymbol{e}$ is an unconstrained variable, generally speaking, it is much easier to solve, although one may employ different strategies for various regularizers $\tilde{h}(\cdot)$. Here, we discuss the solutions for popular choices of the regularizer.

1. $\tilde{h}(\boldsymbol{e}) = \|\boldsymbol{e}\|_1$. The ℓ_1 regularizer results in a closed form solution for $\boldsymbol{e}$ as follows:

$$\boldsymbol{e} = \mathcal{S}_{\lambda_2}[\boldsymbol{z} - \boldsymbol{U}\boldsymbol{v}], \tag{4.18}$$

where $\mathcal{S}_{\lambda_2}[\cdot]$ is the soft-thresholding operator [49].

Algorithm 4 The Coefficients and Noise Update (Problem (4.14))

Require: $\boldsymbol{U} \in \mathbb{R}^{d\times r}$, $\boldsymbol{z} \in \mathbb{R}^d$, parameter λ_2 and a small jitter ϵ.
Ensure: Optimal $\boldsymbol{v}$ and $\boldsymbol{e}$.
1: Initialize $\boldsymbol{e} = \boldsymbol{0}$.
2: **repeat**
3: Compute the potential solution $\boldsymbol{v}_0$ given in (4.15).
4: **if** $\|\boldsymbol{v}_0\| \leq 1$ **then**
5: Update $\boldsymbol{v}$ with

$$\boldsymbol{v} = \boldsymbol{v}_0,$$

6: **else**
7: Update $\boldsymbol{v}$ by Algorithm 3.
8: **end if**
9: Update the noise $\boldsymbol{e}$.
10: **until** convergence

2. $\tilde{h}(\boldsymbol{e}) = \|\boldsymbol{e}\|$. The solution in this case can be characterized as follows (see, for example, [93]):

$$\boldsymbol{e} = \begin{cases} \frac{\|\boldsymbol{z}-\boldsymbol{U}\boldsymbol{v}\|}{\|\boldsymbol{z}-\boldsymbol{U}\boldsymbol{v}\|-\lambda_2}(\boldsymbol{z}-\boldsymbol{U}\boldsymbol{v}), & \text{if } \lambda_2 < \|\boldsymbol{z}-\boldsymbol{U}\boldsymbol{v}\|, \\ \boldsymbol{0}, & \text{otherwise.} \end{cases} \tag{4.19}$$

Finally, for completeness, we summarize the routine for updating the coefficients and the noise in Algorithm 4. The readers may refer to the preceding paragraphs for details.

4.3.2 Update the Basis

With all the past filtration $\mathcal{F}_t = \{\boldsymbol{z}_i, \boldsymbol{v}_i, \boldsymbol{e}_i\}_{i=1}^t$ on hand, we are able to compute a new basis $\boldsymbol{U}_t$ by minimizing the surrogate function (4.13). That is, we are to solve the following program:

$$\min_{\boldsymbol{U}} \quad \frac{1}{t}\sum_{i=1}^{t} \tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{U}, \boldsymbol{v}_i, \boldsymbol{e}_i) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2 . \tag{4.20}$$

By a simple expansion, for any $i \in [t]$, we have

$$\tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{U}, \boldsymbol{v}_i, \boldsymbol{e}_i) = \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{v}_i\boldsymbol{v}_i^\top\right) - \operatorname{Tr}\left(\boldsymbol{U}^\top(\boldsymbol{z}_i - \boldsymbol{e}_i)\boldsymbol{v}_i^\top\right) + \frac{1}{2}\|\boldsymbol{z}_i - \boldsymbol{e}_i\|^2 + \lambda_2\tilde{h}(\boldsymbol{e}_i). \tag{4.21}$$

Substituting back into (4.20), putting $\boldsymbol{A}_t = \sum_{i=1}^{t} \boldsymbol{v}_i \boldsymbol{v}_i^\top$, $\boldsymbol{B}_t = \sum_{i=1}^{t} (\boldsymbol{z}_i - \boldsymbol{e}_i)\boldsymbol{v}_i^\top$ and removing constant terms, we obtain

$$\boldsymbol{U}_t = \arg\min_{\boldsymbol{U}} \frac{1}{t} \left(\frac{1}{2} \operatorname{Tr}\left(\boldsymbol{U}^\top \boldsymbol{U} \boldsymbol{A}_t \right) - \operatorname{Tr}\left(\boldsymbol{U}^\top \boldsymbol{B}_t \right) \right) + \frac{\lambda_1}{2t} \|\boldsymbol{U}\|_{2,\infty}^2 . \tag{4.22}$$

In order to derive the optimal solution, firstly, we need to characterize the subgradient of the squared $\ell_{2,\infty}$-norm. In fact, let $\boldsymbol{Q}$ be a positive semi-definite diagonal matrix, such that $\operatorname{Tr}(\boldsymbol{Q}) = 1$. Denote the set of row index which attains the maximum ℓ_2 row norm of $\boldsymbol{U}$ by $\mathcal{I}$. In this way, the subgradient of $\frac{1}{2}\|\boldsymbol{U}\|_{2,\infty}^2$ is given by

$$\partial \left(\frac{1}{2} \|\boldsymbol{U}\|_{2,\infty}^2 \right) = \boldsymbol{Q}\boldsymbol{U},\ q_{ii} \neq 0 \text{ if and only if } i \in \mathcal{I},\ q_{ij} = 0 \text{ for } i \neq j. \tag{4.23}$$

Equipped with the subgradient, we may apply block coordinate descent to update each column of $\boldsymbol{U}$ sequentially. We assume that the objective function (4.22) is strongly convex w.r.t. $\boldsymbol{U}$, implying that the block coordinate descent scheme can always converge to the global optimum [14].

We summarize the update procedure in Algorithm 5. In practice, we find that after revealing a large number of samples, performing one-pass update for each column of $\boldsymbol{U}$ is sufficient to guarantee a desirable accuracy, which matches the observation in [99].

Algorithm 5 The Basis Update

Require: $\boldsymbol{U} \in \mathbb{R}^{d \times r}$ in the previous iteration, accumulation matrix $\boldsymbol{A}$ and $\boldsymbol{B}$, parameter $\lambda_1 > 0$.
Ensure: Optimal basis $\boldsymbol{U}$ (updated).

1: **repeat**
2: Compute the subgradient of $\frac{1}{2}\|\boldsymbol{U}\|_{2,\infty}^2$:

$$\boldsymbol{G} = \partial \left(\frac{1}{2} \|\boldsymbol{U}\|_{2,\infty}^2 \right).$$

3: **for** $j = 1$ to d **do**
4: Update the jth column of $\boldsymbol{U}$:

$$\boldsymbol{u}_j \leftarrow \boldsymbol{u}_j - \frac{1}{a_{jj}} \left(\boldsymbol{U}\boldsymbol{a}_j - \boldsymbol{b}_j + \lambda_1 \boldsymbol{g}_j \right).$$

5: **end for**
6: **until** convergence

As we discussed in the introduction, one may prefer the formulation (4.3) to (4.1), although in

some sense they are equivalent. It is worth mentioning that our algorithm can easily be tailored to solve (4.3) by modifying Step 5 of Algorithm 2 as follows:

$$\boldsymbol{U}_t = \arg\min_{\boldsymbol{U}} \ \frac{1}{t}\left(\frac{1}{2}\operatorname{Tr}\left(\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{A}_t\right) - \operatorname{Tr}\left(\boldsymbol{U}^\top \boldsymbol{B}_t\right)\right) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}.$$

Again, we are required to derive the optimal solution by examining the subgradient of the last term, which is given by

$$\partial \|\boldsymbol{U}\|_{2,\infty} = \boldsymbol{Q}\boldsymbol{W}, \ q_{ii} \neq 0 \text{ if and only if } i \in \mathcal{I}, \ q_{ij} = 0 \text{ for } i \neq j,$$

where each row of $\boldsymbol{W}$ is as follows:

$$\boldsymbol{w}(i) = \frac{1}{\|\boldsymbol{u}(i)\|}\boldsymbol{u}(i), \ \forall\, 1 \leq i \leq p. \tag{4.24}$$

4.3.3 Memory and Computational Cost

As one of the main contributions of this work, our OMRMD algorithm (i.e., Algorithm 2) is appealing for large-scale problems (the regime $r < d \ll n$) since the memory cost is independent of n. To see this, note that when computing the optimal coefficients and noise, only $\boldsymbol{z}_t$ and $\boldsymbol{U}_{t-1}$ are accessed, which costs $\mathcal{O}(dr)$ memory. To store the accumulation matrix $\boldsymbol{A}_t$, we need $\mathcal{O}\left(r^2\right)$ memory while that for $\boldsymbol{B}_t$ is $\mathcal{O}(dr)$. Finally, we find that only $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$ are needed for the computation of the new iterate $\boldsymbol{U}_t$. Therefore, the total memory cost of OMRMD is $\mathcal{O}(dr)$, i.e., independent of n. In contrast, the SDP formulation introduced by [138] requires $\mathcal{O}\left((d+n)^2\right)$ memory usage, the local-search heuristic algorithm [121] needs $\mathcal{O}(r(d+n))$ and no convergence guarantee was derived. Even for a recently proposed algorithm [87], they require to store the entire data matrix and thus the memory cost is $\mathcal{O}(dn)$.

In terms of computational efficiency, our algorithm can be fast. One may have noticed that the computation is dominated by solving Problem (4.14). The computational complexity of (4.17) involves an inverse of a $r \times r$ matrix followed by a matrix-matrix and a matrix-vector multiplication, totally $\mathcal{O}\left(dr^2\right)$. For the basis update, obtaining a subgradient of the squared $\ell_{2,\infty}$-norm is $\mathcal{O}(dr)$ since we need to calculate the ℓ_2 norm for all rows of $\boldsymbol{U}$ followed by a multiplication with a diag-

onal matrix (see (4.23)). A one-pass update for the columns of $\boldsymbol{U}$, as shown in Algorithm 5 costs $\mathcal{O}\left(dr^2\right)$. Note that the quadratic dependency on r is mild in the low-rank setting.

4.4 Theoretical Analysis and Proof Sketch

In this section we present our main theoretical result regarding the validity of the proposed algorithm. We first discuss some necessary assumptions.

4.4.1 Assumptions

$(A1)$ The observed samples are independent and identically distributed (i.i.d.) with a compact support $\mathcal{Z}$. This is a very common scenario in real-world applications.

$(A2)$ The surrogate functions $g_t(\boldsymbol{U})$ in (4.13) are strongly convex. In particular, we assume that the smallest singular value of the positive semi-definite matrix $\frac{1}{t}\boldsymbol{A}_t$ defined in Algorithm 2 is not smaller than some positive constant β_1.

$(A3)$ The minimizer for (4.12) is unique. Notice that $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e})$ is strongly convex w.r.t. $\boldsymbol{e}$ and convex w.r.t. $\boldsymbol{v}$. We can enforce this assumption by adding a jitter $\frac{\epsilon}{2}\|\boldsymbol{v}\|_2^2$ to the objective function, where ϵ is a small positive constant.

4.4.2 Main Results

It is easy to see that Algorithm 2 is devised to optimize the empirical loss function (4.11). In stochastic optimization, we are mainly interested in the expected loss function, which is defined as the averaged loss incurred when the number of samples goes to infinity. If we assume that each sample is independently and identically distributed (i.i.d.), we have

$$f(\boldsymbol{U}) := \lim_{n\to\infty} f_n(\boldsymbol{U}) = \mathbb{E}_{\boldsymbol{z}}[\ell(\boldsymbol{z}, \boldsymbol{U})]. \tag{4.25}$$

The main theoretical result of this work is stated as follows.

Theorem 4.3 (Convergence to a stationary point of the expected loss function)**.** *Let $\{\boldsymbol{U}_t\}_{t=1}^{\infty}$ be the sequence of solutions produced by Algorithm 2. Then, the sequence converges to a stationary point*

of the expected loss function (4.25) *when* t *tends to infinity.*

Remark 9. The theorem establishes the validity of our algorithm. Note that on one hand, the transformation (4.4) facilitates an amenable way for the online implementation of the max-norm. On the other hand, due to the non-convexity of our new formulation (4.6), it is generally hard to desire a local, or a global minimizer [14]. Although Burer and Monteiro [26] showed that any local minimum of an SDP is also the global optimum under some conditions (note that the max-norm problem can be transformed to an SDP [138]), it is not clear how to determine that a solution is a local optimum or a stationary point. Very recently, [16] showed that global convergence is possible for a family of batch methods. Yet, it is not clear how to apply their results in the stochastic setting. From the empirical study in Section 4.6, we find that the solutions produced by our algorithm always converge to the global optimum when the samples are drawn from a i.i.d. Gaussian distribution.

4.4.3 Proof Outline

The essential tools for our analysis are from stochastic approximation [22] and asymptotic statistics [144]. There are four key stages in our proof and one may find the full proof in Appendix 4.A.

Stage I. We first show that all the stochastic variables $\{\boldsymbol{U}_t, \boldsymbol{v}_t, \boldsymbol{e}_t\}_{t=1}^{\infty}$ are uniformly bounded. The property is crucial because it justifies that the problem we are solving is well-defined. Also, the uniform boundedness will be heavily used for deriving subsequent important results, e.g., the Lipschitz property of the surrogate function.

Proposition 4.4 (Uniform bound of all stochastic variables)**.** *Let* $\{\boldsymbol{v}_t, \boldsymbol{e}_t, \boldsymbol{U}_t\}_{t=1}^{\infty}$ *be the sequence of the solutions produced by Algorithm 2. Then,*

1. *For any* $t > 0$*, the optimal solutions* $\boldsymbol{v}_t$ *and* $\boldsymbol{e}_t$ *are uniformly bounded.*
2. *For any* $t > 0$*, the accumulation matrices* $\frac{1}{t}\boldsymbol{A}_t$ *and* $\frac{1}{t}\boldsymbol{B}_t$ *are uniformly bounded.*
3. *There exists a compact set* $\mathcal{U}$*, such that for any* $t > 0$*, we have* $\boldsymbol{U}_t \in \mathcal{U}$*.*

Proof. (Sketch) The uniform bound of $\boldsymbol{e}_t$ follows by constructing a trivial solution $(\mathbf{0}, \mathbf{0})$ for (4.9), which results in an upper bound for the optimum of the objective function. Notably, the upper bound here only involves a quantity on $\|\boldsymbol{z}_t\|$, which is assumed to be uniformly bounded. Since $\boldsymbol{v}_t$

is always upper bounded by the unit, the first claim follows. The second claim follows immediately by combining the first claim and Assumption $(A1)$. In order to show that $\boldsymbol{U}_t$ is uniformly bounded, we utilize the first order optimality condition of the surrogate (4.13). Since $\frac{1}{t}\boldsymbol{A}_t$ is positive definite, we can represent $\boldsymbol{U}_t$ in terms of $\frac{1}{t}\boldsymbol{B}_t$, $\boldsymbol{G}_t$ and the inverse of $\frac{1}{t}\boldsymbol{A}_t$, where $\boldsymbol{G}_t$ is the subgradient, whose Frobenius norm is in turn bounded by that of $\boldsymbol{U}_t$. Hence, it follows that $\boldsymbol{U}_t$ can be uniformly bounded. □

Note that [99, 59] assumed that the dictionary (or basis) is uniformly bounded. In the above proposition, we prove that such a condition naturally holds in our case.

Corollary 4.5 (Uniform bound and Lipschitz of the surrogate)**.** *Following the notation in Proposition 4.4, we have for all $t > 0$,*

1. $\tilde{\ell}\left(\boldsymbol{z}_t, \boldsymbol{U}_t, \boldsymbol{v}_t, \boldsymbol{e}_t\right)$ (4.9) *and* $\ell\left(\boldsymbol{z}_t, \boldsymbol{U}_t\right)$ (4.12) *are both uniformly bounded.*
2. *The surrogate function, i.e.,* $g_t(\boldsymbol{U})$ *defined in* (4.13) *is uniformly bounded over* $\mathcal{U}$.
3. *Moreover,* $g_t(\boldsymbol{U})$ *is uniformly Lipschitz over the compact set* $\mathcal{U}$.

Stage II. We next show that the positive stochastic process $\{g_t(\boldsymbol{U}_t)\}_{t=1}^{\infty}$ converges almost surely. To establish the convergence, we verify that $\{g_t(\boldsymbol{U}_t)\}_{t=1}^{\infty}$ is a quasi-martingale [22] that converges almost surely. To this end, we illustrate that the expectation of the discrepancy of $g_{t+1}(\boldsymbol{U}_{t+1})$ and $g_t(\boldsymbol{U}_t)$ can be upper bounded by a family of functions $\ell(\cdot, \boldsymbol{U})$ indexed by $\boldsymbol{U} \in \mathcal{U}$. Then we show that the family of the functions is P-Donsker [144], the summands of which concentrate around its expectation within an $\mathcal{O}\left(1/\sqrt{n}\right)$ ball almost surely. Therefore, we conclude that $\{g_t(\boldsymbol{U}_t)\}_{t=1}^{\infty}$ is a quasi-martingale and converges almost surely.

Proposition 4.6. *Let* $\boldsymbol{U} \in \mathcal{U}$ *and denote the minimizer of* $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e})$ *as:*

$$\{\boldsymbol{v}^*, \boldsymbol{e}^*\} = \underset{\boldsymbol{v}, \boldsymbol{e}, \|\boldsymbol{v}\| \leq 1}{\arg\min} \frac{1}{2} \|\boldsymbol{z} - \boldsymbol{U}\boldsymbol{v} - \boldsymbol{e}\|^2 + \lambda_2 \tilde{h}(\boldsymbol{e}).$$

Then, the function $\ell(\boldsymbol{z}, \boldsymbol{U})$ *defined in Problem* (4.12) *is continuously differentiable and*

$$\nabla_{\boldsymbol{U}} \ell(\boldsymbol{z}, \boldsymbol{U}) = (\boldsymbol{U}\boldsymbol{v}^* + \boldsymbol{e}^* - \boldsymbol{z}){\boldsymbol{v}^*}^{\top}.$$

Furthermore, $\ell(\boldsymbol{z}, \cdot)$ is uniformly Lipschitz over the compact set $\mathcal{U}$.

Proof. The gradient of $\ell(\boldsymbol{z}, \cdot)$ follows from Lemma 4.14. Since each term of $\nabla_{\boldsymbol{U}} \ell(\boldsymbol{z}, \boldsymbol{U})$ is uniformly bounded, we conclude the uniform Lipschitz property of $\ell(\boldsymbol{z}, \boldsymbol{U})$ w.r.t. $\boldsymbol{U}$. □

Corollary 4.7 (Uniform bound and Lipschitz of the empirical loss)**.** *Let $f_t(\boldsymbol{U})$ be the empirical loss function defined in* (4.11)*. Then $f_t(\boldsymbol{U})$ is uniformly bounded and Lipschitz over the compact set $\mathcal{U}$.*

Corollary 4.8 (P-Donsker of $\ell(\boldsymbol{z}, \boldsymbol{U})$)**.** *The set of measurable functions $\{\ell(\boldsymbol{z}, \boldsymbol{U}),\ \boldsymbol{U} \in \mathcal{U}\}$ is P-Donsker (see definition in Lemma 4.13).*

Proposition 4.9 (Concentration of the empirical loss)**.** *Let $f_t(\boldsymbol{U})$ and $f(\boldsymbol{U})$ be the empirical and expected loss functions we defined in* (4.11) *and* (4.25)*. Then we have*

$$\mathbb{E}\big[\sqrt{t}\,\|f_t - f\|_\infty\big] = \mathcal{O}(1).$$

Proof. Since $\ell(\boldsymbol{z}, \boldsymbol{U})$ is uniformly upper bounded (Corollary 4.5) and is always non-negative, its square is uniformly upper bounded, hence its expectation. Together with Corollary 4.8, Lemma 4.13 applies. □

Theorem 4.10 (Convergence of the surrogate)**.** *The sequence $\{g_t(\boldsymbol{U}_t)\}_{t=1}^\infty$ we defined in* (4.13) *converges almost surely, where $\{\boldsymbol{U}_t\}_{t=1}^\infty$ is the solution produced by Algorithm 2. Moreover, the infinite summation $\sum_{t=1}^\infty |\mathbb{E}[g_{t+1}(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_t) \mid \mathcal{F}_t]|$ is bounded almost surely.*

Proof. The theorem follows by showing that the sequence of $\{g_t(\boldsymbol{U}_t)\}_{t=1}^\infty$ is a quasi-martingale, and hence converges almost surely. To see this, we note that for any $t > 0$, the expectation of the difference $g_{t+1}(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_t)$ conditioned on the past information $\mathcal{F}_t$ is bounded by $\sup_{\boldsymbol{U}}(f(\boldsymbol{U}) - f_t(\boldsymbol{U}))/(t+1)$, which is of order $\mathcal{O}\left(1/(\sqrt{t}(t+1))\right)$ due to Proposition 4.9. Hence, Lemma 4.15 applies. □

Stage III. Now we prove that the sequence of the empirical loss function, $\{f_t(\boldsymbol{U}_t)\}_{t=1}^\infty$ defined in (4.11) converges almost surely to the same limit of its surrogate $\{g_t(\boldsymbol{U}_t)\}_{t=1}^\infty$. According to the central limit theorem, we assert that $f_t(\boldsymbol{U}_t)$ also converges almost surely to the expected loss $f(\boldsymbol{U}_t)$ defined in (4.25), implying that $g_t(\boldsymbol{U}_t)$ and $f(\boldsymbol{U}_t)$ converge to the same limit almost surely.

We first establish the numerical convergence of the basis sequence $\{\boldsymbol{U}_t\}_{t=1}^{\infty}$, based on which we show the convergence of $\{f_t(\boldsymbol{U}_t)\}_{t=1}^{\infty}$ by applying Lemma 4.16.

Proposition 4.11 (Numerical convergence of the basis component)**.** *Let $\{\boldsymbol{U}_t\}_{t=1}^{\infty}$ be the basis sequence produced by the Algorithm 2. Then, for any $t > 0$, we have*

$$\|\boldsymbol{U}_{t+1} - \boldsymbol{U}_t\|_F = \mathcal{O}\left(\frac{1}{t}\right). \tag{4.26}$$

Theorem 4.12 (Convergence of the empirical and expected loss)**.** *Let $\{f(\boldsymbol{U}_t)\}_{t=1}^{\infty}$ be the sequence of the expected loss where $\{\boldsymbol{U}_t\}_{t=1}^{\infty}$ is the sequence of the solutions produced by the Algorithm 2. Then, we have*

1. *The sequence of the empirical loss $\{f_t(\boldsymbol{U}_t)\}_{t=1}^{\infty}$ converges almost surely to the same limit of the surrogate.*

2. *The sequence of the expected loss $\{f(\boldsymbol{U}_t)\}_{t=1}^{\infty}$ converges almost surely to the same limit of the surrogate.*

Proof. Let $b_t = g_t(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)$. We show that infinite series $\sum_{t=1}^{\infty} b_t/(t+1)$ is bounded by applying the central limit theorem to $f(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)$ and the result of Theorem 4.10. We further prove that $|b_{t+1} - b_t|$ can be bounded by $\mathcal{O}(1/t)$, due to the uniform boundedness and Lipschitz of $g_t(\boldsymbol{U}_t)$, $f_t(\boldsymbol{U}_t)$ and $\ell(\boldsymbol{z}_t, \boldsymbol{U}_t)$. According to Lemma 4.16, we conclude the convergence of $\{b_t\}_{t=1}^{\infty}$ to zero. Hence the first claim. The second claim follows immediately owing to the central limit theorem. □

Final Stage. According to Claim 2 of Theorem 4.12 and the fact that $\mathbf{0}$ belongs to the subgradient of $g_t(\boldsymbol{U})$ evaluated at $\boldsymbol{U} = \boldsymbol{U}_t$, we are to show the gradient of $f(\boldsymbol{U})$ taking at $\boldsymbol{U}_t$ vanishes as t tends to infinity, which establishes Theorem 4.3. To this end, we note that since $\{\boldsymbol{U}_t\}_{t=1}^{\infty}$ is uniformly bounded, the non-differentiable term $\frac{1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2$ vanishes as t goes to infinity, implying the differentiability of $g_\infty(\boldsymbol{U}_\infty)$, i.e. $\nabla g_\infty(\boldsymbol{U}_\infty) = \mathbf{0}$. On the other hand, we show that the gradient of $f(\boldsymbol{U})$ and that of $g_t(\boldsymbol{U})$ are always Lipschitz on the compact set $\mathcal{U}$, implying the existence of their second order derivative even when $t \to \infty$. Thus, by taking a first order Taylor expansion and let t go to infinity, we establish the main theorem.

4.5 Connection to Matrix Completion

While we mainly focus on the matrix decomposition problem, our method can be extended to the matrix completion (MC) problem [27, 36] with max-norm regularization [31, 32] – another popular topic in machine learning and signal processing. We focus on the max-norm regularized MC problem with squared Frobenius loss widely considered in the literature, which can be described as follows:

$$\min_{\boldsymbol{X}} \frac{1}{2} \left\| \mathcal{P}_{\Omega} \left(\boldsymbol{Z} - \boldsymbol{X} \right) \right\|_F^2 + \frac{\lambda}{2} \left\| \boldsymbol{X} \right\|_{\max}^2 ,$$

where Ω is the set of indices of observed entries in $\boldsymbol{Z}$ and $\mathcal{P}_{\Omega}(\boldsymbol{M})$ is the orthogonal projection onto the span of matrices vanishing outside of Ω so that the (i,j)-th entry of $\mathcal{P}_{\Omega}(\boldsymbol{M})$ is equal to m_{ij} if $(i,j) \in \Omega$ and zero otherwise. Interestingly, the max-norm regularized MC problem can be cast into our framework. To see this, let us introduce an auxiliary matrix $\boldsymbol{M}$, with $m_{ij} = c > 0$ if $(i,j) \in \Omega$ and $m_{ij} = 1/c$ otherwise. The reformulated MC problem,

$$\min_{\boldsymbol{X},\boldsymbol{E}} \frac{1}{2} \left\| \boldsymbol{Z} - \boldsymbol{X} - \boldsymbol{E} \right\|_F^2 + \frac{\lambda}{2} \left\| \boldsymbol{X} \right\|_{\max}^2 + \left\| \boldsymbol{M} \circ \boldsymbol{E} \right\|_1 , \tag{4.27}$$

where "$\circ$" denotes the entry-wise product, is similar to our MRMD formulation (4.1). And it is easy to show that when c tends to infinity, the reformulated problem converges to the original MC problem.

4.5.1 Online Implementation

We now derive a stochastic implementation for the max-norm regularized MC problem. Note that the only difference between the Problem (4.27) and Problem (4.1) is the ℓ_1 regularization on $\boldsymbol{E}$, which results a new penalty on $\boldsymbol{e}$ for $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e})$ (which is originally defined in (4.9)):

$$\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e}) = \frac{1}{2} \left\| \boldsymbol{z} - \boldsymbol{U}\boldsymbol{v} - \boldsymbol{e} \right\|^2 + \left\| \boldsymbol{m} \circ \boldsymbol{e} \right\|_1 . \tag{4.28}$$

Here, $\boldsymbol{m}$ is a column of the matrix $\boldsymbol{M}$ in (4.27). According to the definition of $\boldsymbol{M}$, $\boldsymbol{m}$ is a vector with element value being either c or $1/c$. Let us define two support sets as follows:

$$\Omega_1 := \{i \mid m_i = c, 1 \leq i \leq p\},$$
$$\Omega_2 := \{i \mid m_i = 1/c, 1 \leq i \leq p\},$$

where m_i is the ith element of vector $\boldsymbol{m}$. In this way, the newly defined $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e})$ can be written as

$$\begin{aligned}\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e}) = &\left(\frac{1}{2}\left\|\boldsymbol{z}_{\Omega_1} - (\boldsymbol{U}\boldsymbol{v})_{\Omega_1} - \boldsymbol{e}_{\Omega_1}\right\|^2 + c\left\|\boldsymbol{e}_{\Omega_1}\right\|_1\right) \\ &+ \left(\frac{1}{2}\left\|\boldsymbol{z}_{\Omega_2} - (\boldsymbol{U}\boldsymbol{v})_{\Omega_2} - \boldsymbol{e}_{\Omega_2}\right\|^2 + \frac{1}{c}\left\|\boldsymbol{e}_{\Omega_2}\right\|_1\right). \end{aligned} \tag{4.29}$$

Notably, as Ω_1 and Ω_2 are disjoint, given $\boldsymbol{z}$, $\boldsymbol{U}$ and $\boldsymbol{v}$, the variable $\boldsymbol{e}$ in (4.29) can be optimized by soft-thresholding in a separate manner:

$$\boldsymbol{e}_{\Omega_1} = \mathcal{S}_c[\boldsymbol{z}_{\Omega_1} - (\boldsymbol{U}\boldsymbol{v})_{\Omega_1}], \quad \boldsymbol{e}_{\Omega_2} = \mathcal{S}_{1/c}[\boldsymbol{z}_{\Omega_2} - (\boldsymbol{U}\boldsymbol{v})_{\Omega_2}]. \tag{4.30}$$

Hence, we obtain Algorithm 6 for the online max-norm regularized matrix completion (OMRMC) problem. The update principle for $\boldsymbol{v}$ is the same as we described in Algorithm 4 and that for $\boldsymbol{e}$ is given by (4.30). Note that we can use Algorithm 5 to update $\boldsymbol{U}$ as usual.

4.5.2 ℓ_∞-norm Constrained Variant

In some matrix completion applications, one may have to take another ℓ_∞-norm constraint into account, i.e.,

$$\|\boldsymbol{X}\|_\infty \leq \tau, \text{ for some } \tau > 0. \tag{4.31}$$

For example, the rating value of the Netflix dataset is not greater than 5. In the 1-bit setting, the entries of a matrix can either be 1 or -1 [47]. Other examples can be found in, e.g., [82]. Interestingly, Algorithm 6 can be adjusted to such a constraint.

Algorithm 6 Online Max-Norm Regularized Matrix Completion

Require: $\boldsymbol{Z} \in \mathbb{R}^{d\times n}$ (observed samples), parameters λ_1 and λ_2, $\boldsymbol{U}_0 \in \mathbb{R}^{d\times r}$ (initial basis), zero matrices $\boldsymbol{A}_0 \in \mathbb{R}^{d\times d}$ and $\boldsymbol{B}_0 \in \mathbb{R}^{d\times r}$.

Ensure: optimal basis $\boldsymbol{U}_t$.

1: **for** $t = 1$ to n **do**

2: Access the t-th sample $\boldsymbol{z}_t$.

3: Compute the coefficient and noise:

$$\begin{aligned}\{\boldsymbol{v}_t, \boldsymbol{e}_t\} &= \mathop{\arg\min}_{\boldsymbol{v},\boldsymbol{e},\|\boldsymbol{v}\|_2^2\le 1} \tilde{\ell}(\boldsymbol{z}_t, \boldsymbol{U}_{t-1}, \boldsymbol{v}, \boldsymbol{e}) \\ &= \mathop{\arg\min}_{\boldsymbol{v},\boldsymbol{e},\|\boldsymbol{v}\|_2^2\le 1} \left(\frac{1}{2}\|\boldsymbol{z}_t - \boldsymbol{U}_{t-1}\boldsymbol{v} - \boldsymbol{e}\|_2^2 + \|\boldsymbol{m}_t \circ \boldsymbol{e}\|_1\right).\end{aligned}$$

4: Compute the accumulation matrices $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$:

$$\begin{aligned}\boldsymbol{A}_t &\leftarrow \boldsymbol{A}_{t-1} + \boldsymbol{v}_t\boldsymbol{v}_t^\top, \\ \boldsymbol{B}_t &\leftarrow \boldsymbol{B}_{t-1} + (\boldsymbol{z}_t - \boldsymbol{e}_t)\,\boldsymbol{v}_t^\top.\end{aligned}$$

5: Compute the basis $\boldsymbol{U}_t$ by optimizing the surrogate function (4.13):

$$\begin{aligned}\boldsymbol{U}_t &= \mathop{\arg\min}_{\boldsymbol{U}} \frac{1}{t}\sum_{i=1}^{t} \tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{U}, \boldsymbol{v}_i, \boldsymbol{e}_i) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2 \\ &= \mathop{\arg\min}_{\boldsymbol{U}} \frac{1}{t}\sum_{i=1}^{t} \left(\frac{1}{2}\|\boldsymbol{z}_i - \boldsymbol{U}\boldsymbol{v}_i - \boldsymbol{e}_i\|^2 + \|\boldsymbol{m}_i \circ \boldsymbol{e}_i\|_1\right) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2 \\ &= \mathop{\arg\min}_{\boldsymbol{U}} \frac{1}{t}\sum_{i=1}^{t} \left(\frac{1}{2}\|\boldsymbol{z}_i - \boldsymbol{U}\boldsymbol{v}_i - \boldsymbol{e}_i\|^2\right) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2 \\ &= \mathop{\arg\min}_{\boldsymbol{U}} \frac{1}{t}\left(\frac{1}{2}\operatorname{Tr}\left(\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{A}_t\right) - \operatorname{Tr}\left(\boldsymbol{U}^\top\boldsymbol{B}_t\right)\right) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2.\end{aligned}$$

6: **end for**

To see this, we observe that the constraint $\|\boldsymbol{X}\|_\infty \le \tau$ amounts to restricting

$$|x_{ij}| \le \tau$$

for all entries x_{ij} of $\boldsymbol{X}$. Due to the matrix factorization $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top$, we know that it requires

$$\left|\boldsymbol{u}(i)\boldsymbol{v}(j)^\top\right| \le \tau,\ \forall\, i \in\ [p],\ \forall\, j \in [n], \tag{4.32}$$

where we recall that $\boldsymbol{u}(i)$ and $\boldsymbol{v}(j)$ are the ith row of $\boldsymbol{U}$ and the jth row of $\boldsymbol{V}$, respectively. Propo-

sition 4.1 already ensures

$$\|\boldsymbol{v}(j)\| \leq 1, \ \forall \, j \in [n].$$

Since $\left|\boldsymbol{u}(i)\boldsymbol{v}(j)^\top\right| \leq \|\boldsymbol{u}(i)\| \cdot \|\boldsymbol{v}(j)\|$, we obtain a *sufficient* condition for (4.32):

$$\|\boldsymbol{u}(i)\| \leq \tau, \ \forall \, i \in [n].$$

That is,

$$\|\boldsymbol{U}\|_{2,\infty} \leq \tau,$$

which can easily be fulfilled by an orthogonal projection onto the ℓ_2 ball with radius τ, i.e., if $\|\boldsymbol{U}_t\|_{2,\infty} > \tau$, we set $\boldsymbol{U}_t \leftarrow \frac{\tau}{\|\boldsymbol{U}_t\|_{2,\infty}} \boldsymbol{U}_t$.

4.5.3 Other Types of Loss Functions

We in this chapter emphasize on the squared Frobenius loss for the max-norm regularized problems. There is also solid theoretical analysis for other formulations, e.g., logistic regression and probit regression [31]. Unfortunately, it seems that one cannot trivially extend the proposed online algorithms to a general loss function. To be more precise, for Frobenius (or ℓ_2) loss, we are guaranteed with a nice property that minimizing the surrogate (4.20) is equivalent to solving (4.22), for which only $\mathcal{O}(dr)$ memory is needed. For general models, such a property does not hold and we conjecture that more technique is needed to find a good approximation to (4.20).

4.6 Experiments

In this section, we report numerical results on synthetic data to demonstrate the effectiveness and robustness of our online max-norm regularized matrix decomposition (OMRMD) algorithm. Some experimental settings are used throughout this section, as elaborated below.

Data Generation. The simulation data are generated by following a similar procedure in [34]. The clean data matrix $\boldsymbol{X}$ is produced by $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{d \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times r}$. The entries of

$\boldsymbol{U}$ and $\boldsymbol{V}$ are i.i.d. sampled from the normal distribution $N(0,1)$. We choose sparse corruption in the experiments, and introduce a parameter ρ to control the sparsity of the corruption matrix $\boldsymbol{E}$, i.e., a ρ-fraction of the entries are non-zero whose locations are uniformly sampled and the magnitude follows a uniform distribution over $[-1000, 1000]$. Finally, the observation matrix $\boldsymbol{Z}$ is produced by $\boldsymbol{Z} = \boldsymbol{X} + \boldsymbol{E}$.

Baselines. We mainly compare with two methods: Principal Component Pursuit (PCP) and online robust PCA (OR-PCA). PCP is the state-of-the-art batch method for subspace recovery, which was presented as a robust formulation of PCA in [34]. OR-PCA is an online implementation of PCP, which also achieves state-of-the-art performance over the online subspace recovery algorithms. Sometimes, to show the robustness, we will also report the results of online PCA [3], which incrementally learns the principal components without taking the noise into account.

Evaluation Metric. Our goal is to estimate the correct subspace for the underlying data. Here, we evaluate the fitness of our estimated subspace basis $\hat{\boldsymbol{U}}$ (with columns normalized to have unit length) and the ground truth basis $\boldsymbol{U}$ by the Expressed Variance (EV) [153]:

$$\mathrm{EV}(\hat{\boldsymbol{U}}, \boldsymbol{U}) := \frac{\mathrm{Tr}\left(\hat{\boldsymbol{U}}^\top \boldsymbol{U}\boldsymbol{U}^\top \hat{\boldsymbol{U}}\right)}{\mathrm{Tr}\left(\boldsymbol{U}\boldsymbol{U}^\top\right)}. \tag{4.33}$$

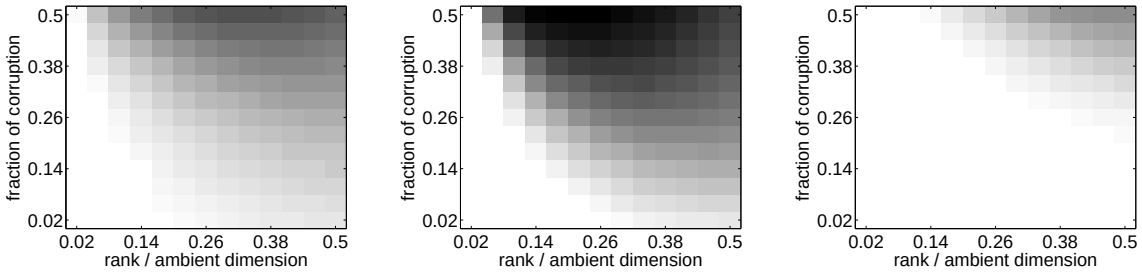
The values of EV range in $[0, 1]$ and a higher value indicates a more accurate recovery.

Other Settings. Throughout the experiments, we set the ambient dimension $d = 400$, the total number of samples $n = 5000$ and pick the value of r as the true rank unless otherwise specified. We fix the tunable parameter $\lambda_1 = \lambda_2 = 1/\sqrt{d}$, and use default parameters for all baselines we compare with. Each experiment is repeated 10 times and we report the averaged EV as the result.

4.6.1 Robustness

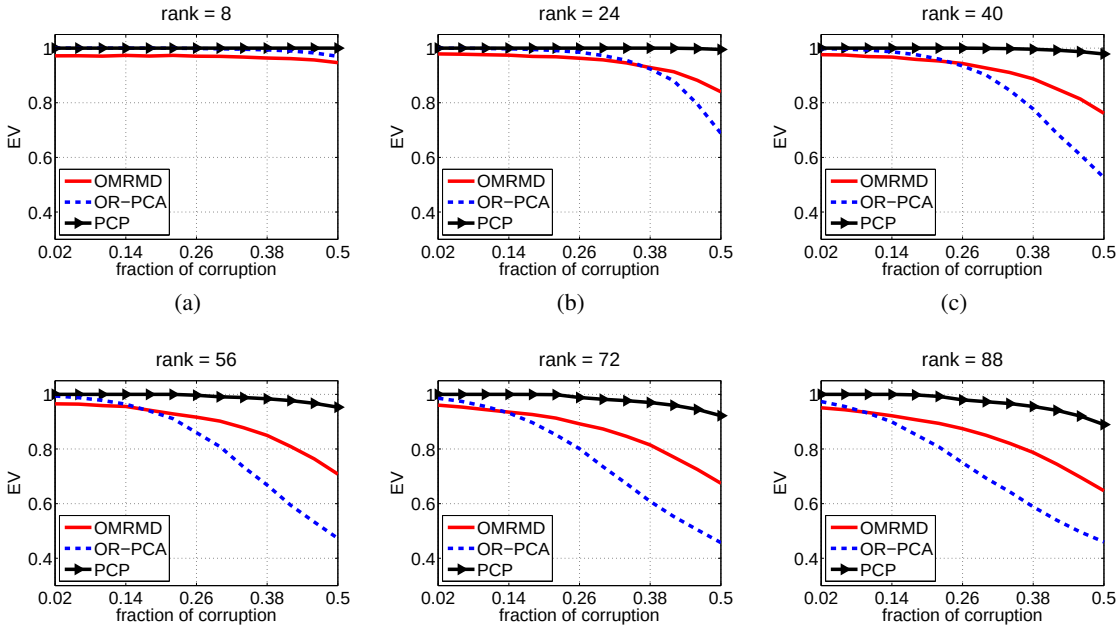
We first study the robustness of OMRMD, measured by the EV value of its output after accessing the last sample, and compare it to the nuclear norm based OR-PCA and the batch algorithm PCP. In order to make a detailed examination, we vary the true rank from $0.02d$ to $0.5d$, with a step size $0.04d$, and the corruption fraction ρ from 0.02 to 0.5, with a step size 0.04.

The general results are illustrated in Figure 4.1 where a brighter color means a higher EV (hence



(a) OMRMD (b) OR-PCA (c) PCP

Figure 4.1: **Performance of subspace recovery under different rank and corruption fraction.**



(a) (b) (c) (d) (e) (f)

Figure 4.2: **EV value against corruption fractions when the matrix has a relatively low rank.**

better performance). We observe that for easy tasks (i.e., few corruption and low rank case), both OMRMD and OR-PCA perform comparably. However, for more difficult cases, OMRMD outperforms OR-PCA. In order to further investigate this phenomenon, we plot the EV curve against the fraction of corruption under a given matrix rank. In particular, we group the results into two parts, one with relatively low rank (Figure 4.2) and the other with middle level of rank (Figure 4.3). Figure 4.2 indicates that when manipulating a low-rank matrix, OR-PCA works as well as OMRMD under a low level of noise. For instance, the EV produced by OR-PCA is as close as that of OMRMD for rank less than 40 and ρ no more than 0.26. However, when the rank becomes

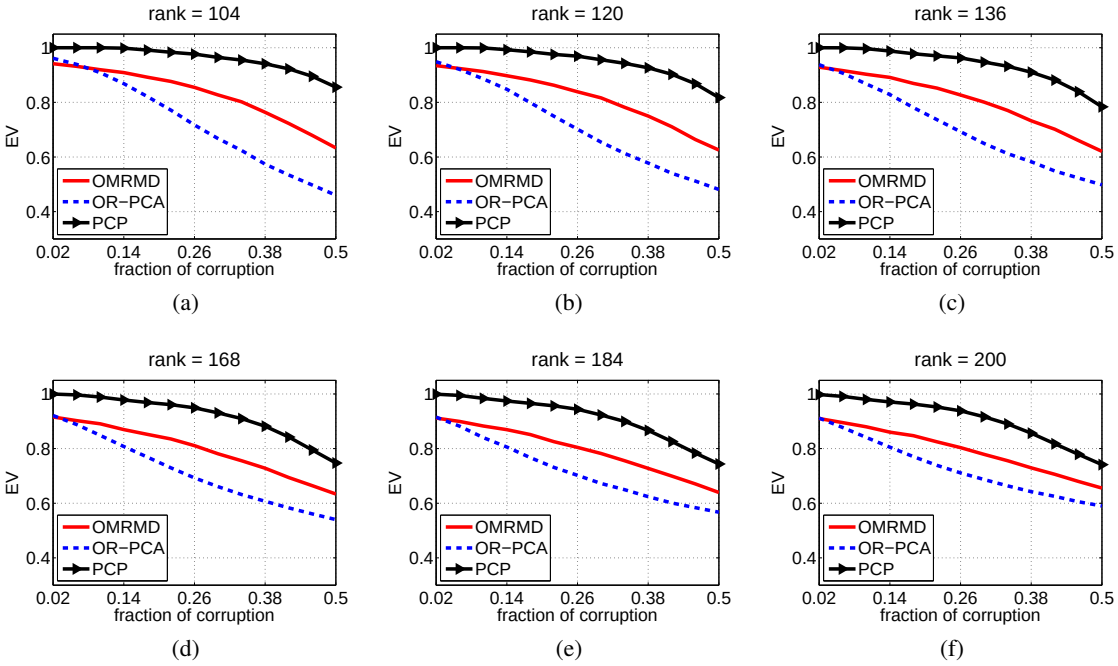


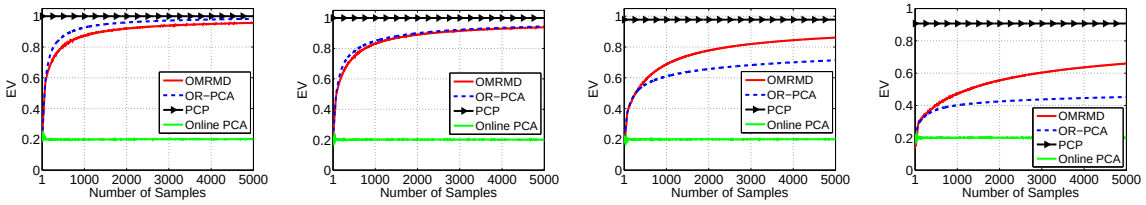
Figure 4.3: **EV value against corruption fractions when the matrix has a middle level of rank.**

larger, OR-PCA degrades quickly compared to OMRMD. This is possibly because the max-norm is a tighter approximation to the matrix rank. Since PCP is a batch formulation and accesses all the data in each iteration, it always achieves the best recovery performance.

4.6.2 Convergence Rate

We next study the convergence of OMRMD by plotting the EV curve against the number of samples. Besides OR-PCA and PCP, we also add online PCA [3] as a baseline algorithm. The results are illustrated in Figure 4.4 where we set $d = 400$ and the true rank as 80. As expected, PCP achieves the best performance since it is a batch method and needs to access all the data during optimization. Online PCA degrades significantly even with low corruption (Figure 4.4a). OMRMD is comparable to OR-PCA when the corruption is low (Figure 4.4a), and converges significantly faster when the data is grossly corrupted (Figure 4.4c and 4.4d). This observation agrees with Figure 4.1, and again suggests that in the noisy scenario, max-norm may be a better fit than the nuclear norm.

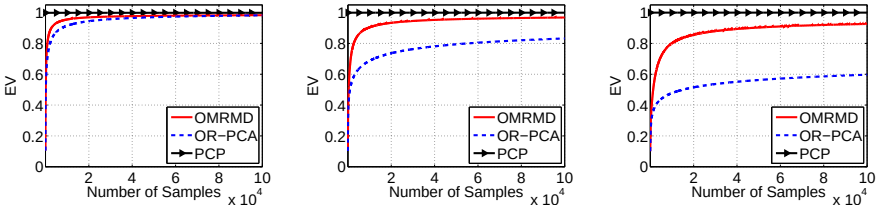
Indeed, OMRMD converges much faster even in large scale problems. In Figure 4.5, we compare the convergence rate of OMRMD and OR-PCA under different ambient dimensions. The rand of the data are set with $0.1d$, indicating a low-rank structure of the underlying data. Again, we



(a) $\rho = 0.01$ (b) $\rho = 0.1$ (c) $\rho = 0.3$ (d) $\rho = 0.5$

Figure 4.4: **EV value against number of samples under different corruption fractions.**

assume the rank is known so $r = 0.1d$. The error corruption ρ is fixed to 0.3 – a difficult task for recovery. We observe that for high dimensional cases ($d = 1000$ and $d = 3000$), OMRMD significantly outperforms OR-PCA. For example, in Figure 4.5b, OMRMD achieves the EV value of 0.8 only with accessing about 2000 samples, whereas OR-PCA needs to reveal $60,000$ samples to obtain the same accuracy!



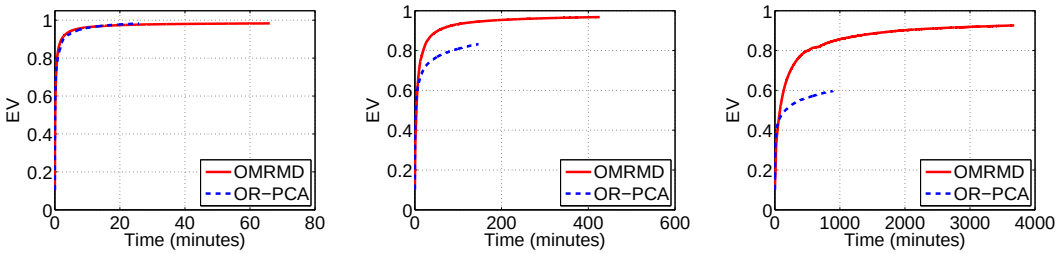
(a) $d = 400$ (b) $d = 1000$ (c) $d = 3000$

Figure 4.5: **EV value against number of samples under different ambient dimensions.** The rank $r = 0.1d$ and the corruption fraction $\rho = 0.3$.

4.6.3 Computational Complexity

We note that OMRMD is a little bit inferior to OR-PCA in terms of computation per iteration, as our algorithm may solve a dual problem to optimize $\boldsymbol{v}$ (see Algorithm 4) if the initial solution $\boldsymbol{v}_0$ violates the constraint. We plot the EV curve with respect to the running time in Figure 4.6. It shows that, OR-PCA is about 3 times faster than OMRMD when processing a data point. However, we point out here that we emphasize on the convergence rate. That is, given an EV value, how much time the algorithm will take to achieve it. In Figure 4.6c, for example, OMRMD takes 50 minutes to achieve the EV value of 0.6, while OR-PCA uses nearly 900 minutes. From Figure 4.5 and Figure 4.6, it is safe to conclude that OMRMD is superior to OR-PCA in terms of convergence

rate in the price of a little more computation per sample.



(a) $d = 400$ (b) $d = 1000$ (c) $d = 3000$

Figure 4.6: **EV value against time under different ambient dimensions.**

4.7 Conclusion

In this chapter, we have developed an online algorithm for the max-norm regularized matrix decomposition problems. Using the matrix factorization form of the max-norm, we converted the original problem to a constrained one which facilitates an online implementation for solving the batch problem. We have established theoretical guarantees that the sequence of the solutions converges to a stationary point of the expected loss function asymptotically. Moreover, we empirically compared our proposed algorithm with OR-PCA, which is a recently proposed online algorithm for nuclear-norm based matrix decomposition. The simulation results have suggested that the proposed algorithm is more robust than OR-PCA, in particular for hard tasks (i.e., when a large fraction of entries are corrupted). We also have investigated the convergence rate for both OMRMD and OR-PCA, and have shown that OMRMD converges much faster than OR-PCA even in large-scale problems. When acquiring sufficient samples, we observed that our algorithm converges to the batch method PCP, which is a state-of-the-art formulation for subspace recovery. Our experiments, to an extent, suggest that the max-norm might be a tighter relaxation of the rank function compared to the nuclear norm.

4.A Proof Details

4.A.1 Proof of Proposition 4.1

Proof. Let us denote $k = \|\boldsymbol{V}\|_{2,\infty}$. We presume that k is positive. Otherwise, the low-rank component $\boldsymbol{X}$ we aim to recover is a zero matrix, which is of little interest. Now we construct two auxiliary variables $\bar{\boldsymbol{U}} = k\boldsymbol{U} \in \mathbb{R}^{d\times r}$ and $\bar{\boldsymbol{V}} = \frac{1}{k}\boldsymbol{V} \in \mathbb{R}^{n\times r}$. Replacing $\boldsymbol{U}$ and $\boldsymbol{V}$ with $\frac{1}{k}\bar{\boldsymbol{U}}$ and $k\bar{\boldsymbol{V}}$ in (4.5) respectively, we have:

$$\min_{\bar{\boldsymbol{U}},\bar{\boldsymbol{V}},\boldsymbol{E}} \frac{1}{2}\left\|\boldsymbol{Z} - \left(\frac{1}{k}\bar{\boldsymbol{U}}\right)\left(k\bar{\boldsymbol{V}}\right)^\top - \boldsymbol{E}\right\|_F^2 + \frac{\lambda_1}{2}\left\|\frac{1}{k}\bar{\boldsymbol{U}}\right\|_{2,\infty}^2 \left\|k\bar{\boldsymbol{V}}\right\|_{2,\infty}^2 + \lambda_2 h(\boldsymbol{E}).$$

That is, we are to solve

$$\min_{\bar{\boldsymbol{U}},\bar{\boldsymbol{V}},\boldsymbol{E}} \frac{1}{2}\left\|\boldsymbol{Z} - \bar{\boldsymbol{U}}\bar{\boldsymbol{V}}^\top - \boldsymbol{E}\right\|_F^2 + \frac{\lambda_1}{2}\left\|\bar{\boldsymbol{U}}\right\|_{2,\infty}^2 \left\|\bar{\boldsymbol{V}}\right\|_{2,\infty}^2 + \lambda_2 h(\boldsymbol{E}).$$

The fact that $\bar{\boldsymbol{V}} = \frac{1}{k}\boldsymbol{V}$ and k is the maximum of the ℓ_2 row norm of $\boldsymbol{V}$ implies $\left\|\bar{\boldsymbol{V}}\right\|_{2,\infty} = 1$. Therefore, we can reformulate our MRMD problem as a constrained program:

$$\min_{\bar{\boldsymbol{U}},\bar{\boldsymbol{V}},\boldsymbol{E}} \frac{1}{2}\left\|\boldsymbol{Z} - \bar{\boldsymbol{U}}\bar{\boldsymbol{V}}^\top - \boldsymbol{E}\right\|_F^2 + \frac{\lambda_1}{2}\left\|\bar{\boldsymbol{U}}\right\|_{2,\infty}^2 + \lambda_2 h(\boldsymbol{E}), \quad \text{s.t. } \left\|\bar{\boldsymbol{V}}\right\|_{2,\infty}^2 = 1.$$

To see why the above program is equivalent to (4.6), we only need to show that each optimal solutions $(\boldsymbol{U}^*, \boldsymbol{V}^*, \boldsymbol{E}^*)$ of (4.6) must satisfy $\|\boldsymbol{V}^*\|_{2,\infty}^2 = 1$. Suppose that $k = \|\boldsymbol{V}^*\|_{2,\infty} < 1$. Let $\boldsymbol{U}' = k\boldsymbol{U}^*$ and $\boldsymbol{V}' = \frac{1}{k}\boldsymbol{V}^*$. Obviously, $(\boldsymbol{U}', \boldsymbol{V}', \boldsymbol{E}^*)$ are still feasible. However, the objective value becomes

$$\begin{aligned}
&\frac{1}{2}\left\|\boldsymbol{Z} - \boldsymbol{U}'\boldsymbol{V}'^\top - \boldsymbol{E}^*\right\|_F^2 + \frac{\lambda_1}{2}\left\|\boldsymbol{U}'\right\|_{2,\infty}^2 + \lambda_2 h(\boldsymbol{E}^*) \\
=&\frac{1}{2}\left\|\boldsymbol{Z} - \boldsymbol{U}^*\boldsymbol{V}^{*\top} - \boldsymbol{E}^*\right\|_F^2 + \frac{\lambda_1}{2}\cdot k^2\left\|\boldsymbol{U}^*\right\|_{2,\infty}^2 + \lambda_2 h(\boldsymbol{E}^*) \\
<&\frac{1}{2}\left\|\boldsymbol{Z} - \boldsymbol{U}^*\boldsymbol{V}^{*\top} - \boldsymbol{E}^*\right\|_F^2 + \frac{\lambda_1}{2}\left\|\boldsymbol{U}^*\right\|_{2,\infty}^2 + \lambda_2 h(\boldsymbol{E}^*),
\end{aligned}$$

which contradicts the assumption that $(\boldsymbol{U}^*, \boldsymbol{V}^*, \boldsymbol{E}^*)$ is the optimal solution. Thus we complete the proof. □

4.A.2 Proof of Proposition 4.4

Proof. Note that for each $t > 0$, $\|\boldsymbol{v}_t\| \leq 1$. Thus $\boldsymbol{v}_t$ is uniformly bounded. Let us consider the optimization problem (4.14). As the trivial solution $\boldsymbol{v}_t = \mathbf{0}$ and $\boldsymbol{e}_t = \mathbf{0}$ are feasible, we have

$$\tilde{\ell}(\boldsymbol{z}_t, \boldsymbol{U}_{t-1}, \mathbf{0}, \mathbf{0}) = \frac{1}{2} \|\boldsymbol{z}_t\|^2 .$$

Therefore, the optimal solution should satisfy:

$$\frac{1}{2} \|\boldsymbol{z}_t - \boldsymbol{U}_{t-1}\boldsymbol{v}_t - \boldsymbol{e}_t\|^2 + \lambda_2 \|\boldsymbol{e}_t\|_1 \leq \frac{1}{2} \|\boldsymbol{z}_t\|^2 ,$$

which implies

$$\|\boldsymbol{e}_t\|_1 \leq \frac{1}{2\lambda_2} \|\boldsymbol{z}_t\|^2 .$$

Since $\boldsymbol{z}_t$ is uniformly bounded (Assumption $(A1)$), $\boldsymbol{e}_t$ is uniformly bounded.

To examine the uniform bound for $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$, note that

$$\frac{1}{t}\boldsymbol{A}_t = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{v}_i \boldsymbol{v}_i^\top, \quad \frac{1}{t}\boldsymbol{B}_t = \frac{1}{t}\sum_{i=1}^{t} (\boldsymbol{z}_i - \boldsymbol{e}_i)\, \boldsymbol{v}_i^\top .$$

Since for each i, $\boldsymbol{v}_i$, $\boldsymbol{e}_i$ and $\boldsymbol{z}_i$ are uniformly bounded, $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$ are uniformly bounded.

Based on Claim 1 and Claim 2, we prove that $\boldsymbol{U}_t$ can be uniformly bounded. First let us denote $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$ by $\widetilde{\boldsymbol{A}}_t$ and $\widetilde{\boldsymbol{B}}_t$, respectively.

Step 1: According to Claim 2, there exist constants C_1 and C that are uniform over t, such that

$$\left\|\widetilde{\boldsymbol{A}}_t\right\|_F \leq \mathrm{C}_1, \quad \left\|\widetilde{\boldsymbol{B}}_t\right\|_F \leq \mathrm{C}.$$

On the other hand, from Assumption $(A2)$, the eigenvalues of $\widetilde{\boldsymbol{A}}_t$ is lower bounded by a positive constant β_1 that is uniform over t, implying the trace norm (sum of the singular values) of $\widetilde{\boldsymbol{A}}_t$ is

uniformly lower bounded by a positive constant. As all norms are equivalent, we can show that

$$\left\|\widetilde{\boldsymbol{A}}_t\right\|_F \geq \mathrm{C}_0 > 0,$$

where C_0 is a positive constant which is uniform over t.

Recall that $\boldsymbol{U}_t$ is the optimal basis for (4.22). Thus, the subgradient of the objective function taken at $\boldsymbol{U}_t$ should contain zero, that is,

$$\boldsymbol{U}_t\widetilde{\boldsymbol{A}}_t - \widetilde{\boldsymbol{B}}_t + \frac{\lambda_1}{t}\boldsymbol{G}_t = 0,$$

where $\boldsymbol{G}_t$ is the subgradient of $\frac{1}{2}\|\boldsymbol{U}_t\|_{2,\infty}^2$ produced by (4.23). Note that, as all of the eigenvalues of $\widetilde{\boldsymbol{A}}_t$ are lower bounded by a positive constant, $\widetilde{\boldsymbol{A}}_t$ is invertible. Thus,

$$\boldsymbol{U}_t = \left(\widetilde{\boldsymbol{B}}_t - \frac{\lambda_1}{t}\boldsymbol{G}_t\right)\widetilde{\boldsymbol{A}}_t^{-1},$$

where $\widetilde{\boldsymbol{A}}_t^{-1}$ is the inverse of $\widetilde{\boldsymbol{A}}_t$.

Now we derive the bound for $\boldsymbol{U}_t$:

$$\begin{aligned}
\|\boldsymbol{U}_t\|_F &= \left\|\left(\widetilde{\boldsymbol{B}}_t - \frac{\lambda_1}{t}\boldsymbol{G}_t\right)\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F \\
&\leq \left\|\widetilde{\boldsymbol{B}}_t - \frac{\lambda_1}{t}\boldsymbol{G}_t\|_F \cdot \|\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F \\
&\leq \left(\left\|\widetilde{\boldsymbol{B}}_t\right\|_F + \frac{\lambda_1}{t}\|\boldsymbol{G}_t\|_F\right)\left\|\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F \\
&= \left\|\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F\left\|\widetilde{\boldsymbol{B}}_t\right\|_F + \frac{\lambda_1}{t}\left\|\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F\|\boldsymbol{G}_t\|_F \\
&\leq \left\|\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F\left\|\widetilde{\boldsymbol{B}}_t\right\|_F + \frac{\lambda_1}{t}\left\|\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F\|\boldsymbol{U}_t\|_F.
\end{aligned}$$

It follows that

$$\left(1 - \frac{\lambda_1}{t}\left\|\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F\right)\|\boldsymbol{U}_t\|_F \leq \left\|\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F\left\|\widetilde{\boldsymbol{B}}_t\right\|_F.$$

As all of the eigenvalues of $\widetilde{\boldsymbol{A}}_t$ are uniformly lower bounded, those of $\widetilde{\boldsymbol{A}}_t^{-1}$ are uniformly upper bounded. Thus the trace norm of $\widetilde{\boldsymbol{A}}_t^{-1}$ are uniformly upper bounded. As all norms are equivalent,

$\|\widetilde{\boldsymbol{A}}_t^{-1}\|_F$ is also uniformly upper bounded by a constant, say C_2. Thus,

$$\left(1-\frac{\lambda_1}{t}\mathrm{C}_2\right)\|\boldsymbol{U}_t\|_F \leq \left(1-\frac{\lambda_1}{t}\left\|\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F\right)\|\boldsymbol{U}_t\|_F \leq \left\|\widetilde{\boldsymbol{A}}_t^{-1}\right\|_F\left\|\widetilde{\boldsymbol{B}}_t\right\|_F \leq \mathrm{C}_2\mathrm{C}.$$

Particularly, let

$$t_0 = \min_t \{t \geq 2\lambda_1\mathrm{C}_2, t \text{ is an integer}\}.$$

Then, for all $t \geq t_0$,

$$\|\boldsymbol{U}_t\|_F \leq 2\mathrm{C}_2\mathrm{C}. \tag{4.34}$$

Step 2: Let us consider a uniform bound for $\boldsymbol{U}_t$, with $0 < t < t_0$. Recall that $\boldsymbol{U}_t$ is the minimizer for $g_t(\boldsymbol{U})$, that is

$$\begin{aligned}
\boldsymbol{U}_t &= \arg\min_{\boldsymbol{U}} g_t(\boldsymbol{U}) \\
&= \arg\min_{\boldsymbol{U}} \frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{2}\|\boldsymbol{z}_i - \boldsymbol{U}\boldsymbol{v}_i - \boldsymbol{e}_i\|^2 + \lambda_2\tilde{h}(\boldsymbol{e}_i)\right) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2 \\
&= \arg\min_{\boldsymbol{U}} \sum_{i=1}^{t}\frac{1}{2}\|\boldsymbol{z}_i - \boldsymbol{U}\boldsymbol{v}_i - \boldsymbol{e}_i\|^2 + \frac{\lambda_1}{2}\|\boldsymbol{U}\|_{2,\infty}^2 \\
&:= \arg\min_{\boldsymbol{U}} \tilde{g}_t(\boldsymbol{U}).
\end{aligned}$$

Consider a trivial but feasible solution with $\boldsymbol{U} = \boldsymbol{0}$,

$$\tilde{g}_t(\boldsymbol{0}) = \sum_{i=1}^{t}\frac{1}{2}\|\boldsymbol{z}_i - \boldsymbol{e}_i\|^2.$$

The inequality

$$\tilde{g}_t(\boldsymbol{U}_t) \leq \tilde{g}_t(\boldsymbol{0})$$

implies

$$\|\boldsymbol{U}_t\|_{2,\infty}^2 \leq \frac{1}{\lambda_1} \sum_{i=1}^{t} \|\boldsymbol{z}_i - \boldsymbol{e}_i\|^2 .$$

Since

$$\|\boldsymbol{U}_t\|_F^2 \leq d \, \|\boldsymbol{U}_t\|_{2,\infty}^2 \leq \frac{d}{\lambda_1} \sum_{i=1}^{t} \|\boldsymbol{z}_i - \boldsymbol{e}_i\|^2 ,$$

we have

$$\|\boldsymbol{U}_t\|_F \leq \sqrt{\frac{d}{\lambda_1} \sum_{i=1}^{t} \|\boldsymbol{z}_i - \boldsymbol{e}_i\|^2}.$$

For all $0 < t < t_0$,

$$\|\boldsymbol{U}_t\|_F \leq \sqrt{\frac{d}{\lambda_1} \sum_{i=1}^{t} \|\boldsymbol{z}_i - \boldsymbol{e}_i\|^2} \leq \sqrt{\frac{d}{\lambda_1} \sum_{i=1}^{t_0} \|\boldsymbol{z}_i - \boldsymbol{e}_i\|^2}. \tag{4.35}$$

Note that each term, particularly t_0, can be uniformly upper bounded, thus $\sqrt{\frac{d}{\lambda_1} \sum_{i=1}^{t_0} \|\boldsymbol{z}_i - \boldsymbol{e}_i\|^2}$ can also be uniformly upper bounded. Namely, for all $0 < t < t_0$, $\boldsymbol{U}_t$ is also uniformly upper bounded.

Step 3: Now let us define

$$U_{\max} = \max \left\{ 2\mathrm{C}_2\mathrm{C}, \sqrt{\frac{d}{\lambda_1} \sum_{i=1}^{t_0} \|\boldsymbol{z}_i - \boldsymbol{e}_i\|^2} \right\}.$$

Then, for all $t > 0$,

$$\|\boldsymbol{U}_t\|_F \leq U_{\max}.$$

All the constants, C_0, C_1, C_2 and C are independent from t, making them uniformly bounded. Also, t_0 is a constant that is uniform over t. Thus, $\boldsymbol{U}_t$ can be uniformly bounded. □

4.A.3 Proof of Corollary 4.5

Proof. The uniform bound of $\boldsymbol{v}_t$, $\boldsymbol{e}_t$ and $\boldsymbol{z}_t$, combined with the uniform bound of $\boldsymbol{U}_t$, implies the uniform boundedness for $\tilde{\ell}\left(\boldsymbol{z}_t, \boldsymbol{U}_t, \boldsymbol{v}_t, \boldsymbol{e}_t\right)$ and $\ell\left(\boldsymbol{z}_t, \boldsymbol{U}_t\right)$. Thus, $g_t(\boldsymbol{U}_t)$ and $f_t(\boldsymbol{U}_t)$ are also uniformly bounded.

To show that $g_t(\boldsymbol{U})$ is uniformly Lipschitz, we compute its subgradient at any $\boldsymbol{U} \in \mathcal{U}$:

$$\begin{aligned}\left\|\nabla_{\boldsymbol{U}} g_t(\boldsymbol{U})\right\|_F = \left\|\frac{1}{t}(\boldsymbol{U}\boldsymbol{A}_t - \boldsymbol{B}_t) + \frac{\lambda_1}{t}\boldsymbol{G}\right\|_F &\leq \left\|\frac{1}{t}(\boldsymbol{U}\boldsymbol{A}_t - \boldsymbol{B}_t)\right\|_F + \frac{\lambda_1}{t}\left\|\boldsymbol{U}\right\|_F \\ &\leq \left\|\frac{1}{t}(\boldsymbol{U}\boldsymbol{A}_t - \boldsymbol{B}_t)\right\|_F + \lambda_1\left\|\boldsymbol{U}\right\|_F\end{aligned}$$

where $\boldsymbol{G} \in \partial\frac{1}{2}\left\|\boldsymbol{U}\right\|_{2,\infty}$. Since $\boldsymbol{U}$, $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$ are all uniformly bounded, the subgradient of $g_t(\boldsymbol{U})$ is uniformly bounded. This implies that $g_t(\boldsymbol{U})$ is uniformly Lipschitz. □

Lemma 4.13 (A corollary of Donsker theorem [144])**.** *Let $F = \{f_\theta : \mathcal{X} \to \mathbb{R}, \theta \in \Theta\}$ be a set of measurable functions indexed by a bounded subset Θ of $\mathbb{R}^d$. Suppose that there exists a constant K such that*

$$\left|f_{\theta_1}(x) - f_{\theta_2}(x)\right| \leq K\left\|\theta_1 - \theta_2\right\|,$$

for every θ_1 and θ_2 in Θ and x in $\mathcal{X}$. Then, F is P-Donsker. For any f in F, let us define $\mathbb{P}_n f$, $\mathbb{P}f$ and $\mathbb{G}_n f$ as

$$\mathbb{P}_n f = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{X}_i), \quad \mathbb{P}f = \mathbb{E}[f(\boldsymbol{X})], \quad \mathbb{G}_n f = \sqrt{n}(\mathbb{P}_n f - \mathbb{P}f).$$

Let us also suppose that for all f, $\mathbb{P}f^2 < \delta^2$ and $\left\|f\right\|_\infty < M$ and that the random elements $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots$ are Borel-measurable. Then, we have

$$\mathbb{E}\left\|\mathbb{G}\right\|_F = \mathcal{O}\left(1\right)$$

where $\left\|\mathbb{G}\right\|_F = \sup_{f\in F}\left|\mathbb{G}_n f\right|$.

Now let us verify that the set of functions $\{\ell(\boldsymbol{z}, \boldsymbol{U}), \boldsymbol{U} \in \mathcal{U}\}$ indexed by $\boldsymbol{U}$ fulfills the hypotheses in the corollary of Donsker Theorem. In particular, we have verified that:

- The index set $\mathcal{U}$ is uniformly bounded (see Proposition 4.4).
- Each $\ell(\boldsymbol{z}, \boldsymbol{U})$ can be uniformly bounded (see Corollary 4.5).
- Any of the functions $\ell(\boldsymbol{z}, \boldsymbol{U})$ in the family is uniformly Lipschitz (see Proposition 4.6).

Next, we show that the family of functions $\ell(\boldsymbol{z}, \boldsymbol{U})$ is uniformly Lipschitz w.r.t. $\boldsymbol{U}$. We introduce the following lemma as it will be useful for our discussion.

Lemma 4.14 (Corollary of Theorem 4.1 from [21])**.** *Let $f : \mathbb{R}^d \times \mathbb{R}^r \to \mathbb{R}$. Suppose that for all $\boldsymbol{x} \in \mathbb{R}^d$ the function $f(\boldsymbol{x}, \cdot)$ is differentiable, and that f and $\nabla_{\boldsymbol{u}} f(\boldsymbol{x}, \boldsymbol{u})$ are continuous on $\mathbb{R}^d \times \mathbb{R}^r$. Let $\boldsymbol{v}(\boldsymbol{u})$ be the optimal value function $\boldsymbol{v}(\boldsymbol{u}) = \min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x}, \boldsymbol{u})$, where $\mathcal{C}$ is a compact subset of $\mathbb{R}^d$. Then $\boldsymbol{v}(\boldsymbol{u})$ is directionally differentiable. Furthermore, if for $\boldsymbol{u}_0 \in \mathbb{R}^r$, $f(\cdot, \boldsymbol{u}_0)$ has unique minimizer $\boldsymbol{x}_0$ then $\boldsymbol{v}(\boldsymbol{u})$ is differentiable in $\boldsymbol{u}_0$ and $\nabla_{\boldsymbol{u}} \boldsymbol{v}(\boldsymbol{u}_0) = \nabla_{\boldsymbol{u}} f(\boldsymbol{x}_0, \boldsymbol{u}_0)$.*

4.A.4 Proof of Proposition 4.6

Proof. By fixing the variable $\boldsymbol{z}$, the function $\tilde{\ell}$ can be seen as a mapping:

$$\mathbb{R}^{r+d} \times \mathcal{U} \to \mathbb{R}$$
$$([\boldsymbol{v};\ \boldsymbol{e}], \boldsymbol{U}) \mapsto \tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e}).$$

It is easy to show that $\forall\ [\boldsymbol{v};\ \boldsymbol{e}] \in \mathbb{R}^{r+d}$, $\tilde{\ell}(\boldsymbol{z}, \cdot, \boldsymbol{v}, \boldsymbol{e})$ is differentiable. Also $\tilde{\ell}(\boldsymbol{z}, \cdot, \cdot, \cdot)$ is continuous on $\mathbb{R}^{r+d} \times \mathcal{U}$. $\nabla_{\boldsymbol{U}} \tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e}) = (\boldsymbol{U}\boldsymbol{v} + \boldsymbol{e} - \boldsymbol{z})\boldsymbol{v}^\top$ is continuous on $\mathbb{R}^{r+d} \times \mathcal{U}$. $\forall \boldsymbol{U} \in \mathcal{U}$, according to Assumption $(A3)$, $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \cdot, \cdot)$ has a unique minimizer. Thus Lemma 4.14 applies and we prove that $\ell(\boldsymbol{z}, \boldsymbol{U})$ is differentiable in $\boldsymbol{U}$ and

$$\nabla_{\boldsymbol{U}} \ell(\boldsymbol{z}, \boldsymbol{U}) = (\boldsymbol{U}\boldsymbol{v}^* + \boldsymbol{e}^* - \boldsymbol{z}){\boldsymbol{v}^*}^\top.$$

Since every term in $\nabla_{\boldsymbol{U}} \ell(\boldsymbol{z}, \boldsymbol{U})$ is uniformly bounded (Assumption $(A1)$ and Proposition 4.4), we conclude that the gradient of $\ell(\boldsymbol{z}, \cdot)$ is uniformly bounded, implying that $\ell(\boldsymbol{z}, \boldsymbol{U})$ is uniformly Lipschitz w.r.t. $\boldsymbol{U}$. □

4.A.5 Proof of Corollary 4.7

Proof. As $\ell(\boldsymbol{z}, \boldsymbol{U})$ can be uniformly bounded (Corollary 4.5), we derive the uniform boundedness of $f_t(\boldsymbol{U})$. Let $\boldsymbol{G} \in \frac{1}{2}\partial \left\|\boldsymbol{U}\right\|_{2,\infty}$. By computing the subgradient of $f_t(\boldsymbol{U})$ at $\boldsymbol{U}$, we have

$$\begin{aligned}
\left\|\nabla f_t(\boldsymbol{U})\right\|_F &= \left\| \frac{1}{t}\sum_{i=1}^{t} \nabla_{\boldsymbol{U}} \ell(\boldsymbol{z}_i, \boldsymbol{U}) + \frac{\lambda_1}{t}\boldsymbol{G} \right\|_F \\
&\leq \frac{1}{t}\sum_{i=1}^{t} \left\| (\boldsymbol{U}\boldsymbol{v}_i + \boldsymbol{e}_i - \boldsymbol{z}_i)\boldsymbol{v}_i^\top \right\|_F + \frac{\lambda_1}{t}\left\|\boldsymbol{U}\right\|_F \\
&= \frac{1}{t}\sum_{i=1}^{t} \left\| \boldsymbol{U}\boldsymbol{v}_i\boldsymbol{v}_i^\top + (\boldsymbol{e}_i - \boldsymbol{z}_i)\boldsymbol{v}_i^\top \right\|_F + \frac{\lambda_1}{t}\left\|\boldsymbol{U}\right\|_F \\
&\leq \frac{1}{t}\sum_{i=1}^{t} \left(\left\|\boldsymbol{U}\right\|_F \cdot \left\| \boldsymbol{v}_i\boldsymbol{v}_i^\top \right\|_F + \left\| (\boldsymbol{e}_i - \boldsymbol{z}_i)\boldsymbol{v}_i^\top \right\|_F \right) + \frac{\lambda_1}{t}\left\|\boldsymbol{U}\right\|_F .
\end{aligned}$$

Note that all the terms (i.e. $\boldsymbol{z}_i$, $\boldsymbol{U}$, $\boldsymbol{v}_i$, $\boldsymbol{e}_i$) in the right hand inequality are uniformly bounded. Thus, we say that the subgradient of $f_t(\boldsymbol{U})$ is uniformly bounded and $f_t(\boldsymbol{U})$ is uniformly Lipschitz. □

4.A.6 Proof of Proposition 4.9

Proof. Based on Proposition 4.4 and Proposition 4.6, we argue that the set of measurable functions $\{\ell(\boldsymbol{z}, \boldsymbol{U}), \boldsymbol{U} \in \mathcal{U}\}$ is P-Donsker (defined in Lemma 4.13). From Corollary 4.5, we know that $\ell(\boldsymbol{z}, \boldsymbol{U})$ can be uniformly bounded by a constant, say C. Also note that from the definition of $\ell(\boldsymbol{z}, \boldsymbol{U})$ (see (4.12)), it is always non-negative. Thus, we have

$$\ell^2(\boldsymbol{z}, \boldsymbol{U}) \leq \mathrm{C}^2,$$

implying the uniform boundedness of $\mathbb{E}[\ell^2(\boldsymbol{z}, \boldsymbol{U})]$. Thus, Lemma 4.13 applies and we have

$$\mathbb{E}\big[\sqrt{t}\left\|(f_t - f)\right\|_\infty\big] = \mathcal{O}(1) .$$

The proof is complete. □

4.A.7 Proof of Theorem 4.10

We are ready to prove the convergence of $g_t(\boldsymbol{U}_t)$, which requires to justify that the stochastic process $\{g_t(\boldsymbol{U}_t)\}_{t=1}^{\infty}$ is a quasi-martingale, defined as follows:

Lemma 4.15 (Sufficient condition of convergence for a stochastic process [22])**.** *Let $(\Omega, \mathcal{F}, P)$ be a measurable probability space, u_t, for $t \geq 0$, be the realization of a stochastic process and $\mathcal{F}_t$ be the filtration by the past information at time t. Let*

$$\delta_t = \begin{cases} 1 & if\ \mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t] > 0, \\ 0 & otherwise. \end{cases}$$

If for all t, $u_t \geq 0$ and $\sum_{t=1}^{\infty} \mathbb{E}[\delta_t(u_{t+1} - u_t)] < \infty$, then u_t is a quasi-martingale and converges almost surely. Moreover,

$$\sum_{t=1}^{\infty} |\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]| < +\infty\ a.s.$$

Proof. For convenience, let us first define the stochastic positive process

$$u_t = g_t(\boldsymbol{U}_t) \geq 0.$$

We consider the difference between u_{t+1} and u_t:

$$\begin{aligned} u_{t+1} - u_t &= g_{t+1}(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_t) \\ &= g_{t+1}(\boldsymbol{U}_{t+1}) - g_{t+1}(\boldsymbol{U}_t) + g_{t+1}(\boldsymbol{U}_t) - g_t(\boldsymbol{U}_t) \\ &= g_{t+1}(\boldsymbol{U}_{t+1}) - g_{t+1}(\boldsymbol{U}_t) + \frac{1}{t+1}\ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_t) - \frac{1}{t+1}g_t(\boldsymbol{U}_t) \\ &= g_{t+1}(\boldsymbol{U}_{t+1}) - g_{t+1}(\boldsymbol{U}_t) + \frac{f_t(\boldsymbol{U}_t) - g_t(\boldsymbol{U}_t)}{t+1} + \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)}{t+1}. \end{aligned} \tag{4.36}$$

As $\boldsymbol{U}_{t+1}$ minimizes $g_{t+1}(\boldsymbol{U})$, we have

$$g_{t+1}(\boldsymbol{U}_{t+1}) - g_{t+1}(\boldsymbol{U}_t) \leq 0.$$

As $g_t(\boldsymbol{U}_t)$ is the surrogate function of $f_t(\boldsymbol{U}_t)$, we have

$$f_t(\boldsymbol{U}_t) - g_t(\boldsymbol{U}_t) \leq 0.$$

Thus,

$$u_{t+1} - u_t \leq \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)}{t+1}. \tag{4.37}$$

Let us consider the filtration of the past information $\mathcal{F}_t$ and take the expectation of (4.37) conditioning on $\mathcal{F}_t$:

$$\begin{aligned}
\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t] &\leq \frac{\mathbb{E}[\ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_t) \mid \mathcal{F}_t] - f_t(\boldsymbol{U}_t)}{t+1} \\
&\leq \frac{f(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)}{t+1} \\
&= \frac{f(\boldsymbol{U}_t) - f'_t(\boldsymbol{U}_t) - \frac{\lambda_1}{2t}\|\boldsymbol{U}_t\|_{2,\infty}^2}{t+1} \\
&\leq \frac{\|f - f'_t\|_\infty}{t+1} - \frac{\lambda_1}{2t(t+1)}\|\boldsymbol{U}_t\|_{2,\infty}^2 \\
&\leq \frac{\|f - f'_t\|_\infty}{t+1},
\end{aligned} \tag{4.38}$$

where

$$f'_t(\boldsymbol{U}) = \frac{1}{t}\sum_{i=1}^{t} \ell(\boldsymbol{z}_i, \boldsymbol{U}).$$

Note that

$$f'(\boldsymbol{U}) = \lim_{t\to\infty} f'_t(\boldsymbol{U}) = \mathbb{E}_{\boldsymbol{z}}[\ell(\boldsymbol{z}, \boldsymbol{U})] = f(\boldsymbol{U}).$$

From Proposition 4.9, we have

$$\mathbb{E}\Big[\, \Big\| \sqrt{t}(f'_t - f') \Big\|_\infty \Big] = \mathcal{O}\left(1\right).$$

Also note that according to Proposition 4.4, we have $\|\boldsymbol{U}_t\|_F \leq \boldsymbol{U}_{\max}$. Thus, considering the

positive part of $\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]$ in (4.38) and taking the expectation, we have

$$\mathbb{E}\big[\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]^+\big] = \mathbb{E}\big[\max\{\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t], 0\}\big] \leq \frac{\mathrm{C}}{\sqrt{t}(t+1)},$$

where C is a constant. Therefore, defining the set $\mathcal{T} = \{t \mid \mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t] > 0\}$ and

$$\delta_t = \begin{cases} 1 & \text{if } t \in \mathcal{T}, \\ 0 & \text{otherwise}, \end{cases}$$

we have

$$\begin{aligned} \sum_{t=1}^{\infty} \mathbb{E}[\delta_t(u_{t+1} - u_t)] &= \sum_{t \in \mathcal{T}} \mathbb{E}[(u_{t+1} - u_t)] \\ &= \sum_{t \in \mathcal{T}} \mathbb{E}[\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]] \\ &= \sum_{t=1}^{\infty} \mathbb{E}[\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]^+] \\ &< +\infty. \end{aligned}$$

According to Lemma 4.15, we conclude that $g_t(\boldsymbol{U}_t)$ is a quasi-martingale and converges almost surely. Moreover,

$$\sum_{t=1}^{\infty} |\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]| < +\infty \; a.s. \tag{4.39}$$

The proof is complete. □

4.A.8 Proof of Proposition 4.11

We now show that $g_t(\boldsymbol{U}_t)$ and $f(\boldsymbol{U}_t)$ converge to the same limit almost surely. Consequently, $f(\boldsymbol{U}_t)$ converges almost surely. First, we prove that $b_t := g_t(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)$ converges to 0 almost surely. We utilize the lemma from [99] for the proof.

Lemma 4.16 (Lemma 8 from [99])**.** *Let a_t, b_t be two real sequences such that for all t, $a_t \geq 0$, $b_t \geq 0$, $\sum_{t=1}^{\infty} a_t = \infty$, $\sum_{t=1}^{\infty} a_t b_t < \infty$, $\exists K > 0$, such that $|b_{t+1} - b_t| < K a_t$. Then,* $\lim_{t \to +\infty} b_t = 0$.

We notice that another sequence $\{a_t\}_{t=1}^{\infty}$ should be constructed in Lemma 4.16. Here, we

take the $a_t = \frac{1}{t} \geq 0$, which satisfies the condition $\sum_{t=1}^{\infty} a_t = \infty$. Next, we need to show that $|b_{t+1} - b_t| < Ka_t$, where K is a constant. To do this, we alternatively show that $|b_{t+1} - b_t|$ can be upper bounded by $\|\boldsymbol{U}_{t+1} - \boldsymbol{U}_t\|_F$, which can be further bounded by Ka_t.

Proof. Let us define

$$\hat{g}_t(\boldsymbol{U}) = \frac{1}{t}\left(\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{A}_t\right) - \mathrm{Tr}\left(\boldsymbol{U}^\top \boldsymbol{B}_t\right)\right) + \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2 .$$

According the strong convexity of $\boldsymbol{A}_t$ (Assumption $(A2)$), and the convexity of $\|\boldsymbol{U}\|_{2,\infty}^2$, we can derive the strong convexity of $\hat{g}_t(\boldsymbol{U})$. That is,

$$\hat{g}_t(\boldsymbol{U}_{t+1}) - \hat{g}_t(\boldsymbol{U}_t) \geq \langle \boldsymbol{G}_t, \boldsymbol{U}_{t+1} - \boldsymbol{U}_t \rangle + \frac{\beta_1}{2}\|\boldsymbol{U}_{t+1} - \boldsymbol{U}_t\|_F^2 ,$$

where $\boldsymbol{G}_t \in \partial \hat{g}_t(\boldsymbol{U}_t)$. As $\boldsymbol{U}_t$ is the minimizer of $\hat{g}_t$, we have

$$\mathbf{0} \in \partial \hat{g}_t(\boldsymbol{U}_t).$$

Let $\boldsymbol{G}_t$ be the zero matrix. Then we have

$$\hat{g}_t(\boldsymbol{U}_{t+1}) - \hat{g}_t(\boldsymbol{U}_t) \geq \frac{\beta_1}{2}\|\boldsymbol{U}_{t+1} - \boldsymbol{U}_t\|_F^2 . \tag{4.40}$$

On the other hand,

$$\begin{aligned}\hat{g}_t(\boldsymbol{U}_{t+1}) - \hat{g}_t(\boldsymbol{U}_t) =& \hat{g}_t(\boldsymbol{U}_{t+1}) - \hat{g}_{t+1}(\boldsymbol{U}_{t+1}) + \hat{g}_{t+1}(\boldsymbol{U}_{t+1}) - \hat{g}_{t+1}(\boldsymbol{U}_t) + \hat{g}_{t+1}(\boldsymbol{U}_t) - \hat{g}_t(\boldsymbol{U}_t) \\ \leq& \hat{g}_t(\boldsymbol{U}_{t+1}) - \hat{g}_{t+1}(\boldsymbol{U}_{t+1}) + \hat{g}_{t+1}(\boldsymbol{U}_t) - \hat{g}_t(\boldsymbol{U}_t). \end{aligned} \tag{4.41}$$

Note that the inequality is derived by the fact that $\hat{g}_{t+1}(\boldsymbol{U}_{t+1}) - \hat{g}_{t+1}(\boldsymbol{U}_t) \leq 0$, as $\boldsymbol{U}_{t+1}$ is the minimizer of $\hat{g}_{t+1}(\boldsymbol{U})$. Let us denote $\hat{g}_t(\boldsymbol{U}) - \hat{g}_{t+1}(\boldsymbol{U})$ by $\delta_t(\boldsymbol{U})$. We have

$$\begin{aligned}\delta_t(\boldsymbol{U}) = &\frac{1}{t}\left(\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{A}_t\right) - \mathrm{Tr}\left(\boldsymbol{U}^\top \boldsymbol{B}_t\right)\right) - \frac{1}{t+1}\left(\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{A}_{t+1}\right) - \mathrm{Tr}\left(\boldsymbol{U}^\top \boldsymbol{B}_{t+1}\right)\right) \\ &+ \frac{\lambda_1}{2t}\|\boldsymbol{U}\|_{2,\infty}^2 - \frac{\lambda_1}{2(t+1)}\|\boldsymbol{U}\|_{2,\infty}^2 .\end{aligned}$$

By a simple calculation, we have the gradient of $\delta_t(\boldsymbol{U})$:

$$\begin{aligned}\nabla\delta_t(\boldsymbol{U}) &= \frac{1}{t}\left(\boldsymbol{U}\boldsymbol{A}_t - \boldsymbol{B}_t\right) - \frac{1}{t+1}\left(\boldsymbol{U}\boldsymbol{A}_{t+1} - \boldsymbol{B}_{t+1}\right) + \left(\frac{1}{t} - \frac{1}{t+1}\right)\lambda_1\boldsymbol{G} \\ &= \frac{1}{t}\left(\boldsymbol{U}(\boldsymbol{A}_t - \frac{t}{t+1}\boldsymbol{A}_{t+1}) + \frac{t}{t+1}\boldsymbol{B}_{t+1} - \boldsymbol{B}_t + \frac{\lambda_1}{t+1}\boldsymbol{G}\right),\end{aligned}$$

where $\boldsymbol{G} \in \partial\left\|\boldsymbol{U}\right\|_{2,\infty}^2$. We then compute the Frobenius norm of the gradient of $\delta_t(\boldsymbol{U})$:

$$\begin{aligned}\left\|\nabla\delta_t(\boldsymbol{U})\right\|_F &\leq \frac{1}{t}\left(\left\|\boldsymbol{U}(\boldsymbol{A}_t - \frac{t}{t+1}\boldsymbol{A}_{t+1})\right\|_F + \left\|\frac{t}{t+1}\boldsymbol{B}_{t+1} - \boldsymbol{B}_t\right\|_F + \frac{\lambda_1}{t+1}\left\|\boldsymbol{U}\right\|_F\right) \\ &\leq \frac{1}{t}\left(\left\|\boldsymbol{U}\right\|_F \cdot \left\|\boldsymbol{A}_t - \frac{t}{t+1}\boldsymbol{A}_{t+1}\right\|_F + \left\|\frac{t}{t+1}\boldsymbol{B}_{t+1} - \boldsymbol{B}_t\right\|_F + \frac{\lambda_1}{t+1}\left\|\boldsymbol{U}\right\|_F\right) \\ &= \frac{1}{t}\Big(\left\|\boldsymbol{U}\right\|_F \cdot \left\|\frac{1}{t+1}\boldsymbol{A}_t - \frac{t}{t+1}\boldsymbol{v}_{t+1}\boldsymbol{v}_{t+1}^\top\right\|_F \\ &\quad + \left\|\frac{1}{t+1}\boldsymbol{B}_t - \frac{t}{t+1}\left(\boldsymbol{z}_{t+1} - \boldsymbol{e}_{t+1}\right)\boldsymbol{v}_{t+1}^\top\right\|_F + \frac{\lambda_1}{t+1}\left\|\boldsymbol{U}\right\|_F\Big). \qquad (4.42)\end{aligned}$$

According to the first order Taylor expansion,

$$\begin{aligned}\delta_t(\boldsymbol{U}_{t+1}) - \delta_t(\boldsymbol{U}_t) &= \mathrm{Tr}\left(\left(\boldsymbol{U}_{t+1} - \boldsymbol{U}_t\right)^\top \nabla\delta_t\left(\alpha\boldsymbol{U}_t + (1-\alpha)\boldsymbol{U}_{t+1}\right)\right) \\ &\leq \left\|\boldsymbol{U}_{t+1} - \boldsymbol{U}_t\right\|_F \cdot \left\|\nabla\delta_t\left(\alpha\boldsymbol{U}_t + (1-\alpha)\boldsymbol{U}_{t+1}\right)\right\|_F,\end{aligned}$$

where α is a constant between 0 and 1. According to Proposition 4.4, $\boldsymbol{U}_t$ and $\boldsymbol{U}_{t+1}$ are uniformly bounded, so $\alpha\boldsymbol{U}_t + (1-\alpha)\boldsymbol{U}_{t+1}$ is uniformly bounded. According to Proposition 4.4, $\frac{1}{t+1}\boldsymbol{A}_t$, $\frac{t}{t+1}\boldsymbol{v}_{t+1}\boldsymbol{v}_{t+1}^\top$, $\frac{1}{t+1}\boldsymbol{B}_t$ and $\frac{t}{t+1}\left(\boldsymbol{z}_{t+1} - \boldsymbol{e}_{t+1}\right)\boldsymbol{v}_{t+1}^\top$ are all uniformly bounded. Thus, there exists a constant C_1, such that

$$\left\|\nabla\delta_t\left(\alpha\boldsymbol{U}_t + (1-\alpha)\boldsymbol{U}_{t+1}\right)\right\|_F \leq \frac{\mathrm{C}_1}{t},$$

resulting that

$$\delta_t(\boldsymbol{U}_{t+1}) - \delta_t(\boldsymbol{U}_t) \leq \frac{\mathrm{C}_1}{t}\left\|\boldsymbol{U}_{t+1} - \boldsymbol{U}_t\right\|_F.$$

Applying this property in (4.41), we have

$$\hat{g}_t(\boldsymbol{U}_{t+1}) - \hat{g}_t(\boldsymbol{U}_t) \leq \delta_t(\boldsymbol{U}_{t+1}) - \delta_t(\boldsymbol{U}_t) \leq \frac{\mathrm{C}_1}{t} \left\| \boldsymbol{U}_{t+1} - \boldsymbol{U}_t \right\|_F . \tag{4.43}$$

From (4.40) and (4.43), we conclude that

$$\left\| \boldsymbol{U}_{t+1} - \boldsymbol{U}_t \right\|_F \leq \frac{2\mathrm{C}_1}{\beta_1} \cdot \frac{1}{t},$$

which completes the proof. □

4.A.9 Proof of Theorem 4.12

Proof. We start our proof by deriving an upper bound for $g_t(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)$.

Step 1: According to (4.36),

$$\begin{aligned}
\frac{b_t}{t+1} &= g_{t+1}(\boldsymbol{U}_{t+1}) - g_{t+1}(\boldsymbol{U}_t) + \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)}{t+1} + u_t - u_{t+1} \\
&\leq \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)}{t+1} + u_t - u_{t+1}.
\end{aligned}$$

Taking the expectation conditioning on the past information $\mathcal{F}_t$ in the above equation, and noting that

$$\begin{aligned}
\mathbb{E}\Big[\frac{b_t}{t+1} \mid \mathcal{F}_t\Big] &= \frac{g_t(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)}{t+1}, \\
\mathbb{E}\Big[\frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)}{t+1} \mid \mathcal{F}_t\Big] &= \frac{f(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)}{t+1},
\end{aligned}$$

we have

$$\frac{b_t}{t+1} \leq \frac{f(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)}{t+1} + \mathbb{E}[u_t - u_{t+1} \mid \mathcal{F}_t].$$

Thus,

$$\sum_{t=1}^{\infty} \frac{b_t}{t+1} \leq \sum_{t=1}^{\infty} \frac{f(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)}{t+1} + \sum_{t=1}^{\infty} \mathbb{E}[u_t - u_{t+1} \mid \mathcal{F}_t].$$

According to the central limit theorem, $\sqrt{t}(f(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t))$ is bounded almost surely. Also, from (4.39),

$$\sum_{t=1}^{\infty} \mathbb{E}[u_t - u_{t+1} \mid \mathcal{F}_t] \leq \sum_{t=1}^{\infty} |\mathbb{E}[u_t - u_{t+1} \mid \mathcal{F}_t]| < +\infty.$$

Thus,

$$\sum_{t=1}^{\infty} \frac{b_t}{t+1} < +\infty.$$

Step 2: We examine the difference between b_{t+1} and b_t:

$$\begin{aligned}
|b_{t+1} - b_t| = & \, |g_{t+1}(\boldsymbol{U}_{t+1}) - f_{t+1}(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_t) + f_t(\boldsymbol{U}_t)| \\
\leq & \, |g_{t+1}(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_t)| + |f_{t+1}(\boldsymbol{U}_{t+1}) - f_t(\boldsymbol{U}_t)| \\
= & \, |g_{t+1}(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_{t+1}) + g_t(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_t)| \\
& + |f_{t+1}(\boldsymbol{U}_{t+1}) - f_t(\boldsymbol{U}_{t+1}) + f_t(\boldsymbol{U}_{t+1}) - f_t(\boldsymbol{U}_t)| \\
\leq & \, |g_{t+1}(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_{t+1})| + |g_t(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_t)| \\
& + |f_{t+1}(\boldsymbol{U}_{t+1}) - f_t(\boldsymbol{U}_{t+1})| + |f_t(\boldsymbol{U}_{t+1}) - f_t(\boldsymbol{U}_t)| \\
= & \left| \frac{1}{t+1} \ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_{t+1}) - \frac{1}{t+1} g_t(\boldsymbol{U}_{t+1}) \right| + |g_t(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_t)| \\
& + \left| \frac{1}{t+1} \ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_{t+1}) - \frac{1}{t+1} f_t(\boldsymbol{U}_{t+1}) \right| + |f_t(\boldsymbol{U}_{t+1}) - f_t(\boldsymbol{U}_t)| \,.
\end{aligned}$$

According to Corollary 4.5 and Corollary 4.7, we know that there exist constant C_1 and C_2 that are uniformly over t, such that

$$\begin{aligned}
|g_t(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_t)| \leq \mathrm{C}_1 \left\| \boldsymbol{U}_{t+1} - \boldsymbol{U}_t \right\|_F , \\
|f_t(\boldsymbol{U}_{t+1}) - f_t(\boldsymbol{U}_t)| \leq \mathrm{C}_2 \left\| \boldsymbol{U}_{t+1} - \boldsymbol{U}_t \right\|_F .
\end{aligned}$$

Combing with Proposition 4.11, there exists a constant C_3 that is uniformly over t, such that

$$|g_t(\boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_t)| + |f_t(\boldsymbol{U}_{t+1}) - f_t(\boldsymbol{U}_t)| \leq \frac{\mathrm{C}_3}{t}.$$

As we shown, $\ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_{t+1})$, $g_t(\boldsymbol{U}_{t+1})$ and $f_t(\boldsymbol{U}_{t+1})$ are all uniformly bounded. Therefore, there exists a constant C_4, such that

$$|\ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_{t+1}) - g_t(\boldsymbol{U}_{t+1})| + |\ell(\boldsymbol{z}_{t+1}, \boldsymbol{U}_{t+1}) - f_t(\boldsymbol{U}_t + 1)| \leq \mathrm{C}_4.$$

Finally, we have

$$b_{t+1} - b_t \leq \frac{\mathrm{C}_4}{t+1} + \frac{\mathrm{C}_3}{t} \leq \frac{\mathrm{C}_5}{t},$$

where C_5 is a constant that is uniformly over t.

Applying Lemma 4.16, we conclude that $\{b_t\}$ converges to zero. That is,

$$\lim_{t \to +\infty} g_t(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t) = 0. \tag{4.44}$$

In Theorem 4.10, we have shown that $g_t(\boldsymbol{U}_t)$ converges almost surely. This implies that $f_t(\boldsymbol{U}_t)$ also converges almost surely to the same limit of $g_t(\boldsymbol{U}_t)$.

According to the central limit theorem, $\sqrt{t}(f(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t)$ is bounded, implying

$$\lim_{t \to +\infty} f(\boldsymbol{U}_t) - f_t(\boldsymbol{U}_t) = 0, \quad a.s.$$

Thus, we conclude that $f(\boldsymbol{U}_t)$ converges almost surely to the same limit of $f_t(\boldsymbol{U}_t)$ (or, $g_t(\boldsymbol{U}_t)$). □

4.A.10 Proof of Theorem 4.3

According to Theorem 4.12, we can see that $g_t(\boldsymbol{U}_t)$ and $f(\boldsymbol{U}_t)$ converge to the same limit almost surely. Let t tends to infinity, as $\boldsymbol{U}_t$ is uniformly bounded (Proposition 4.4), the term $\frac{\lambda_1}{2t} \|\boldsymbol{U}_t\|_{2,\infty}^2$ in $g_t(\boldsymbol{U}_t)$ vanishes. Thus $g_t(\boldsymbol{U}_t)$ becomes differentiable. On the other hand, we have the following proposition about the gradient of $f(\boldsymbol{U})$.

Proposition 4.17 (Subgradient of $f(\boldsymbol{U})$)**.** *Let $f(\boldsymbol{U})$ be the expected loss function defined in* (4.25)*. Then, $f(\boldsymbol{U})$ is continuously differentiable and $\nabla f(\boldsymbol{U}) = \mathbb{E}_{\boldsymbol{z}}[\nabla_{\boldsymbol{U}} \ell(\boldsymbol{z}, \boldsymbol{U})]$. Moreover, $\nabla f(\boldsymbol{U})$ is uniformly Lipschitz on $\mathcal{U}$.*

Proof. Since $\ell(\boldsymbol{z}, \boldsymbol{U})$ is continuously differentiable (Proposition 4.6), $f(\boldsymbol{U})$ is continuously differentiable and $\nabla f(\boldsymbol{U}) = \mathbb{E}_{\boldsymbol{z}}[\nabla_{\boldsymbol{U}} \ell(\boldsymbol{z}, \boldsymbol{U})]$.

Now we prove the second claim. Let us consider a matrix $\boldsymbol{U}$ and a sample $\boldsymbol{z}$, and denote $\boldsymbol{v}^*(\boldsymbol{z}, \boldsymbol{U})$ and $\boldsymbol{e}^*(\boldsymbol{z}, \boldsymbol{U})$ as the optimal solutions for (4.12).

Step 1: First, $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{v}, \boldsymbol{e})$ is continuous in $\boldsymbol{z}$, $\boldsymbol{U}$, $\boldsymbol{v}$ and $\boldsymbol{e}$, and has a unique minimizer. This implies that $\boldsymbol{v}^*(\boldsymbol{z}, \boldsymbol{U})$ and $\boldsymbol{e}^*(\boldsymbol{z}, \boldsymbol{U})$ is continuous in $\boldsymbol{z}$ and $\boldsymbol{U}$.

Let us denote $\mathit{\Lambda}$ as the set of the indices such that $\forall j \in \mathit{\Lambda}$, $\boldsymbol{e}_j^* \neq 0$. According to the first order optimal condition for (4.14) w.r.t $\boldsymbol{e}$, we have

$$\boldsymbol{z} - \boldsymbol{U}\boldsymbol{v} - \boldsymbol{e} \in \lambda_2 \partial \left\| \boldsymbol{e} \right\|_1 ,$$

implying

$$|(\boldsymbol{z} - \boldsymbol{U}\boldsymbol{v} - \boldsymbol{e})_j| = \lambda_2, \ \forall j \in \mathit{\Lambda}.$$

Since $\boldsymbol{z} - \boldsymbol{U}\boldsymbol{v} - \boldsymbol{e}$ is continuous in $\boldsymbol{z}$ and $\boldsymbol{U}$, we consider a small perturbation of $(\boldsymbol{z}, \boldsymbol{U})$ in one of their open neighborhood V, such that for all $(\boldsymbol{z}', \boldsymbol{U}') \in V$, we have if $j \notin \mathit{\Lambda}$, then $\left|(\boldsymbol{z}' - \boldsymbol{U}'\boldsymbol{v}^{*\prime} - \boldsymbol{e}^{*\prime})_j\right| < \lambda_2$ and $\boldsymbol{e}^{*\prime}_j = 0$, where $\boldsymbol{v}^{*\prime} = \boldsymbol{v}^*(\boldsymbol{z}', \boldsymbol{U}')$ and $\boldsymbol{e}^{*\prime} = \boldsymbol{e}^*(\boldsymbol{z}', \boldsymbol{U}')$. That is, the support set of $\boldsymbol{e}^*$ does not change.

Let us denote $\boldsymbol{D} = [\boldsymbol{U} \ \boldsymbol{I}]$ and $\boldsymbol{b} = [\boldsymbol{v}; \ \boldsymbol{e}]$ and consider the function

$$\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}_{\mathit{\Lambda}}, \boldsymbol{b}_{\mathit{\Lambda}}) := \frac{1}{2} \left\| \boldsymbol{z} - \boldsymbol{D}_{\mathit{\Lambda}} \boldsymbol{b}_{\mathit{\Lambda}} \right\|^2 + \lambda_2 \left\| [\boldsymbol{0} \ \boldsymbol{I}] \boldsymbol{b}_{\mathit{\Lambda}} \right\|_1 .$$

According to Assumption $(A3)$, $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}_{\mathit{\Lambda}}, \cdot)$ is strongly convex with a Hessian lower-bounded by a positive constant C_1. Thus,

$$\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}_{\mathit{\Lambda}}, \boldsymbol{b}'^*_{\mathit{\Lambda}}) - \tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}_{\mathit{\Lambda}}, \boldsymbol{b}^*_{\mathit{\Lambda}}) \geq \mathrm{C}_1 \left\| \boldsymbol{b}_{\mathit{\Lambda}} - \boldsymbol{b}'_{\mathit{\Lambda}} \right\|^2 = \mathrm{C}_1 \left(\left\| \boldsymbol{v}^* - \boldsymbol{v}'^* \right\|^2 + \left\| \boldsymbol{e}^*_{\mathit{\Lambda}} - \boldsymbol{e}'^*_{\mathit{\Lambda}} \right\|^2 \right). \quad (4.45)$$

Step 2: We shall prove that $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \cdot) - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}', \cdot)$ is Lipschitz w.r.t. $\boldsymbol{b}$.

$$
\begin{aligned}
& 2\left(\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{b}) - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}', \boldsymbol{b})\right) - 2\left(\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{b}') - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}', \boldsymbol{b}')\right) \\
&= \|\boldsymbol{z} - \boldsymbol{D}\boldsymbol{b}\|_2^2 - \|\boldsymbol{z} - \boldsymbol{D}\boldsymbol{b}'\|_2^2 + \|\boldsymbol{z}' - \boldsymbol{D}'\boldsymbol{b}'\|_2^2 - \|\boldsymbol{z}' - \boldsymbol{D}'\boldsymbol{b}\|_2^2 \\
&= 2\boldsymbol{z}^\top \boldsymbol{D}(\boldsymbol{b}' - \boldsymbol{b}) + \boldsymbol{b}^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{b} - \boldsymbol{b}'^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{b}' - 2\boldsymbol{z}'^\top \boldsymbol{D}'(\boldsymbol{b}' - \boldsymbol{b}) - \boldsymbol{b}^\top \boldsymbol{D}'^\top \boldsymbol{D}'\boldsymbol{b} + \boldsymbol{b}'^\top \boldsymbol{D}'^\top \boldsymbol{D}'\boldsymbol{b}' \\
&= 2\left[(\boldsymbol{z}^\top \boldsymbol{D} - \boldsymbol{z}'^\top \boldsymbol{D}')(\boldsymbol{b}' - \boldsymbol{b})\right] + \left[\boldsymbol{b}^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{b} - \boldsymbol{b}^\top \boldsymbol{D}'^\top \boldsymbol{D}'\boldsymbol{b} + \boldsymbol{b}'^\top \boldsymbol{D}'^\top \boldsymbol{D}'\boldsymbol{b}' - \boldsymbol{b}'^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{b}'\right].
\end{aligned}
$$

For the first term,

$$
\begin{aligned}
(\boldsymbol{z}^\top \boldsymbol{D} - \boldsymbol{z}'^\top \boldsymbol{D}')(\boldsymbol{b}' - \boldsymbol{b}) =& (\boldsymbol{z}^\top \boldsymbol{D} - \boldsymbol{z}^\top \boldsymbol{D}' + \boldsymbol{z}^\top \boldsymbol{D}' - \boldsymbol{z}'^\top \boldsymbol{D}'^\top)(\boldsymbol{b}' - \boldsymbol{b}) \\
=& \left(\boldsymbol{z}^\top (\boldsymbol{D} - \boldsymbol{D}') + (\boldsymbol{z}^\top - \boldsymbol{z}'^\top)\boldsymbol{D}'\right)(\boldsymbol{b}' - \boldsymbol{b}).
\end{aligned}
$$

As each sample is bounded, $\boldsymbol{D}$ is bounded (as $\boldsymbol{U}$ is bounded), so the ℓ_2 norm of the first term can be bounded as follows:

$$
\begin{aligned}
& \left\|(\boldsymbol{z}^\top \boldsymbol{D} - \boldsymbol{z}'^\top \boldsymbol{D}')(\boldsymbol{b}' - \boldsymbol{b})\right\| \\
=& \left\|\left(\boldsymbol{z}^\top (\boldsymbol{D} - \boldsymbol{D}') + (\boldsymbol{z}^\top - \boldsymbol{z}'^\top)\boldsymbol{D}'\right)(\boldsymbol{b}' - \boldsymbol{b})\right\| \\
\leq& \left(\|\boldsymbol{z}\|_2 \|\boldsymbol{D} - \boldsymbol{D}'\|_F + \|\boldsymbol{z} - \boldsymbol{z}'\|_2 \|\boldsymbol{D}'\|_F\right) \cdot \|\boldsymbol{b}' - \boldsymbol{b}\| \\
\leq& \left(\mathrm{C}_1 \|\boldsymbol{D} - \boldsymbol{D}'\|_F + \mathrm{C}_2 \|\boldsymbol{z} - \boldsymbol{z}'\|_2\right) \cdot \|\boldsymbol{b}' - \boldsymbol{b}\|. \qquad (4.46)
\end{aligned}
$$

For the second term, we have

$$
\begin{aligned}
&\boldsymbol{b}^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{b} - \boldsymbol{b}^\top \boldsymbol{D}'^\top \boldsymbol{D}'\boldsymbol{b} + \boldsymbol{b}'^\top \boldsymbol{D}'^\top \boldsymbol{D}'\boldsymbol{b}' - \boldsymbol{b}'^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{b}' \\
&= \boldsymbol{b}^\top \left(\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{D}'^\top \boldsymbol{D}'\right) \boldsymbol{b} - \boldsymbol{b}'^\top \left(\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{D}'^\top \boldsymbol{D}'\right) \boldsymbol{b}' \\
&= \boldsymbol{b}^\top \left(\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{D}'^\top \boldsymbol{D}'\right) \boldsymbol{b} - \boldsymbol{b}^\top \left(\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{D}'^\top \boldsymbol{D}'\right) \boldsymbol{b}' + \boldsymbol{b}^\top \left(\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{D}'^\top \boldsymbol{D}'\right) \boldsymbol{b}' \\
&\quad - \boldsymbol{b}'^\top \left(\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{D}'^\top \boldsymbol{D}'\right) \boldsymbol{b}' \\
&= \boldsymbol{b}^\top \left(\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{D}'^\top \boldsymbol{D}'\right) (\boldsymbol{b} - \boldsymbol{b}') + (\boldsymbol{b} - \boldsymbol{b}')^\top \left(\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{D}'^\top \boldsymbol{D}'\right) \boldsymbol{b}' \\
&= \boldsymbol{b}^\top \left(\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{D}^\top \boldsymbol{D}' + \boldsymbol{D}^\top \boldsymbol{D}' - \boldsymbol{D}'^\top \boldsymbol{D}'\right) (\boldsymbol{b} - \boldsymbol{b}') \\
&\quad + (\boldsymbol{b} - \boldsymbol{b}')^\top \left(\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{D}^\top \boldsymbol{D}' + \boldsymbol{D}^\top \boldsymbol{D}' - \boldsymbol{D}'^\top \boldsymbol{D}'\right) \boldsymbol{b}' \\
&= \boldsymbol{b}^\top \left(\boldsymbol{D}^\top (\boldsymbol{D} - \boldsymbol{D}') + \left(\boldsymbol{D}^\top - \boldsymbol{D}'\right) \boldsymbol{D}'\right) (\boldsymbol{b} - \boldsymbol{b}') \\
&\quad + (\boldsymbol{b} - \boldsymbol{b}')^\top \left(\boldsymbol{D}^\top (\boldsymbol{D} - \boldsymbol{D}') + \left(\boldsymbol{D}^\top - \boldsymbol{D}'\right) \boldsymbol{D}'\right) \boldsymbol{b}'.
\end{aligned}
$$

Since $\boldsymbol{D}$ is bounded and $\boldsymbol{b}$ is bounded, the second term can be bounded as follows:

$$
\begin{aligned}
&\left\| \boldsymbol{b}^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{b} - \boldsymbol{b}^\top \boldsymbol{D}'^\top \boldsymbol{D}'\boldsymbol{b} + \boldsymbol{b}'^\top \boldsymbol{D}'^\top \boldsymbol{D}'\boldsymbol{b}' - \boldsymbol{b}'^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{b}' \right\| \\
&= \left\| \boldsymbol{b}^\top \left(\boldsymbol{D}^\top (\boldsymbol{D} - \boldsymbol{D}') + \left(\boldsymbol{D}^\top - \boldsymbol{D}'^\top\right) \boldsymbol{D}'\right) (\boldsymbol{b} - \boldsymbol{b}') \right. \\
&\quad \left. + (\boldsymbol{b} - \boldsymbol{b}')^\top \left(\boldsymbol{D}^\top (\boldsymbol{D} - \boldsymbol{D}') + \left(\boldsymbol{D}^\top - \boldsymbol{D}'^\top\right) \boldsymbol{D}'\right) \boldsymbol{b}' \right\| \\
&\leq \mathrm{C}_3 \left\| \boldsymbol{D} - \boldsymbol{D}' \right\|_F \cdot \left\| \boldsymbol{b} - \boldsymbol{b}' \right\|. \qquad (4.47)
\end{aligned}
$$

Combining (4.46) and (4.47), we prove that the function $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \cdot) - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}', \cdot)$ is Lipschitz:

$$
\begin{aligned}
&\left(\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{b}) - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}', \boldsymbol{b})\right) - \left(\tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}, \boldsymbol{b}') - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}', \boldsymbol{b}')\right) \\
\leq\ & \left((\mathrm{C}_1 + \mathrm{C}_3) \left\| \boldsymbol{D} - \boldsymbol{D}' \right\|_F + \mathrm{C}_2 \left\| \boldsymbol{z} - \boldsymbol{z}' \right\|\right) \left\| \boldsymbol{b} - \boldsymbol{b}' \right\| \\
=\ & \left((\mathrm{C}_1 + \mathrm{C}_3) \left\| \boldsymbol{D} - \boldsymbol{D}' \right\|_F + \mathrm{C}_2 \left\| \boldsymbol{z} - \boldsymbol{z}' \right\|\right) \sqrt{\|\boldsymbol{v} - \boldsymbol{v}'\|^2 + \|\boldsymbol{e} - \boldsymbol{e}'\|^2}. \qquad (4.48)
\end{aligned}
$$

Step 3: According to (4.45) and (4.48), and the fact that $\boldsymbol{b}'^*$ minimizes $\tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}', \cdot)$, we have

$$
\begin{aligned}
&\mathrm{C}_1 \left(\left\| \boldsymbol{v}^* - \boldsymbol{v}'^* \right\|^2 + \left\| \boldsymbol{e}_\Lambda^* - \boldsymbol{e}_\Lambda'^* \right\|^2 \right) \\
\leq\ & \tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}_\Lambda, \boldsymbol{b}_\Lambda'^*) - \tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}_\Lambda, \boldsymbol{b}_\Lambda^*) \\
=\ & \tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}_\Lambda, \boldsymbol{b}_\Lambda'^*) - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}'_\Lambda, \boldsymbol{b}_\Lambda^*) + \tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}'_\Lambda, \boldsymbol{b}_\Lambda^*) - \tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}_\Lambda, \boldsymbol{b}_\Lambda^*) \\
\leq\ & \tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}_\Lambda, \boldsymbol{b}_\Lambda'^*) - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}'_\Lambda, \boldsymbol{b}_\Lambda'^*) + \tilde{\ell}(\boldsymbol{z}', \boldsymbol{U}'_\Lambda, \boldsymbol{b}_\Lambda^*) - \tilde{\ell}(\boldsymbol{z}, \boldsymbol{U}_\Lambda, \boldsymbol{b}_\Lambda^*) \\
\leq\ & \left((\mathrm{C}_1 + \mathrm{C}_3) \left\| \boldsymbol{D} - \boldsymbol{D}' \right\|_F + \mathrm{C}_2 \left\| \boldsymbol{z} - \boldsymbol{z}' \right\| \right) \sqrt{\left\| \boldsymbol{v}^* - \boldsymbol{v}'^* \right\|^2 + \left\| \boldsymbol{e}_\Lambda^* - \boldsymbol{e}_\Lambda'^* \right\|^2}.
\end{aligned}
$$

Therefore, $\boldsymbol{v}^*(\boldsymbol{z}, \boldsymbol{U})$ and $\boldsymbol{e}^*(\boldsymbol{z}, \boldsymbol{U})$ are Lipschitz, which concludes the proof. □

Finally, taking a first order Taylor expansion for $f(\boldsymbol{U}_t)$ and $g_t(\boldsymbol{U}_t)$, we can show that the gradient of $f(\boldsymbol{U}_t)$ equals to that of $g_t(\boldsymbol{U}_t)$ when t tends to infinity. Since $\boldsymbol{U}_t$ is the minimizer for $g_t(\boldsymbol{U})$, we know that the gradient of $f(\boldsymbol{U}_t)$ vanishes. Therefore, we have proved Theorem 4.3.

Proof. According to Proposition 4.4, the sequences $\{\frac{1}{t}\boldsymbol{A}_t\}$ and $\{\frac{1}{t}\boldsymbol{B}_t\}$ are uniformly bounded. Then, there exist sub-sequences of $\{\frac{1}{t}\boldsymbol{A}_t\}$ and $\{\frac{1}{t}\boldsymbol{B}_t\}$ that converge to A_∞ and B_∞ respectively. In that case, $\boldsymbol{U}_t$ converges to $\boldsymbol{U}_\infty$. Let $\boldsymbol{W}$ be an arbitrary matrix in $\mathbb{R}^{d\times r}$, and $\{h_k\}$ be a positive sequence that converges to zero.

Since g_t is the surrogate function of f_t, for all t and k, we have

$$g_t(\boldsymbol{U}_t + h_k\boldsymbol{W}) \geq f_t(\boldsymbol{U}_t + h_k\boldsymbol{W}).$$

Let t tend to infinity:

$$g_\infty(\boldsymbol{U}_\infty + h_k\boldsymbol{W}) \geq f(\boldsymbol{U}_\infty + h_k\boldsymbol{W}).$$

Since $\boldsymbol{U}_t$ is uniformly bounded, when t tends to infinity, the term $\frac{\lambda_1}{2t}\left\|\boldsymbol{U}_t\right\|_\infty^2$ will vanish. In this way, $g_t(\cdot)$ becomes differentiable. Also, the Lipschitz of $\nabla f(\boldsymbol{U})$ (proved in Proposition 4.17) implies that the second derivative of $f(\boldsymbol{U}_t)$ can be uniformly bounded. And by a simple calculation, this also holds for $g_t(\boldsymbol{U}_t)$. Thus, we can take the first order Taylor expansion even when t tends to

infinity. Using a first order Taylor expansion, and note the fact that $g_\infty(\boldsymbol{U}_\infty) = f(\boldsymbol{U}_\infty)$, we have

$$\mathrm{Tr}\left(h_k \boldsymbol{W}^\top \nabla g_\infty(\boldsymbol{U}_\infty)\right) + o(h_k \boldsymbol{W}) \geq \mathrm{Tr}\left(h_k \boldsymbol{W}^\top \nabla f(\boldsymbol{U}_\infty)\right) + o(h_k \boldsymbol{W}).$$

Since $\{h_k\}$ is a positive sequence, by multiplying $\frac{1}{h_k \|\boldsymbol{W}\|_F}$ on both side, it follows that

$$\mathrm{Tr}\left(\frac{1}{\|\boldsymbol{W}\|_F} \boldsymbol{W}^\top \nabla g_\infty(\boldsymbol{U}_\infty)\right) + \frac{o(h_k \boldsymbol{W})}{h_k \|\boldsymbol{W}\|_F} \geq \mathrm{Tr}\left(\frac{1}{\|\boldsymbol{W}\|_F} \boldsymbol{W}^\top \nabla f(\boldsymbol{U}_\infty)\right) + \frac{o(h_k \boldsymbol{W})}{h_k \|\boldsymbol{W}\|_F}.$$

Now let k tend to infinity:

$$\mathrm{Tr}\left(\frac{1}{\|\boldsymbol{W}\|_F} \boldsymbol{W}^\top \nabla g_\infty(\boldsymbol{U}_\infty)\right) \geq \mathrm{Tr}\left(\frac{1}{\|\boldsymbol{W}\|_F} \boldsymbol{W}^\top \nabla f(\boldsymbol{U}_\infty)\right).$$

Since the inequality holds for all matrix $\boldsymbol{W} \in \mathbb{R}^{d \times r}$, it can easily show that

$$\nabla g_\infty(\boldsymbol{U}_\infty) = \nabla f(\boldsymbol{U}_\infty).$$

Since $\boldsymbol{U}_t$ always minimizes $g_t(\cdot)$, we have

$$\nabla f(\boldsymbol{U}_\infty) = \nabla g_\infty(\boldsymbol{U}_\infty) = 0,$$

which implies that when t tend to infinity, $\boldsymbol{U}_t$ is a stationary point of $f(\cdot)$. □

Chapter 5

Incremental Minimization for Low-Rank Subspace Clustering

5.1 Background

In modern scientific computing, data are routinely generated from a union of small subspaces for which, traditional tools such as principal component analysis are not able to identify the group structure. As an alternative, in the past a few years, subspace clustering [145, 136] has been extensively studied and has established solid applications in, for example, computer vision [56] and network topology inference [57]. Among many subspace clustering methods which seek a structured representation to fit the underlying data, two prominent examples are sparse subspace clustering (SSC) [56, 137] and low-rank representation (LRR) [92]. Both of them utilize the idea of self-expressiveness, i.e., expressing each sample as a linear combination of the remaining. Hence, the coefficients quickly suggest which data points belong to the same group (i.e., subspace). What is of difference is that SSC pursues a sparse solution, i.e., using a small fraction of samples to represent each data point. In this light, not only the clustering structure can be found by SSC, but also the most correlated samples are detected. For LRR, it prefers a (global) low-rank structure. This is motivated by many applications where the union of the subspaces is still of low rank. An appealing property of LRR is that it comes up with the theoretical guarantee of recovering the true low-rank model in addition to clustering the samples.

In this chapter, we are interested in the LRR method, which is shown to achieve the state-of-the-art performance on a broad range of real-world problems [92]. Recently, [91] demonstrated that, when equipped with a proper dictionary, LRR can even handle the coherent data – a challenging issue in the literature [36, 34] but is ubiquitous in realistic data sets such as the Netflix movie rating problem.

Formally, the LRR program we investigate here is formulated as follows [92]:

$$\min_{\boldsymbol{X},\boldsymbol{E}} \frac{\lambda_1}{2} \|\boldsymbol{Z} - \boldsymbol{Y}\boldsymbol{X} - \boldsymbol{E}\|_F^2 + \|\boldsymbol{X}\|_* + \lambda_2 \|\boldsymbol{E}\|_1 . \tag{5.1}$$

Here, $\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n) \in \mathbb{R}^{d\times n}$ is the observation matrix with n samples, each of which is a d-dimensional column vector. The matrix $\boldsymbol{Y} \in \mathbb{R}^{d\times n}$ is a given dictionary, $\boldsymbol{E}$ is some possible sparse corruption and $\lambda_1 > 0$ and $\lambda_2 > 0$ are two tunable parameters. Typically, $\boldsymbol{Y}$ is chosen as the data set $\boldsymbol{Z}$ itself, hence the idea of self-expressiveness. As $\boldsymbol{X}$ is penalized by the nuclear norm which is a convex surrogate of the rank function, the program seeks a low-rank representation among all samples, each of which can be approximated by a linear combination of the atoms in the dictionary $\boldsymbol{Y}$. Note that a column of $\boldsymbol{X}$ is the coefficients associated with an individual sample. When the optimal solution is obtained, one may perform spectral clustering [109] on $\boldsymbol{X}$ since the entries reflect the correlation between the (uncorrupted) data.

To aid intuition and to introduce more background for the LRR program (5.1), we discuss the ability of subspace clustering of LRR in the noiseless case. That is, the data matrix $\boldsymbol{Z}$ is generated from a union of (low-rank) subspaces. Since each sample can be represented only by those belonging to the same subspace, one may simply solve the linear system

$$\boldsymbol{Z} = \boldsymbol{Z}\boldsymbol{X},$$

which certainly produces a block diagonal solution of $\boldsymbol{X}$ as long as the small subspaces are independent. That being said, a careful reader may observe that the above linear system does not exclude the trivial identity matrix $\boldsymbol{X} = \boldsymbol{I}_n$. In order to alleviate it and to conform the low-rank structure of

the data, LRR looks for the one with the lowest rank, i.e.,

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_*, \text{ s.t. } \boldsymbol{Z} = \boldsymbol{Z}\boldsymbol{X}. \tag{5.2}$$

The following lemma, which is due to [93], justifies that the above program correctly segments the data.

Lemma 5.1. *Suppose that there are k number of small subspaces and without loss of generality, the data matrix is organized as $\boldsymbol{Z} = (\boldsymbol{Z}_1, \boldsymbol{Z}_2, \cdots, \boldsymbol{Z}_k)$, where $\boldsymbol{Z}_i \in \mathbb{R}^{d \times n_i}$ is the collection of samples coming from the ith subspace. Further assume that there are sufficient samples for each subspace such that $n_i > \operatorname{rank}(\boldsymbol{Z}_i)$. If the subspaces are mutually independent, then there exists an optimal solution to* (5.2)*,*

$$\boldsymbol{X}^* = \begin{pmatrix} \boldsymbol{X}_1^* & 0 & 0 & 0 \\ 0 & \boldsymbol{X}_2^* & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \boldsymbol{X}_k^* \end{pmatrix}$$

such that $\operatorname{rank}(\boldsymbol{X}_i^*) = \operatorname{rank}(\boldsymbol{Z}_i)$ *for all* $1 \leq i \leq k$.

More generally, one may think of the noisy model $\boldsymbol{Z} = \boldsymbol{Y}\boldsymbol{X} + \boldsymbol{E} + \boldsymbol{G}$ for some given dictionary $\boldsymbol{Y}$, a sparse corruption $\boldsymbol{E}$ and a Gaussian noise $\boldsymbol{G}$. In this case, program (5.1) is a straightforward extension to achieve the robustness. While a large body of work has shown that LRR is able to segment the data (see, e.g., [91]), three issues are immediately incurred for LRR in the face of big data:

- $(I1)$ Memory cost of $\boldsymbol{X}$. In the LRR formulation (5.1), there is typically no sparsity assumption on $\boldsymbol{X}$. Hence, the memory footprint of $\boldsymbol{X}$ is proportional to n^2 which precludes most of the recently developed nuclear norm solvers [90, 71, 4, 69].

- $(I2)$ Computational cost of $\|\boldsymbol{X}\|_*$. Since the size of the nuclear norm regularized matrix $\boldsymbol{X}$ is $n \times n$, optimizing such problems can be computationally expensive even when n is moderate, say $n = 1000$ [120].

$(I3)$ Memory cost of $\boldsymbol{Y}$. As the size of the dictionary $\boldsymbol{Y}$ is proportional to sample size n, it is prohibitive to store the entire dictionary $\boldsymbol{Y}$ during optimization when n is large.

To remedy these problems, especially the memory bottleneck, one potential way is solving the LRR program in an online manner. That is, we sequentially reveal the samples $\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n$ and update the components in $\boldsymbol{X}$ and $\boldsymbol{E}$. Nevertheless, it turns out that the LRR program (5.1) is not suitable for online minimization. To see this, we note that each column of $\boldsymbol{X}$ is the coefficients of a sample with respect to the *entire* dictionary $\boldsymbol{Y}$, for example, $\boldsymbol{z}_1 \approx \boldsymbol{Y}\boldsymbol{x}_1 + \boldsymbol{e}_1$. This indicates that without further technique, we have to load the entire dictionary Y so as to update $\boldsymbol{x}_1$ and $\boldsymbol{e}_1$. Again, this yields an $\mathcal{O}(n)$ memory cost. Hence, for our purpose, we need to tackle a more serious challenge:

$(I4)$ Partial realization of $\boldsymbol{Y}$. We are required to guarantee the optimality of the solution but can only access part of the atoms of $\boldsymbol{Y}$ in each iteration.

5.1.1 Contributions

In this chapter, henceforth, we propose a new algorithm termed online low-rank subspace clustering (OLRSC), which admits a low computational complexity. Compared to existing solvers, OLRSC reduces the memory cost of LRR from $\mathcal{O}\left(n^2\right)$ to $\mathcal{O}(dr)$ where r is an estimated rank of the uncorrupted data ($r < d \ll n$). This nice property makes OLRSC an appealing solution for large-scale subspace clustering problems. Furthermore, we prove that the sequence of the solutions produced by OLRSC converges to a stationary point of the expected loss function asymptotically even though one atom of $\boldsymbol{Y}$ is available at each iteration. In a nutshell, OLRSC resolves *all* practical issues of LRR and still promotes global low-rank structure – the merit of LRR. Finally, concerning the robustness of the algorithm, our empirical study suggests accurate recovery of the true subspace even when a large fraction of the entries are corrupted.

5.1.2 Related Work

Low-rankness has been studied for more than two decades. As one of the earliest work, [58] appealed to a rank minimization program for system identification and signal processing. It was further suggested that the nuclear norm (also known as trace norm) is a good convex surrogate to the rank of a matrix. Such a key observation was utilized in machine learning for recommender

systems [138, 121]. Though empirically effective, it was not clear under which conditions such a nuclear norm based program exactly returns the true low-rank solution. In the seminal work of [34], it was shown that if the singular vectors of the true low-rank matrix does not concentrate around the canonical basis, formally referred to as the incoherence property, then a convex program with proper parameter setting recovers the underlying matrix even a constant fraction of the entries are corrupted. Following [34], a considerable number of work extends the low-rank model to accommodate outliers [154], high-dimensional case [153], graph clustering [43], to name a few. This work considers the variant of multiple subspaces, which is due to [93]. In particular, in place of drawing insight on the statistical performance, we mainly focus on an efficient and provable algorithm for subspace clustering. There is a plethora of work attempting to mitigate the memory and computational bottleneck of the nuclear norm regularizer. However, to the best of our knowledge, none of them can handle Issue $(I3)$ and Issue $(I4)$.

One of the most popular ways to alleviate the huge memory cost is online implementation. [59] devised an online algorithm for the robust principal component analysis (RPCA) problem, which makes the memory cost independent of the sample size. Yet, compared to RPCA where the size of the nuclear norm regularized matrix is $d \times n$, that of LRR is $n \times n$ – a worse and more challenging case. Moreover, their algorithm cannot address the partial dictionary issue that emerges in our case.

To tackle the computational overhead, [71] utilized a sparse semi-definite programming solver to derive a simple yet efficient algorithm. Unfortunately, the memory requirement of their algorithm is proportional to the number of observed entries, making it impractical when the regularized matrix is large and dense (which is the case of LRR). [4] combined stochastic subgradient and incremental SVD to boost efficiency. But for the LRR problem, the type of the loss function does not meet the requirements and thus, it is still not practical to use that algorithm in our case.

Another line in the literature explores a structured formulation of LRR beyond the low-rankness. For example, [150] provably showed that combining LRR and SSC can take advantages of both methods. Whereas [150] promotes the sparsity on the representation matrix $\boldsymbol{X}$, [128] demonstrated how to pursue a sparsity structure on the factors of $\boldsymbol{X}$, which is known to be more flexible [68]. Due to the non-convexity of matrix factorization, a large body of work is dedicated to characterizing the conditions under which gradient descent ensures global optimum [72, 88, 64].

5.2 Problem Formulation and Algorithm

Recall the LRR program given in (5.1). Our goal is to efficiently learn the representation matrix $\boldsymbol{X}$ and the corruption matrix $\boldsymbol{E}$ in an online manner so as to mitigate the issues mentioned. To this end, we reformulate it as an empirical risk minimization problem which is amenable for online optimization.

The first technique for our purpose is a *non-convex reformulation* of the nuclear norm. Assume that the rank of $\boldsymbol{X}$ is at most r. Then [58] showed that,

$$\|\boldsymbol{X}\|_* = \min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{X}=\boldsymbol{U}^\top\boldsymbol{V}} \frac{1}{2}\left(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2\right), \tag{5.3}$$

where $\boldsymbol{U} \in \mathbb{R}^{r\times n}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times n}$. The minimum can be attained at, for example, $\boldsymbol{U}^\top = \boldsymbol{U}_0\boldsymbol{S}_0^{1/2}$ and $\boldsymbol{V}^\top = \boldsymbol{V}_0\boldsymbol{S}_0^{1/2}$ where $\boldsymbol{X} = \boldsymbol{U}_0 S_0 \boldsymbol{V}_0^\top$ is the singular value decomposition. In this way, (5.1) can be written as follows:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} \frac{\lambda_1}{2}\left\|\boldsymbol{Z} - \boldsymbol{Y}\boldsymbol{U}^\top\boldsymbol{V} - \boldsymbol{E}\right\|_F^2 + \frac{1}{2}\|\boldsymbol{U}\|_F^2 + \frac{1}{2}\|\boldsymbol{V}\|_F^2 + \lambda_2\|\boldsymbol{E}\|_1. \tag{5.4}$$

Note that by this reformulation, updating the entries in $\boldsymbol{X}$ amounts to sequentially updating the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$, as shown below:

$$\underbrace{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{pmatrix}}_{\boldsymbol{X}\in\mathbb{R}^{n\times n}} = \underbrace{\begin{pmatrix} \boldsymbol{u}_1^\top \\ \boldsymbol{u}_2^\top \\ \vdots \\ \boldsymbol{u}_n^\top \end{pmatrix}}_{\boldsymbol{U}^\top\in\mathbb{R}^{n\times r}} \times \underbrace{\begin{pmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{pmatrix}}_{\boldsymbol{V}\in\mathbb{R}^{r\times n}}.$$

For instance, when we have $\{\boldsymbol{u}_i\}_{i=1}^t$ and $\{\boldsymbol{v}_j\}_{j=1}^t$ on hand, we are able to recover the components $\{x_{ij}\}_{1\le i,j\le t}$ since each x_{ij} equals $\boldsymbol{u}_i^\top\boldsymbol{v}_j$. It is worth mentioning that this technique is utilized in [59] for online RPCA. Unfortunately, the size of $\boldsymbol{U}$ and $\boldsymbol{V}$ in our problem are both proportional to the sample size n and the dictionary $\boldsymbol{Y}$ is partially observed in each iteration, making the algorithm in [59] not applicable to LRR. Related to the online implementation, another challenge is that, all the columns of $\boldsymbol{U}$ are coupled together at this moment as $\boldsymbol{U}^\top$ is left multiplied by $\boldsymbol{Y}$ in the first term

of (5.4). Since we do not want to load the entire dictionary $\boldsymbol{Y}$, this makes it difficult to sequentially compute the columns of $\boldsymbol{U}$.

For the sake of decoupling the columns of $\boldsymbol{U}$, as part of the crux of our techniques, we introduce an auxiliary variable $\boldsymbol{D} = \boldsymbol{Y}\boldsymbol{U}^\top$, whose size is $d \times r$ (i.e., independent of the sample size n). Interestingly, in this way, we are approximating the term $\boldsymbol{Z} - \boldsymbol{E}$ with $\boldsymbol{D}\boldsymbol{V}$, which provides an intuition on the role of $\boldsymbol{D}$: namely, $\boldsymbol{D}$ can be seen as a *basis dictionary* of the clean data, with $\boldsymbol{V}$ being the coefficients.

These key observations allow us to derive an equivalent reformulation to LRR (5.1):

$$\min_{\boldsymbol{D},\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} \frac{\lambda_1}{2}\left\|\boldsymbol{Z}-\boldsymbol{D}\boldsymbol{V}-\boldsymbol{E}\right\|_F^2+\frac{1}{2}\left(\left\|\boldsymbol{U}\right\|_F^2+\left\|\boldsymbol{V}\right\|_F^2\right)+\lambda_2\left\|\boldsymbol{E}\right\|_1, \quad \text{s.t. } \boldsymbol{D}=\boldsymbol{Y}\boldsymbol{U}^\top.$$

By penalizing the constraint in the objective, we obtain a *regularized* version of LRR on which our new algorithm is based:

$$\min_{\boldsymbol{D},\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} \frac{\lambda_1}{2}\left\|\boldsymbol{Z}-\boldsymbol{D}\boldsymbol{V}-\boldsymbol{E}\right\|_F^2+\frac{1}{2}\left(\left\|\boldsymbol{U}\right\|_F^2+\left\|\boldsymbol{V}\right\|_F^2\right)+\lambda_2\left\|\boldsymbol{E}\right\|_1+\frac{\lambda_3}{2}\left\|\boldsymbol{D}-\boldsymbol{Y}\boldsymbol{U}^\top\right\|_F^2. \tag{5.5}$$

We note that there are two advantages of (5.5) compared to (5.1). First, it is amenable for online optimization. Second, it is more informative since it explicitly models the basis of the union of subspaces, which yields a better subspace recovery and clustering to be shown in Section 5.4.

We also point out that due to our explicit modeling of the basis, we unify LRR and RPCA as follows: for LRR, $\boldsymbol{D} \approx \boldsymbol{Y}\boldsymbol{U}^\top$ (or $\boldsymbol{D} = \boldsymbol{Y}\boldsymbol{U}^\top$ if λ_3 tends to infinity) while for RPCA, $\boldsymbol{D} = \boldsymbol{U}^\top$. That is, ORPCA [59] considers a problem of $\boldsymbol{Y} = \boldsymbol{I}_d$ whose size is independent of n, hence can be kept in memory which naturally resolves Issue $(I3)$ and $(I4)$. This is why RPCA can be easily implemented in an online fashion while LRR cannot.

Now we return to the online implementation of LRR. The main idea is optimizing a surrogate of the empirical risk function (to be defined) in each iteration, which is fast and memory efficient.

Let $\boldsymbol{z}_i$, $\boldsymbol{y}_i$, $\boldsymbol{e}_i$, $\boldsymbol{u}_i$, and $\boldsymbol{v}_i$ be the ith column of matrices $\boldsymbol{Z}$, $\boldsymbol{Y}$, $\boldsymbol{E}$, $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively and let

$$\begin{aligned}\tilde{\ell}(\boldsymbol{D},\boldsymbol{v},\boldsymbol{e};\boldsymbol{z}) &:= \frac{\lambda_1}{2}\left\|\boldsymbol{z}-\boldsymbol{D}\boldsymbol{v}-\boldsymbol{e}\right\|^2+\frac{1}{2}\left\|\boldsymbol{v}\right\|^2+\lambda_2\left\|\boldsymbol{e}\right\|_1,\\ \ell(\boldsymbol{D};\boldsymbol{z}) &:= \min_{\boldsymbol{v},\boldsymbol{e}}\tilde{\ell}(\boldsymbol{D},\boldsymbol{v},\boldsymbol{e};\boldsymbol{z}).\end{aligned} \tag{5.6}$$

Further, we define

$$\begin{aligned}\tilde{h}(\boldsymbol{D},\boldsymbol{U};\boldsymbol{Y}) &:= \sum_{i=1}^{n}\frac{1}{2}\left\|\boldsymbol{u}_i\right\|^2+\frac{\lambda_3}{2}\left\|\boldsymbol{D}-\sum_{i=1}^{n}\boldsymbol{y}_i\boldsymbol{u}_i^\top\right\|_F^2,\\ h(\boldsymbol{D};\boldsymbol{Y}) &:= \min_{\boldsymbol{U}}\tilde{h}(\boldsymbol{D},\boldsymbol{U};\boldsymbol{Y}).\end{aligned} \tag{5.7}$$

Then (5.5) can be rewritten as follows:

$$\min_{\boldsymbol{D}}\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}}\sum_{i=1}^{n}\tilde{\ell}(\boldsymbol{D},\boldsymbol{v}_i,\boldsymbol{e}_i;\boldsymbol{z}_i)+\tilde{h}(\boldsymbol{D},\boldsymbol{U};\boldsymbol{Y}), \tag{5.8}$$

which amounts to minimizing the *empirical loss* function:

$$\min_{\boldsymbol{D}} f_n(\boldsymbol{D}) := \frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{D};\boldsymbol{z}_i)+\frac{1}{n}h(\boldsymbol{D};\boldsymbol{Y}). \tag{5.9}$$

5.2.1 Expected Loss

In stochastic approximation, we are also interested in analyzing the optimality of the obtained solution with respect to the expected loss function [23]. To this end, we first derive the optimal solutions $\tilde{\boldsymbol{U}}$, $\tilde{\boldsymbol{V}}$ and $\tilde{\boldsymbol{E}}$ that minimize (5.8) which renders a concrete form of the empirical loss function $f_n(\boldsymbol{D})$.

Given $\boldsymbol{D}$, we need to compute the optimal solutions $\tilde{\boldsymbol{U}}$, $\tilde{\boldsymbol{V}}$ and $\tilde{\boldsymbol{E}}$ to evaluate the objective value of $f_n(\boldsymbol{D})$. What is of interest here is that, the optimization procedure of $\boldsymbol{U}$ is quite different from that of $\boldsymbol{V}$ and $\boldsymbol{E}$. According to (5.6), when $\boldsymbol{D}$ is given, each $\tilde{\boldsymbol{v}}_i$ and $\tilde{\boldsymbol{e}}_i$ can be solved by only accessing the ith sample $\boldsymbol{z}_i$. However, the optimal $\tilde{\boldsymbol{u}}_i$ depends on the whole dictionary $\boldsymbol{Y}$ as the second term in $\tilde{h}(\boldsymbol{Y},\boldsymbol{D},\boldsymbol{U})$ couples all the $\boldsymbol{u}_i$'s. Fortunately, we can obtain a closed form solution for each $\tilde{\boldsymbol{u}}_i$, as stated below.

Proposition 5.2. *Suppose that $\boldsymbol{Y}$ and $\boldsymbol{D}$ are fixed. Then the optimal solution $\tilde{\boldsymbol{U}}=(\tilde{\boldsymbol{u}}_1,\tilde{\boldsymbol{u}}_2,\ldots,\tilde{\boldsymbol{u}}_n)$*

that minimizes $\tilde{h}(\boldsymbol{D}, \boldsymbol{U}; \boldsymbol{Y})$ *and Eq.* (5.5) *is given by*

$$\tilde{\boldsymbol{u}}_i = \boldsymbol{D}^\top \left(\frac{1}{\lambda_3} \boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top \right)^{-1} \boldsymbol{y}_i, \ \forall\, i \in [n]. \tag{5.10}$$

Hence,

$$h(\boldsymbol{D}; \boldsymbol{Y}) = \frac{1}{2} \left\| \boldsymbol{D}^\top \left(\frac{1}{\lambda_3} \boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top \right)^{-1} \boldsymbol{Y} \right\|_F^2 + \frac{1}{2\lambda_3} \left\| \left(\frac{1}{\lambda_3} \boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top \right)^{-1} \boldsymbol{D} \right\|_F^2. \tag{5.11}$$

The result follows by noting the first order optimality condition and some basic algebra. The proof can be found in Appendix 5.B.2.

Let $\sigma_i(\boldsymbol{Y})$ be the singular values of the matrix $\boldsymbol{Y} \in \mathbb{R}^{d \times n}$ where $1 \leq i \leq d$. By calculation, it is not hard to see that

$$\sigma_{\max} \left(\left(\frac{1}{\lambda_3} \boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top \right)^{-1} \right) = \frac{1}{(\sigma_p(\boldsymbol{Y}))^2 + \lambda_3^{-1}} \leq \lambda_3,$$
$$\sigma_{\max} \left(\left(\frac{1}{\lambda_3} \boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top \right)^{-1} \boldsymbol{Y} \right) = \frac{\sigma_i(\boldsymbol{Y})}{(\sigma_i(\boldsymbol{Y}))^2 + \lambda_3^{-1}} \leq \frac{\sqrt{\lambda_3}}{2} \text{ for some } i.$$

Thereby, we obtain the upper bound

$$h(\boldsymbol{D}; \boldsymbol{Y}) \leq \frac{\lambda_3}{8} \|\boldsymbol{D}\|_F^2 + \frac{\lambda_3}{2} \|\boldsymbol{D}\|_F^2 \leq \frac{5\lambda_3}{8} \|\boldsymbol{D}\|_F^2.$$

Suppose that $\boldsymbol{D}$ is fixed. Also notice that the size of $\boldsymbol{D}$ does not grow with n. It follows that

$$\lim_{n \to \infty} \frac{1}{n} h(\boldsymbol{D}; \boldsymbol{Y}) = 0,$$

if λ_3 is such that $\lim_{n \to \infty} \lambda_3 / n = 0$. Hence, asymptotically, minimizing the empirical loss $f_n(\boldsymbol{D})$ amounts to optimizing $n^{-1} \sum_{i=1}^n \ell(\boldsymbol{D}; \boldsymbol{z})$. Note that when λ_1 and λ_2 are independent of n, we quickly get

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n \ell(\boldsymbol{D}; \boldsymbol{z}_i) = \mathbb{E}_{\boldsymbol{z}}[\ell(\boldsymbol{D}; \boldsymbol{z})], \tag{5.12}$$

assuming that all the samples are drawn i.i.d. from some unknown distribution. This gives the

expected loss function

$$f(\boldsymbol{D}) := \mathbb{E}_{\boldsymbol{z}}[\ell(\boldsymbol{z}, \boldsymbol{D})] = \lim_{n\to\infty} f_n(\boldsymbol{D}). \tag{5.13}$$

5.2.2 Algorithm

We are now in the position to elaborate an online optimization algorithm for low-rank subspace clustering. Namely, we show how to solve (5.9) in an efficient manner. From a high level, we alternatively minimize over the variables $\boldsymbol{D}$ and $\{\boldsymbol{u}_i, \boldsymbol{v}_i, \boldsymbol{e}_i\}_{i=1}^n$. That is, we begin with an initial (and inaccurate) guess of the variable $\boldsymbol{D}$, and solve the subproblems (5.6) and (5.7). Using the obtained solution, we construct a surrogate function of the empirical loss (5.9), which is further optimized to refine our initial guess of $\boldsymbol{D}$. Such a paradigm was utilized in [99, 98, 133] for different problems. It turns out that the case studied here is more challenging due to the component $h(\boldsymbol{D}; \boldsymbol{Y})$ of the empirical loss function. To optimize it in an online manner and to analyze the performance, we develop novel techniques that will be described in the following.

Optimize $\boldsymbol{v}$ and $\boldsymbol{e}$

For exposition, we first investigate the sum of $\ell(\boldsymbol{D}; \boldsymbol{z}_i)$ where $1 \leq i \leq n$ in (5.9). This is where the variables $\boldsymbol{v}_i$ and $\boldsymbol{e}_i$ are involved. Suppose that at the t-th iteration, a fresh sample $\boldsymbol{z}_t$ is drawn and we have an initial guess of $\boldsymbol{D}$, say $\boldsymbol{D}_{t-1}$. Then it is easy to see that the optimal solution $\{\boldsymbol{v}_t^*, \boldsymbol{e}_t^*\}$ that minimizes $\tilde{\ell}(\boldsymbol{D}_{t-1}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z}_t)$ is given by

$$\{\boldsymbol{v}_t^*, \boldsymbol{e}_t^*\} = \arg\min_{\boldsymbol{v},\boldsymbol{e}} \ \frac{\lambda_1}{2} \left\|\boldsymbol{z}_t - \boldsymbol{D}_{t-1}\boldsymbol{v} - \boldsymbol{e}\right\|^2 + \frac{1}{2}\left\|\boldsymbol{v}\right\|^2 + \lambda_2 \left\|\boldsymbol{e}\right\|_1.$$

Since the objective function is jointly strongly convex over the variables $\{\boldsymbol{v}, \boldsymbol{e}\}$, one may apply coordinate descent to obtain the optimum [14]. In particular, we observe that if $\boldsymbol{e}$ is fixed, we can optimize $\boldsymbol{v}$ in closed form:

$$\boldsymbol{v} = (\boldsymbol{D}_{t-1}^\top \boldsymbol{D}_{t-1} + \boldsymbol{I}_r/\lambda_1)^{-1} \boldsymbol{D}_{t-1}^\top (\boldsymbol{z}_t - \boldsymbol{e}). \tag{5.14}$$

Conversely, given $\boldsymbol{v}$, the variable $\boldsymbol{e}$ is obtained via soft-thresholding [49]:

$$\boldsymbol{e} = \mathcal{S}_{\lambda_2/\lambda_1}[\boldsymbol{z}_t - \boldsymbol{D}_{t-1}\boldsymbol{v}]. \tag{5.15}$$

See Algorithm 8 in Appendix 5.A for details. It follows that $\tilde{\ell}(\boldsymbol{D}_{t-1}, \boldsymbol{v}_t^*, \boldsymbol{e}_t^*; \boldsymbol{z}_t)$ is a surrogate to $\ell(\boldsymbol{D}; \boldsymbol{z}_t)$, since $\boldsymbol{D}_{t-1}$ is our guess of the optimal $\boldsymbol{D}$.

Optimize $\boldsymbol{u}$

We move on to describe how to optimize the second component $h(\boldsymbol{D}; \boldsymbol{Y})$, which contains the variables $\{\boldsymbol{u}_i\}_{i\geq 1}$. Again, we will assume that an initial guess of $\boldsymbol{D}$ is available, so we only need to minimize a surrogate. However, looking at the solution (5.10), we notice that even $\boldsymbol{D}$ is given (e.g., some initial guess $\boldsymbol{D}_{t-1}$), we have to load the *entire* dictionary $\boldsymbol{Y} \in \mathbb{R}^{d\times n}$ so as to obtain the local optimum $\tilde{\boldsymbol{u}}_t$, and hence a surrogate of $h(\boldsymbol{D}; \boldsymbol{Y})$. This is quite different from the scheme of optimizing $\boldsymbol{v}$ and $\boldsymbol{e}$, where optimum only depends on the new sample. Indeed, though the framework of our algorithm follows from [99, 59, 133], none of them considers the situation as we stated here (the optimum in their work can be computed directly when a new sample arrives).

In order to remedy the issue, our novelty here is constructing a proxy function by minimizing which, we are generating a solution that gradually approximates to the one of $\tilde{h}(\boldsymbol{D}, \boldsymbol{U}; \boldsymbol{Y})$. The proxy function is given as follows

$$\tilde{\ell}_2(\boldsymbol{D}, \boldsymbol{u}; \boldsymbol{M}, \boldsymbol{y}) := \frac{1}{2}\|\boldsymbol{u}\|^2 + \frac{\lambda_3}{2}\left\|\boldsymbol{D} - \boldsymbol{M} - \boldsymbol{y}\boldsymbol{u}^\top\right\|_F^2. \tag{5.16}$$

Suppose that we store the accumulation matrix

$$\boldsymbol{M}_{t-1} := \sum_{i=1}^{t-1} \boldsymbol{y}_i(\boldsymbol{u}_i^*)^\top \in \mathbb{R}^{d\times r}, \tag{5.17}$$

where we define M_0 as a zero matrix. At the t-th iteration, when a new atom $\boldsymbol{y}_t$ is revealed, we calculate $\boldsymbol{u}_t^*$ as the minimizer of $\tilde{\ell}_2(\boldsymbol{D}_{t-1}, \boldsymbol{u}; \boldsymbol{M}_{t-1}, \boldsymbol{y}_t)$, for which we have the closed form solution

$$\boldsymbol{u}_t^* = (\|\boldsymbol{y}_t\|^2 + 1/\lambda_3)^{-1}(\boldsymbol{D}_{t-1} - \boldsymbol{M}_{t-1})^\top \boldsymbol{y}_t. \tag{5.18}$$

Algorithm 7 Online Low-Rank Subspace Clustering

Require: $\boldsymbol{Z} \in \mathbb{R}^{d\times n}$ (observed samples), $\boldsymbol{Y} \in \mathbb{R}^{d\times n}$, parameters λ_1, λ_2 and λ_3, random matrix $\boldsymbol{D}_0 \in \mathbb{R}^{d\times r}$ (initial basis), zero matrices $\boldsymbol{M}_0$, $\boldsymbol{A}_0$ and $\boldsymbol{B}_0$.

Ensure: Optimal basis $\boldsymbol{D}_n$.

1: **for** $t = 1$ to n **do**

2: Access the t-th sample $\boldsymbol{z}_t$ and the t-th atom $\boldsymbol{y}_t$.

3: Compute the coefficient and noise:

$$\{\boldsymbol{v}_t^*, \boldsymbol{e}_t^*\} = \arg\min_{\boldsymbol{v},\boldsymbol{e}} \tilde{\ell}(\boldsymbol{D}_{t-1}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z}_t),$$
$$\boldsymbol{u}_t^* = \arg\min_{\boldsymbol{u}} \tilde{\ell}_2(\boldsymbol{D}_{t-1}, \boldsymbol{M}_{t-1}, \boldsymbol{u}; \boldsymbol{y}_t).$$

4: Update the accumulation matrices:

$$\begin{aligned}\boldsymbol{M}_t &\leftarrow \boldsymbol{M}_{t-1} + \boldsymbol{y}_t(\boldsymbol{u}_t^*)^\top,\\ \boldsymbol{A}_t &\leftarrow A_{t-1} + \boldsymbol{v}_t^*(\boldsymbol{v}_t^*)^\top,\\ \boldsymbol{B}_t &\leftarrow B_{t-1} + (\boldsymbol{z}_t - \boldsymbol{e}_t^*)(\boldsymbol{v}_t^*)^\top.\end{aligned}$$

5: Update the basis:

$$\boldsymbol{D}_t = \arg\min_{\boldsymbol{D}} \frac{1}{t}\left[\frac{1}{2}\operatorname{Tr}\left(\boldsymbol{D}^\top\boldsymbol{D}(\lambda_1\boldsymbol{A}_t + \lambda_3\boldsymbol{I}_r)\right) - \operatorname{Tr}\left(\boldsymbol{D}^\top(\lambda_1\boldsymbol{B}_t + \lambda_3\boldsymbol{M}_t)\right)\right].$$

6: **end for**

The solution in (5.18) differs from that of (5.10) a lot. Though (5.18) is not an accurate solution that minimizes $\tilde{h}(\boldsymbol{D}, \boldsymbol{U}; \boldsymbol{Y})$, the computation is efficient. More importantly, $\boldsymbol{u}_t^*$ only depends on $\boldsymbol{y}_t$ rather than the entire atom dictionary. We will also show that (5.18) suffices to guarantee the convergence of the algorithm to a stationary point of the expected loss function.

To gain intuition on the connection between $\tilde{\ell}_2(\boldsymbol{D}, \boldsymbol{u}; \boldsymbol{M}, \boldsymbol{y})$ and $\tilde{h}(\boldsymbol{D}, \boldsymbol{U}; \boldsymbol{Y})$, assume we have n atoms in total (i.e., $\boldsymbol{Y}$ has n columns). It turns out that for the function $\tilde{h}(\boldsymbol{D}, \boldsymbol{U}; \boldsymbol{Y})$, $\boldsymbol{U}$ can be optimized using block coordinate minimization (BCM), with each column of $\boldsymbol{U}$ being a block variable. Initially, we may set all these columns to a zero vector. As BCM proceeds, we update $\boldsymbol{u}_t$ while keeping the other $\boldsymbol{u}_i$'s for $i \neq t$. Note that in this way, optimizing $\tilde{h}(\boldsymbol{D}, \boldsymbol{U}; \boldsymbol{Y})$ over $\boldsymbol{u}_t$ amounts to minimizing the function $\tilde{\ell}_2(\boldsymbol{D}, \boldsymbol{u}; \boldsymbol{M}_{t-1}, \boldsymbol{y}_t)$ where M_{t-1} is defined in (5.17). So after revealing all the atoms, each $\boldsymbol{u}_t$ is sequentially updated only once. Henceforth, our strategy can be seen as a one-pass BCM algorithm for the function $\tilde{h}(\boldsymbol{D}, \boldsymbol{U}; \boldsymbol{Y})$.

Optimize $\boldsymbol{D}$

As soon as $\{\boldsymbol{v}_i^*, \boldsymbol{e}_i^*, \boldsymbol{u}_i^*\}_{i=1}^t$ are available, we can refine the initial guess $\boldsymbol{D}_{t-1}$ by optimizing the surrogate function

$$g_t(\boldsymbol{D}) := \frac{1}{t}\Big(\sum_{i=1}^t \tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}_i^*, \boldsymbol{e}_i^*; \boldsymbol{z}_i) + \sum_{i=1}^t \frac{1}{2}\|\boldsymbol{u}_i^*\|^2 + \frac{\lambda_3}{2}\|\boldsymbol{D} - \boldsymbol{M}_t\|_F^2 \Big). \tag{5.19}$$

Let us look at the first term:

$$\tilde{g}_t(\boldsymbol{D}) := \frac{1}{t}\sum_{i=1}^t \tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}_i^*, \boldsymbol{e}_i^*; \boldsymbol{z}_i), \tag{5.20}$$

which is a surrogate of $\frac{1}{t}\sum_{i=1}^t \ell(\boldsymbol{D}; \boldsymbol{z}_i)$. Though the above objective function involves the past iterates $\{\boldsymbol{v}_i^*, \boldsymbol{e}_i^*\}_{i=1}^t$ whose memory cost is proportional to the sample size, we show that the minimizer of (5.20) can be computed by accessing only two accumulation matrices whose sizes are independent of the sample size. To see this, we may expand the function $\tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}_i^*, \boldsymbol{e}_i^*; \boldsymbol{z}_i)$ as follows:

$$\begin{aligned}
\tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}_i^*, \boldsymbol{e}_i^*; \boldsymbol{z}_i) &= \frac{\lambda_1}{2}\|\boldsymbol{z}_i - \boldsymbol{D}\boldsymbol{v}_i^* - \boldsymbol{e}_i^*\|^2 + \frac{1}{2}\|\boldsymbol{v}_i^*\|^2 + \lambda_2\|\boldsymbol{e}_i^*\|_1 \\
&= \frac{\lambda_1}{2}\operatorname{Tr}\left(\boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{v}_i^*(\boldsymbol{v}_i^*)^\top\right) + \lambda_1 \operatorname{Tr}\left(\boldsymbol{D}^\top(\boldsymbol{e}_i^* - \boldsymbol{z}_i)(\boldsymbol{v}_i^*)^\top\right) \\
&\quad + \frac{\lambda_1}{2}\|\boldsymbol{z}_i - \boldsymbol{e}_i^*\|^2 + \frac{1}{2}\|\boldsymbol{v}_i^*\|^2 + \lambda_2\|\boldsymbol{e}_i^*\|_1 .
\end{aligned}$$

Since the variable here is $\boldsymbol{D}$, it holds that the minimizer of (5.20) is given by the following:

$$\min_{\boldsymbol{D}} \ \frac{1}{t}\sum_{i=1}^t \left[\frac{\lambda_1}{2}\operatorname{Tr}\left(\boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{v}_i^*(\boldsymbol{v}_i^*)^\top\right) + \lambda_1 \operatorname{Tr}\left(\boldsymbol{D}^\top(\boldsymbol{e}_i^* - \boldsymbol{z}_i)(\boldsymbol{v}_i^*)^\top\right) \right]. \tag{5.21}$$

As the trace is a linear mapping, solving the above program only needs to record two accumulation matrices

$$\boldsymbol{A}_t := \sum_{i=1}^t \boldsymbol{v}_i^*(\boldsymbol{v}_i^*)^\top \in \mathbb{R}^{r\times r}, \quad \boldsymbol{B}_t := \sum_{i=1}^t (\boldsymbol{z}_i - \boldsymbol{e}_i^*)(\boldsymbol{v}_i^*)^\top \in \mathbb{R}^{d\times r}. \tag{5.22}$$

Using the result of (5.21) and doing some calculation, we derive $\boldsymbol{D}_t$ as follows:

$$\begin{aligned}\boldsymbol{D}_t &= \arg\min_{\boldsymbol{D}} \frac{1}{t}\Big[\frac{1}{2}\operatorname{Tr}\Big(\boldsymbol{D}^\top \boldsymbol{D}(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}_r)\Big) - \operatorname{Tr}\Big(\boldsymbol{D}^\top(\lambda_1 \boldsymbol{B}_t + \lambda_3 \boldsymbol{M}_t)\Big)\Big] \\ &= (\lambda_1 \boldsymbol{B}_t + \lambda_3 \boldsymbol{M}_t)(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}_r)^{-1}, \end{aligned} \tag{5.23}$$

where $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$ are given in (5.22) and $\boldsymbol{M}_t$ is defined in (5.17). Numerically, we apply coordinate descent to solve the above program owing to its efficiency. See more details in Appendix 5.A.

Memory Cost

It is remarkable that the memory cost of Algorithm 7 is $\mathcal{O}(dr)$. To see this, note that when solving $\boldsymbol{v}_t^*$ and $\boldsymbol{e}_t^*$, we load $\boldsymbol{D}_{t-1}$ and a sample $\boldsymbol{z}_t$ into the memory, which costs $\mathcal{O}(dr)$. To compute the optimal $\boldsymbol{u}_t^*$, we need to access $\boldsymbol{D}_{t-1}$ and $\boldsymbol{M}_{t-1} \in \mathbb{R}^{d\times r}$. Although we aim to minimize (5.19), which seems to require all the past information, we actually only need to record $\boldsymbol{A}_t$, $\boldsymbol{B}_t$ and $\boldsymbol{M}_t$, whose sizes are at most $\mathcal{O}(dr)$ (recall that $r < d$).

Time Complexity

In addition to the memory efficiency, we further elaborate that the the computation in each iteration is cheap. To compute $\{\boldsymbol{v}_t^*, \boldsymbol{e}_t^*\}$, one may utilize the block coordinate method in [122] which enjoys linear convergence due to strong convexity. One may also apply the stochastic variance reduced algorithms which also ensure a geometric rate of convergence [152, 48]. The $\boldsymbol{u}_t^*$ is obtained by simple matrix-vector multiplications, which costs $\mathcal{O}(dr)$. It is easy to see the complexity of Step 4 is $\mathcal{O}(dr)$ and that of Step 5 is $\mathcal{O}\left(dr^2\right)$.

A Fully Online Subspace Clustering Scheme

Now we have provided a way to learn the low-rank representation matrix $\boldsymbol{X}$ in an online manner. Usually, researchers in the literature will take an optional post-processing step to refine the segmentation accuracy, for example, applying spectral clustering [109] on the obtained representation matrix $\boldsymbol{X}^*$. In this case, one has to collect all the $\boldsymbol{u}_i^*$'s and $\boldsymbol{v}_i^*$'s to compute $\boldsymbol{X}^* = (\boldsymbol{U}^*)^\top \boldsymbol{V}^*$ which will again increase the memory cost to $\mathcal{O}\left(n^2\right)$. Here, we suggest an alternative scheme which admits $\mathcal{O}(kr)$ memory usage where k is the number of subspaces. The idea is appealing to

the well-known k-means clustering in place of spectral clustering. One notable advantage is that updating the k-means model can be implemented in an online manner. In fact, the online k-means algorithm can be easily integrated into Algorithm 7 by observing that $\boldsymbol{v}_i^*$ is the robust feature for the ith sample. On the other hand, updating the k-means model is quite cheap, since the computational cost is $\mathcal{O}(kr)$.

An Alternative Online Implementation

Our strategy for solving $\boldsymbol{u}_t$ is based on a carefully designed proxy function which resolves Issue $(I4)$ and has a low complexity. Yet, to tackle Issue $(I4)$, another potential way is to avoid the variable $\boldsymbol{u}_t$. Recall that we derive the optimal solution $\tilde{\boldsymbol{U}}$ (provided that $\boldsymbol{D}$ is given) to $\tilde{h}(\boldsymbol{D}, \boldsymbol{U}; \boldsymbol{Y})$ as follows (see Proposition 5.2):

$$\tilde{\boldsymbol{U}} = \boldsymbol{Y}^\top \left({\lambda_3}^{-1} \boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top \right)^{-1} \boldsymbol{D}.$$

Plugging it back to $\tilde{h}(\boldsymbol{D}, \boldsymbol{U}; \boldsymbol{Y})$, we obtain

$$\tilde{h}(\boldsymbol{D}, \boldsymbol{U}; \boldsymbol{Y}) = \frac{1}{2} \operatorname{Tr} \left(\boldsymbol{D}\boldsymbol{D}^\top \left(\boldsymbol{Q}_n - {\lambda_3}^{-1} \boldsymbol{Q}_n^2 \right) \right) + \frac{\lambda_3}{2} \left\| \boldsymbol{D} - {\lambda_3}^{-1} \boldsymbol{Q}_n \boldsymbol{D} \right\|_F^2,$$

where

$$\boldsymbol{Q}_n = \left({\lambda_3}^{-1} \boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top \right)^{-1}.$$

Here, the subscript of $\boldsymbol{Q}_n$ denotes the number of atoms in $\boldsymbol{Y}$. Note that the size of $\boldsymbol{Q}_n$ is $d \times d$. Hence, if we incrementally compute the accumulation matrix $\boldsymbol{Y}\boldsymbol{Y}^\top = \sum_{i=1}^t \boldsymbol{y}_i \boldsymbol{y}_i^\top$, we can update the variable $\boldsymbol{D}$ in an online fashion. Namely, at t-th iteration, we re-define the surrogate function as follows:

$$g_t(\boldsymbol{D}) := \frac{1}{t} \left[\sum_{i=1}^t \tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{D}, \boldsymbol{v}_i, \boldsymbol{e}_i) + \frac{\lambda_3}{2} \left\| \boldsymbol{D} - \frac{1}{\lambda_3} \boldsymbol{Q}_t \boldsymbol{D} \right\|_F^2 + \frac{1}{2} \operatorname{Tr} \left(\boldsymbol{D}\boldsymbol{D}^\top \left(\boldsymbol{Q}_t - \frac{1}{\lambda_3} \boldsymbol{Q}_t^2 \right) \right) \right].$$

Again, by noting the fact that $\tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{D}, \boldsymbol{v}_i, \boldsymbol{e}_i)$ only involves recording $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$, we show that the memory cost is independent of sample size. However, the main shortcoming is that the time

complexity of computing the inverse of a $d \times d$ matrix in each iteration is $\mathcal{O}\left(d^3\right)$ which is not efficient in the high-dimensional regime (the time complexity of ours is proportional to dr^2). Either, it is not clear how to analyze the convergence.

5.3 Theoretical Analysis

In this section, we present theoretical evidence that our algorithm guarantees that the sequence $\{\boldsymbol{D}_t\}_{t\geq 1}$ converges to a stationary point of the expected loss (5.9) asymptotically. We note that the problem studied here is non-convex, and hence convergence to a stationary point is the best one can hope in general [14]. As we have mentioned in related work, there are elegant results showing that convergence to global optimum is possible for batch alternating minimization under more stringent conditions on the data. See, for example, [72, 64].

We make two assumptions throughout our analysis.

$(A1)$ The observed data are generated i.i.d. from some (unknown) distribution and there exist constants α_0 and α_1, such that the conditions $0 < \alpha_0 \leq \|\boldsymbol{z}_t\| \leq \alpha_1$ and $\alpha_0 \leq \|\boldsymbol{y}_t\| \leq \alpha_1$ hold almost surely for all $t \geq 1$.

$(A2)$ The smallest singular value of $\frac{1}{t}\boldsymbol{A}_t$ is bounded away from zero almost surely.

Note that the first assumption is very mild. In fact, in many applications the data points are normalized to unit length, and in this case $\alpha_0 = \alpha_1 = 1$. To understand the second assumption, recall that $\boldsymbol{D}_t$ is computed in (5.23). If $\lambda_3 = o(t)$, then asymptotically the solution $\boldsymbol{D}_t$ is not unique. Hence, Assumption $(A2)$ ensures that we always have a unique solution that minimizes the surrogate function $g_t(\boldsymbol{D})$. Geometrically, as we have noted that $\boldsymbol{v}_i$ is the coefficient of $\boldsymbol{z}_i$ in terms of the basis dictionary $\boldsymbol{D}$, the assumption simply requires that the data points are in general positions.

We also need the following parameter settings.

$(P1)$ λ_1 and λ_2 are independent of the sample size n.

$(P2)$ $\lim_{n\to\infty} \lambda_3/n = 0$.

As we have shown in Section 5.2, these parameter scalings give the expected loss function as in (5.13). We note that $(P1)$ is also required in previous work [99, 59, 133], though not set out

explicitly. $(P2)$ is specific to our problem. It facilitates the characterization of the empirical loss function when n tends to infinity. Otherwise, even the convergence of the sequence $\{f_n(\boldsymbol{D})\}_{n\geq 1}$ is not clear (consider, e.g., $\lambda_3 = n \sin n$).

We present a fundamental result that will be heavily invoked in the subsequent analysis.

Proposition 5.3. *Let $\{\boldsymbol{u}_t^*\}_{t\geq 1}$, $\{\boldsymbol{v}_t^*\}_{t\geq 1}$, $\{\boldsymbol{e}_t^*\}_{t\geq 1}$ and $\{\boldsymbol{D}_t\}_{t\geq 1}$ be the sequence of the optimal solutions produced by Algorithm 7. Assume $(A1)$ and $(A2)$. Further suppose that λ_2 does not grow with t. Then, for all $t \geq 1$,*

1. *$\boldsymbol{v}_t^*$, $\boldsymbol{e}_t^*$, $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$ are uniformly bounded from above;*
2. *$\boldsymbol{M}_t$ is uniformly upper bounded;*
3. *$\boldsymbol{D}_t$ is supported on some compact set $\mathcal{D}$;*
4. *$\boldsymbol{u}_t^*$ is uniformly upper bounded.*

Note that we prove the result by assuming a weaker condition than $(P1)$: the parameter λ_2 does not grow with the sample size t. While it is not surprising to see that $\boldsymbol{u}_t^*, \boldsymbol{v}_t^*, \boldsymbol{e}_t^*$ and $\boldsymbol{D}_t$ are bounded from above due to the regularization, it is interesting to note that $\boldsymbol{M}_t = \sum_{i=1}^t \boldsymbol{y}_i(\boldsymbol{u}_i^*)^\top$ is also upper bounded. Intuitively, this holds since the last term in (5.5) imposes that $\boldsymbol{M}_t$ cannot deviate far from $\boldsymbol{D}_t$. However, the technical challenge is that $\boldsymbol{D}_t$ itself depends on $\boldsymbol{M}_t$, as shown in (5.23), and neither $\boldsymbol{D}_t$ nor $\boldsymbol{M}_t$ is bounded without the boundedness of the other. To remedy this, we propose a novel technique which conducts mathematical induction simultaneously on $\boldsymbol{D}_t$ and $\boldsymbol{M}_t$. This tremendously simplifies the proof of our earlier version [132]. See Appendix 5.B.3 for the proof.

The proposition has many implications. For example, the uniform boundedness of the solutions immediately implies that the surrogate function $g_t(\boldsymbol{D})$ and the empirical loss function $f_t(\boldsymbol{D})$ are bounded from above for all $\boldsymbol{D} \in \mathcal{D}$, which is necessary for the convergence.

Next, we prove that the sequence of $\{g_t(\boldsymbol{D}_t)\}_{t\geq 1}$ converges almost surely. We will make use of the following lemma, which is due to [22].

Lemma 5.4. *Let $(\Omega, \mathcal{F}, P)$ be a measurable probability space, ψ_t, for $t \geq 1$, be the realization of a stochastic process and $\mathcal{F}_t$ be the filtration by the past information at time t. Let*

$$\delta_t = \begin{cases} 1 & \text{if } \mathbb{E}[\psi_{t+1} - \psi_t \mid \mathcal{F}_t] > 0, \\ 0 & \text{otherwise}. \end{cases}$$

If for all t, $\psi_t \geq 0$ and $\sum_{t=1}^{\infty} \mathbb{E}[\delta_t(\psi_{t+1} - \psi_t)] < \infty$, then ψ_t is a quasi-martingale and converges almost surely. Moreover, it holds almost surely that

$$\sum_{t=1}^{\infty} |\mathbb{E}[\psi_{t+1} - \psi_t \mid \mathcal{F}_t]| < +\infty.$$

Based on the lemma, we show the following result.

Theorem 5.5. *Assume $(A1)$ and $(A2)$. Set the parameters λ_1, λ_2 and λ_3 such that they satisfy $(P1)$ and $(P2)$. Then the sequence of $\{g_t(\boldsymbol{D}_t)\}_{t\geq 1}$ converges almost surely, where $\{\boldsymbol{D}_t\}_{t\geq 1}$ is the solution produced by Algorithm 7. Moreover, the following holds almost surely:*

$$\sum_{t=1}^{\infty} \left|\mathbb{E}\left[\psi_{t+1} - \psi_t \mid \mathcal{F}_t\right]\right| < +\infty, \tag{5.24}$$

where $\mathcal{F}_t = \{\boldsymbol{z}_i, \boldsymbol{y}_i\}_{i=1}^{t}$.

Proof. (Sketch) We will view $\{g_t(\boldsymbol{D}_t)\}_{t\geq 1}$ as a non-negative stochastic process, with $\mathcal{F}_t = \{\boldsymbol{z}_i, \boldsymbol{y}_i\}_{i=1}^{t}$ be the filtration of the past information. In order to apply Lemma 5.4, we write $\psi_t := g_t(\boldsymbol{D}_t)$ and compute the variation between two consecutive iterations:

$$\begin{aligned} \psi_{t+1} - \psi_t = & \underbrace{g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t)}_{\zeta_1} + \underbrace{\frac{\tilde{f}_t(\boldsymbol{D}_t) - \tilde{g}_t(\boldsymbol{D}_t)}{t+1}}_{\zeta_2} + \underbrace{\frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - \tilde{f}_t(\boldsymbol{D}_t)}{t+1}}_{\zeta_3} \\ & + \left[\frac{1}{t+1} \sum_{i=1}^{t+1} \frac{1}{2} \|\boldsymbol{u}_i^*\|^2 + \frac{\lambda_3}{2(t+1)} \|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\|_F^2 \right. \\ & \left. - \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} \|\boldsymbol{u}_i^*\|^2 - \frac{\lambda_3}{2t} \|\boldsymbol{D}_t - \boldsymbol{M}_t\|_F^2 \right]. \end{aligned}$$

Here,

$$\tilde{f}_t(\boldsymbol{D}_t) = \frac{1}{t}\sum_{i=1}^{t} \ell(\boldsymbol{D}_t; \boldsymbol{z}_i).$$

Since $\boldsymbol{D}_{t+1}$ minimizes $g_{t+1}(\boldsymbol{D})$, we have $\zeta_1 \leq 0$. Also, recall that $\tilde{g}_t(\boldsymbol{D})$ is a surrogate of $\tilde{f}_t(\boldsymbol{D})$, implying $\zeta_2 \leq 0$. For ζ_3, the P-Donsker lemma (see Proposition 5.16) gives

$$\mathbb{E}\big[\mathbb{E}[\zeta_3 \mid \mathcal{F}_t]\big] \leq \frac{\mathrm{C}}{\sqrt{t}(t+1)}$$

for some constant C. Finally, we need to upper bound the terms in the brackets which are specific to the online subspace clustering problem. In light of the bound of ζ_3, one may look for a way to bound them with $\mathcal{O}\left(1/t^{3/2}\right)$. Surprisingly, it turns out that our algorithm automatically guarantees that the sum of the terms in the brackets is not greater than zero. This follows by noting the closed-form solution of $\boldsymbol{u}_i^*$ (5.18) and the fact that $\boldsymbol{M}_t = \sum_{i=1}^{t} \boldsymbol{y}_i(\boldsymbol{u}_i^*)^\top$. Putting all the pieces together completes the proof. □

We move on to show that the sequence of $\{f_t(\boldsymbol{D}_t)\}_{t\geq 1}$ converges. In particular, we justify that $g_t(\boldsymbol{D})$ acts as a good surrogate of $f_t(\boldsymbol{D})$ in the sense that the sequence of the surrogate converges to the same limit of that of the empirical loss.

Theorem 5.6. *Assume* $(A1)$ *and* $(A2)$. *Set the parameters* λ_1, λ_2 *and* λ_3 *such that they satisfy* $(P1)$ *and* $(P2)$. *Let* $\{\boldsymbol{D}_t\}_{t\geq 1}$ *be the solution produced by Algorithm 7. Then, the sequence of* $\{f_t(\boldsymbol{D}_t)\}_{t\geq 1}$ *converges almost surely to the same limit of* $\{g_t(\boldsymbol{D}_t)\}_{t\geq 1}$.

Corollary 5.7. *Assume sames conditions as in Theorem 5.6. Then the sequence of* $\{f(\boldsymbol{D}_t)\}_{t\geq 1}$ *converges almost surely to the same limit of* $\{f_t(\boldsymbol{D}_t)\}_{t\geq 1}$ *or equivalently,* $\{g_t(\boldsymbol{D}_t)\}_{t\geq 1}$.

Proof. (Sketch) We need several tools in the literature to prove this result. From a high level, since we have already shown that $\{g_t(\boldsymbol{D}_t)\}_{t\geq 1}$ converges, we only need to deduce that the limit of $g_t(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t)$ is zero. A useful result for this purpose is stated below, which is borrowed from [99].

Lemma 5.8 (Lemma 8 in [99])**.** *Let* $\{a_t\}_{t\geq 1}$, $\{b_t\}_{t\geq 1}$ *be two real sequences such that for all* t, $a_t \geq 0$, $b_t \geq 0$, $\sum_{t=1}^{\infty} a_t = \infty$, $\sum_{t=1}^{\infty} a_t b_t < \infty$, *there exists a scalar* $\mathrm{C} > 0$, *such that* $|b_{t+1} - b_t| < \mathrm{C}\cdot a_t$.

Then, $\lim_{t\to\infty} b_t = 0$.

In particular, we set $a_t = (t+1)^{-1}$ and $b_t = g_t(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t) \geq 0$. First, we verify that the sum of the infinite series $\{a_t b_t\}_{t\geq 1}$ is finite. By algebra, we obtain

$$\frac{b_t}{t+1} \leq \frac{\tilde{g}_t(\boldsymbol{D}_t) - \tilde{f}_t(\boldsymbol{D}_t)}{t+1} + \frac{q_t(\boldsymbol{D}_t)}{t+1},$$

where

$$q_t(\boldsymbol{D}_t) = \frac{1}{t}\sum_{i=1}^{t} \frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 + \frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t - \boldsymbol{M}_t\right\|_F^2.$$

Then by utilizing the uniform boundedness of $\boldsymbol{D}_t$ and $\boldsymbol{M}_t$, it is possible to derive

$$\frac{b_t}{t+1} \leq \frac{\ell(\boldsymbol{D}_t; \boldsymbol{z}_{t+1}) - \tilde{f}_t(\boldsymbol{D}_t)}{t+1} + \psi_t - \psi_{t+1} + \frac{\mathrm{C}}{2t(t+1)}$$

for some absolute constant C. As we have shown in the proof sketch of Theorem 5.5, the first term scales as $\mathcal{O}\left(1/t^{3/2}\right)$, which combined with (5.24) imply that $\sum_{t=1}^{\infty} \frac{b_t}{t+1}$ is finite.

Next, we claim that

$$|b_{t+1} - b_t| \leq \frac{\mathrm{C}_0}{t+1},$$

for some absolute constant C_0. The follows heavily from Proposition 5.3 where we showed that all the variables are uniformly bounded, and hence all involved functions evaluated at these solutions are bounded from above. To be more concrete, using triangle inequality we get

$$\begin{aligned} |b_{t+1} - b_t| \leq\ & |g_{t+1}(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_{t+1})| + |f_{t+1}(\boldsymbol{D}_{t+1}) - f_t(\boldsymbol{D}_{t+1})| \\ & + |g_t(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t)| + |f_t(\boldsymbol{D}_{t+1}) - f_t(\boldsymbol{D}_t)|. \end{aligned}$$

The first two terms on the right-hand side are upper bounded by $\mathcal{O}(1/t)$ due to uniform boundedness. For the last two terms, we utilize the fact that $g_t(\boldsymbol{D})$ and $f_t(\boldsymbol{D})$ are both Lipschitz to show that they are bounded by $\mathcal{O}\left(\left\|\boldsymbol{D}_{t+1} - \boldsymbol{D}_t\right\|_F\right)$ from above. The following proposition illustrates that the variation of $\boldsymbol{D}_{t+1}$ and $\boldsymbol{D}_t$ vanishes with the rate $\mathcal{O}(1/t)$, whose proof can be found in

Appendix 5.B.6.

Proposition 5.9. *Assume* $(A1)$ *and* $(A2)$. *Further suppose that* λ_1 *and* λ_2 *do not grow with* t. *Let* $\{\boldsymbol{D}_t\}_{t\geq 1}$ *be the basis sequence produced by Algorithm 7. Then,*

$$\|\boldsymbol{D}_{t+1} - \boldsymbol{D}_t\|_F = O\left(\frac{1}{t}\right).$$

Thus, in allusion to Lemma 5.8, we complete the proof of Theorem 5.6. The corollary follows immediately because the central limit theorem asserts that $\sqrt{t}(f(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t))$ is upper bounded. □

Finally, we show that asymptotically, $\boldsymbol{D}_t$ acts as a stationary point of the expected loss function.

Theorem 5.10. *Assume* $(A1)$, $(A2)$, $(P1)$ *and* $(P2)$. *Let* $\{\boldsymbol{D}_t\}_{t=1}^{\infty}$ *be the sequence of optimal bases produced by Algorithm 7. Then, the sequence converges to a stationary point of the expected loss function* $f(\boldsymbol{D})$ *when* t *goes to infinity.*

5.4 Experiments

This section gives empirical evidence that our algorithm OLRSC is fast and robust. We demonstrate by simulations that the solution is good enough. In fact, we find that after revealing all the data points, OLRSC is able to recover the true subspace even with grossly corrupted entries. Thus, it is interesting to study the statistical performance of OLRSC in the future work. We also illustrate that OLRSC is orders of magnitude faster than batch methods such as LRR and SSC.

5.4.1 Settings

Before presenting the empirical results, we first introduce the universal settings used throughout the section.

Baselines

For the subspace recovery task, we compare our algorithm with ORPCA [59], LRR [92] and PCP [34]. For the subspace clustering task, we choose ORPCA, LRR and SSC [56] as the compet-

itive baselines. Recently, [91] improved the vanilla LRR by utilizing some low-rank matrix for $\boldsymbol{Y}$. We denote this variant of LRR by LRR2 and accordingly, our algorithm equipped with such a basis dictionary $\boldsymbol{Y}$ is denoted as OLRSC2.

Evaluation Metric

We evaluate the fitness of the recovered subspaces $\boldsymbol{D}$ (with each column being normalized) and the ground truth $\boldsymbol{L}$ by the Expressed Variance (EV) [153]:

$$\mathrm{EV}(\boldsymbol{D}, \boldsymbol{L}) := \frac{\mathrm{Tr}\left(\boldsymbol{D}\boldsymbol{D}^\top \boldsymbol{L}\boldsymbol{L}^\top\right)}{\mathrm{Tr}\left(\boldsymbol{L}\boldsymbol{L}^\top\right)}. \tag{5.25}$$

The value of EV is between 0 and 1, and a higher value means better recovery (EV $= 1$ means exact recovery).

The performance of subspace clustering is measured by clustering accuracy which is provided in the SSC toolkit. Its value also ranges in the interval $[0, 1]$, and a higher value indicates a more accurate clustering.

Parameters

We set $\lambda_1 = 1$, $\lambda_2 = 1/\sqrt{d}$ and $\lambda_3 = \sqrt{t/d}$, where t is the iteration counter. Note that the parameter settings satisfy $(P1)$ and $(P2)$. In particular, $\lim_{t\to\infty} \lambda_3/t = 0$. We follow the default parameter setting for the baselines.

5.4.2 Subspace Recovery

Simulation Data

We use 4 disjoint subspaces $\{\mathcal{S}_k\}_{k=1}^4 \subset \mathbb{R}^d$, whose bases are denoted by $\{\boldsymbol{L}_k\}_{k=1}^4 \in \mathbb{R}^{d\times r_k}$. The clean data matrix $\bar{\boldsymbol{Z}}_k \in \mathcal{S}_k$ is then produced by $\bar{\boldsymbol{Z}}_k = \boldsymbol{L}_k \boldsymbol{R}_k^\top$, where $\boldsymbol{R}_k \in \mathbb{R}^{n_k\times r_k}$. The entries of $\boldsymbol{L}_k$'s and $\boldsymbol{R}_k$'s are sampled i.i.d. from the normal distribution. Finally, the observed data matrix $\boldsymbol{Z}$ is generated by $\boldsymbol{Z} = \bar{\boldsymbol{Z}} + \boldsymbol{E}$, where $\bar{\boldsymbol{Z}}$ is the column-wise concatenation of $\bar{\boldsymbol{Z}}_k$'s followed by a random permutation, $\boldsymbol{E}$ is the sparse corruption whose ρ fraction entries are non-zero and follow an i.i.d. uniform distribution over $[-2, 2]$. We independently conduct each experiment 10 times and report the averaged results.

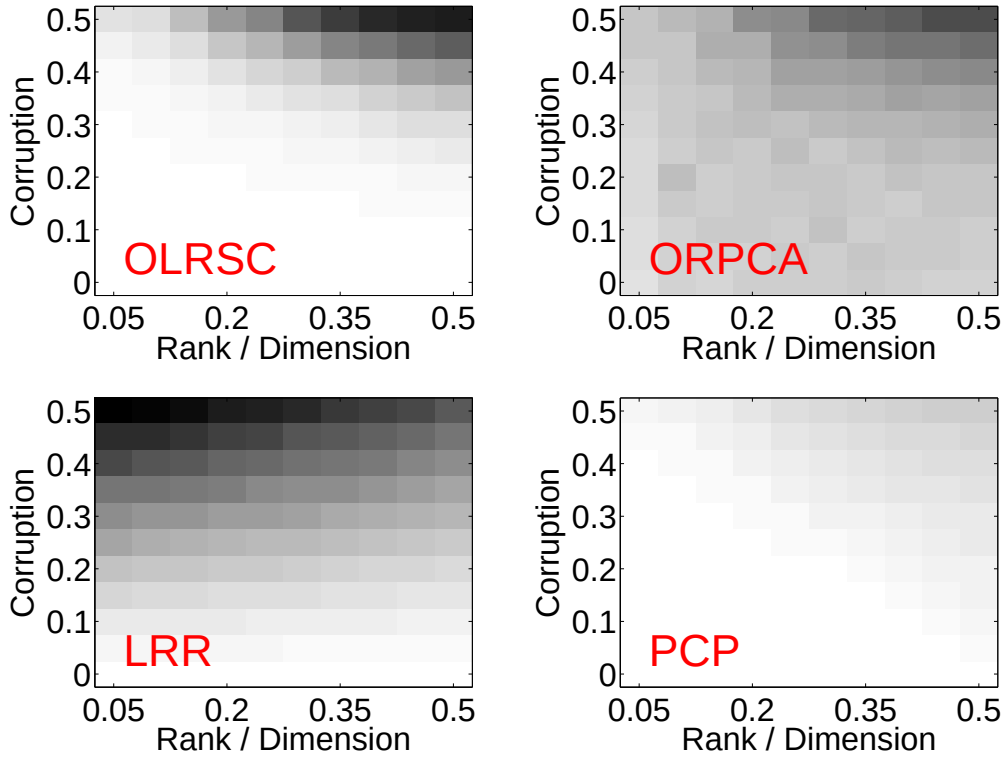


Figure 5.1: **Subspace recovery under different intrinsic dimensions and corruptions.** Brighter is better.

Robustness

We illustrate by simulation results that OLRSC can effectively recover the underlying subspaces, confirming that $\boldsymbol{D}_t$ converges to the union of subspaces. For the two online algorithms OLRSC and ORPCA, We compute the EV after revealing all the samples. We examine the performance under different intrinsic dimension r_k's and corruption ρ. To be more detailed, the r_k's are varied from $0.01d$ to $0.1d$ with a step size $0.01d$, and the ρ is from 0 to 0.5, with a step size 0.05.

The results are presented in Figure 5.1. The most intriguing observation is that OLRSC as an online algorithm outperforms its batch counterpart LRR! Such improvement may come from the explicit modeling for the basis, which makes OLRSC more informative than LRR. To fully understand the rationale behind this phenomenon is an important direction for future research. Notably, OLRSC consistently beats ORPCA (an online version of PCP), in that OLRSC takes into account that the data are produced by a union of small subspaces. While PCP works well for almost all scenarios, OLRSC degrades a little when addressing difficult cases (high rank and corruption). This

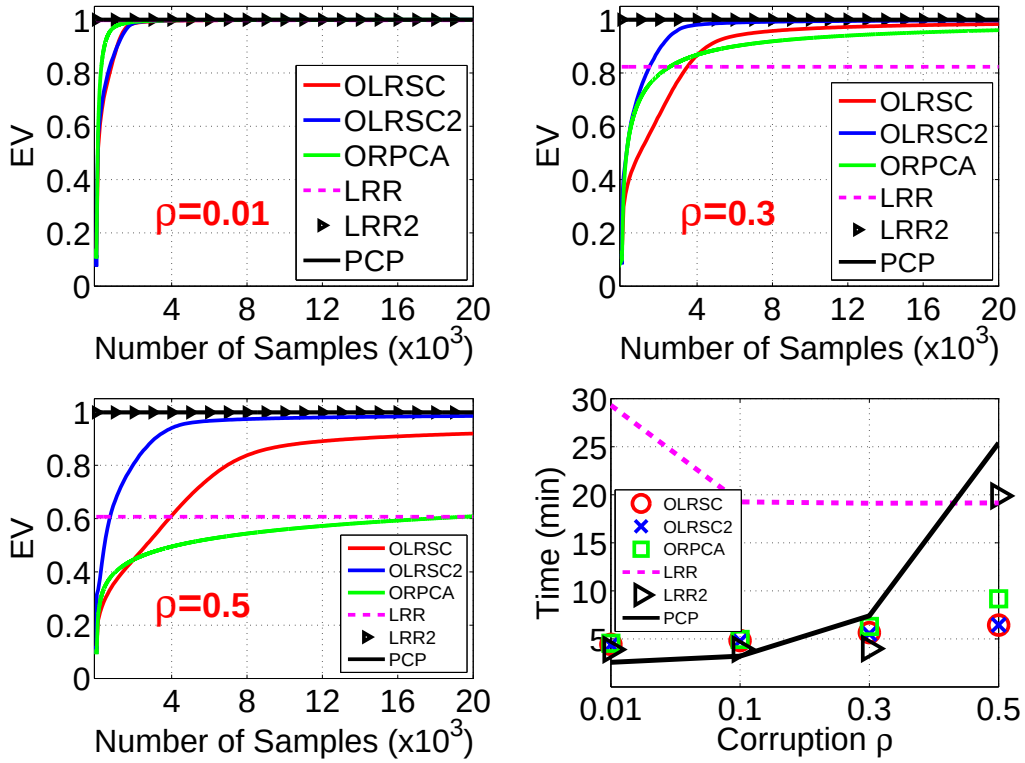


Figure 5.2: **Convergence rate and time complexity of our algorithm.**

is not surprising since Theorem 5.10 is based on asymptotic analysis and hence, we expect that OLRSC will converge to the true subspace after acquiring more samples.

Convergence Rate and Time Complexity

Now we test on a large data set to show that our algorithm usually converges to the true subspace faster than ORPCA. We plot the EV curve against the number of samples in Figure 5.2. Firstly, when equipped with a proper matrix $\boldsymbol{Y}$, OLRSC2 and LRR2 can always produce an exact recovery of the subspace as PCP does. When using the data set itself for $\boldsymbol{Y}$, OLRSC still converges to a favorable point after revealing all the samples. Compared to ORPCA, OLRSC is more robust and converges much faster for hard cases (see, e.g., $\rho = 0.5$). Again, we note that in such hard cases, OLRSC outperforms LRR, which agrees with the observation in Figure 5.1.

We also illustrate the time complexity of the algorithms in the last panel of Figure 5.2. In short, our algorithms (OLRSC and OLRSC2) admit the lowest computational complexity for all cases. One may argue that PCP spends slightly less time than ours for a small ρ (0.01 and 0.1). However,

we remark here that PCP utilizes a highly optimized C++ toolkit to boost computation while our algorithms are fully written in Matlab. We believe that ours will work more efficiently if properly optimized by, e.g., the blas routine. Another important message conveyed by the figure is that, OLRSC is always being orders of magnitude computationally more efficient than the batch method LRR, as well as producing comparable or even better solution.

5.4.3 Subspace Clustering

Datasets

We examine the performance of subspace clustering on 5 realistic databases shown in Table 5.1, which can be downloaded from the LibSVM website. For MNIST, We randomly select 20 thousands samples to form MNIST-20K since we find it time consuming to run the batch methods on the entire database.

Table 5.1: **Datasets for subspace clustering.**

	#classes	#samples	#features
Mushrooms	2	8124	112
DNA	3	3186	180
Protein	3	24,387	357
USPS	10	9298	256
MNIST-20K	10	20,000	784

Standard Clustering Pipeline

In order to focus on the solution quality of different algorithms, we follow the standard pipeline which feeds $\boldsymbol{X}$ to a spectral clustering algorithm [109]. To this end, we collect all the $\boldsymbol{u}$'s and $\boldsymbol{v}$'s produced by OLRSC to form the representation matrix $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top$. For ORPCA, we use $\boldsymbol{R}_0\boldsymbol{R}_0^\top$ as the similarity matrix [92], where $\boldsymbol{R}_0$ is the row space of $\boldsymbol{Z}_0 = \boldsymbol{L}_0\boldsymbol{\Sigma}_0\boldsymbol{R}_0^\top$ and $\boldsymbol{Z}_0$ is the clean matrix recovered by ORPCA. We run our algorithm and ORPCA with 2 epochs so as to refine the coefficients (i.e., $\boldsymbol{U}$ and $\boldsymbol{V}$ in ours and R_0 in ORPCA). Note that for subspace clustering, this step is essential because the initial guess of $\boldsymbol{D}$ results in bad solutions of the coefficients at the beginning.

Fully Online Pipeline

As we discussed in Section 5.2.2, the (optional) spectral clustering procedure needs the similarity matrix $\boldsymbol{X}$, making the memory proportional to n^2. To tackle this issue, we proposed a fully online scheme where the key idea is performing k-means on $\boldsymbol{V}$. Here, we examine the efficacy of this variant, which is called OLRSC-F.

Results

The results are recorded in Table 5.2, where the time cost of spectral clustering or k-means is not included so we can focus on comparing the efficiency of the algorithms themselves. Also note that we use the data set itself as the dictionary $\boldsymbol{Y}$ because we find that an alternative choice of $\boldsymbol{Y}$ does not help too much on this task. For OLRSC and ORPCA, they require an estimation on the true rank. Here, we use $5k$ as such estimation where k is the number of classes of a data set. Our algorithm significantly outperforms the two state-of-the-art methods LRR and SSC both in terms of accuracy and efficiency. One may argue that SSC is slightly better than OLRSC on Protein. Yet, it spends 1 hour while OLRSC only costs 25 seconds. Hence, SSC is not practical. Compared to ORPCA, OLRSC always identifies more correct samples as well as consumes comparable running time. For example, on the USPS data set, OLRSC achieves the accuracy of 65.95% while that of ORPCA is 55.7%. Regarding the running time, OLRSC uses only 7 seconds more than ORPCA – same order of computational complexity, which agrees with the qualitative analysis in Section 5.2.2 and the one in [59].

More interestingly, it shows that the k-means alternative (OLRSC-F) usually outperforms the spectral clustering pipeline. This suggests that perhaps for *robust* subspace clustering formulations, the simple k-means paradigm suffices to guarantee an appealing result. On the other hand, we report the running time of spectral clustering and k-means in Table 5.3. As expected, since spectral clustering computes SVD for an n-by-n similarity matrix, it is quite slow. In fact, it sometimes dominates the running time of the whole pipeline. In contrast, k-means is extremely fast and scalable, as it can be implemented in online fashion.

Table 5.2: **Clustering accuracy (%) and computational time (seconds in default).** For each data set, the first row indicates the accuracy and the second row the running time.

	OLRSC	OLRSC-F	ORPCA	LRR	SSC
Mush-rooms	85.09 8.78	**89.36** 8.78	65.26 8.30	58.44 46.82	54.16 32 min
DNA	67.11 2.58	**83.08** 2.58	53.11 2.09	44.01 23.67	52.23 3 min
Protein	43.30 24.66	43.94 24.66	40.22 22.90	40.31 921.58	**44.27** 65 min
USPS	65.95 33.93	**70.29** 33.93	55.70 27.01	52.98 257.25	47.58 50 min
MNIST-20K	**57.74** 129	55.50 129	54.10 121	55.23 32 min	43.91 7 hours

Table 5.3: **Time cost (in seconds) of spectral clustering and k-means.**

	Mushrooms	DNA	Protein	USPS	MNIST-20K
Spectral	295	18	7567	482	4402
k-means	2	6	5	19	91

5.5 Conclusion

In this chapter, we have proposed an online algorithm termed OLRSC for subspace clustering, which dramatically reduces the memory cost of LRR from $\mathcal{O}\left(n^2\right)$ to $\mathcal{O}\left(dr\right)$. One of the key techniques is an explicit basis modeling, which essentially renders the model more informative than LRR. Another important component is a non-convex reformulation of the nuclear norm. Combining these techniques allows OLRSC to simultaneously recover the union of the subspaces, identify the possible corruptions and perform subspace clustering. We have also established the theoretical guarantee that solutions produced by our algorithm converge to a stationary point of the expected loss function. Moreover, we have analyzed the time complexity and empirically demonstrated that our algorithm is computationally very efficient compared to competing baselines. Our extensive experimental study on synthetic and realistic data sets also illustrates the robustness of OLRSC. In a nutshell, OLRSC is an appealing algorithm in all three worlds: memory cost, computation and robustness.

5.A Algorithm Details

Algorithm 8 Solving $\boldsymbol{v}$ and $\boldsymbol{e}$

Require: $\boldsymbol{D} \in \mathbb{R}^{d\times r}$, $\boldsymbol{z} \in \mathbb{R}^d$, parameters $\lambda_1 > 0$ and $\lambda_2 > 0$.
Ensure: Optimal $\boldsymbol{v}$ and $\boldsymbol{e}$.

1: Set $\boldsymbol{e} = \boldsymbol{0}$.
2: **repeat**
3: Update $\boldsymbol{v}$:

$$\boldsymbol{v} = (\boldsymbol{D}^\top \boldsymbol{D} + \frac{1}{\lambda_1}\boldsymbol{I})^{-1}\boldsymbol{D}^\top(\boldsymbol{z} - \boldsymbol{e}).$$

4: Update $\boldsymbol{e}$:

$$\boldsymbol{e} = \mathcal{S}_{\lambda_2/\lambda_1}[\boldsymbol{z} - \boldsymbol{D}\boldsymbol{v}].$$

5: **until** convergence

Algorithm 9 Solving $\boldsymbol{D}$

Require: $\boldsymbol{D} \in \mathbb{R}^{d\times r}$ in the previous iteration, accumulation matrix $\boldsymbol{M}$, $\boldsymbol{A}$ and $\boldsymbol{B}$, parameters $\lambda_1 > 0$ and $\lambda_3 > 0$.
Ensure: Optimal $\boldsymbol{D}$ (updated).

1: Denote $\widehat{\boldsymbol{A}} = \lambda_1\boldsymbol{A} + \lambda_3\boldsymbol{I}$ and $\widehat{\boldsymbol{B}} = \lambda_1\boldsymbol{B} + \lambda_3\boldsymbol{M}$.
2: **repeat**
3: **for** $j = 1$ to r **do**
4: Update the jth column of $\boldsymbol{D}$:

$$\boldsymbol{d}_j \leftarrow \boldsymbol{d}_j - \frac{1}{\widehat{a}_{jj}}\left(\boldsymbol{D}\widehat{\boldsymbol{a}}_j - \widehat{\boldsymbol{b}}_j\right)$$

5: **end for**
6: **until** convergence

For Algorithm 8, we set a threshold $\epsilon = 10^{-3}$. Let $\{\boldsymbol{v}', \boldsymbol{e}'\}$ and $\{\boldsymbol{v}'', \boldsymbol{e}''\}$ be the two consecutive iterates. If the maximum of $\|\boldsymbol{v}' - \boldsymbol{v}''\|/\|\boldsymbol{v}'\|$ and $\|\boldsymbol{e}' - \boldsymbol{e}''\|/\|\boldsymbol{e}'\|$ is less than ϵ, then we stop Algorithm 8.

For Algorithm 9, we observe that a one-pass update on the dictionary $\boldsymbol{D}$ is enough for the final convergence of $\boldsymbol{D}$, as we showed in the experiments. This is also observed in [99].

5.B Proofs

5.B.1 Technical Lemmas

We need several technical lemmas for our proof.

Lemma 5.11 (Corollary of Thm. 4.1 in [21])**.** *Let $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$. Suppose that for all $\boldsymbol{x} \in \mathbb{R}^p$ the function $f(\boldsymbol{x}, \cdot)$ is differentiable, and that f and $\nabla_{\boldsymbol{u}} f(\boldsymbol{x}, \boldsymbol{u})$ are continuous on $\mathbb{R}^p \times \mathbb{R}^q$. Let $\boldsymbol{v}(\boldsymbol{u})$ be the optimal value function $\boldsymbol{v}(\boldsymbol{u}) = \min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x}, \boldsymbol{u})$, where $\mathcal{C}$ is a compact subset of $\mathbb{R}^p$. Then $\boldsymbol{v}(\boldsymbol{u})$ is directionally differentiable. Furthermore, if for $\boldsymbol{u}_0 \in \mathbb{R}^q$, $f(\cdot, \boldsymbol{u}_0)$ has unique minimizer $\boldsymbol{x}_0$ then $\boldsymbol{v}(\boldsymbol{u})$ is differentiable in $\boldsymbol{u}_0$ and $\nabla_{\boldsymbol{u}} \boldsymbol{v}(\boldsymbol{u}_0) = \nabla_{\boldsymbol{u}} f(\boldsymbol{x}_0, \boldsymbol{u}_0)$.*

Lemma 5.12 (Corollary of Donsker theorem [144])**.** *Let $F = \{f_\theta : \mathcal{X} \to \mathbb{R}, \theta \in \Theta\}$ be a set of measurable functions indexed by a bounded subset Θ of $\mathbb{R}^d$. Suppose that there exists a constant K such that*

$$|f_{\theta_1}(x) - f_{\theta_2}(x)| \leq K \|\theta_1 - \theta_2\|,$$

for every θ_1 and θ_2 in Θ and x in $\mathcal{X}$. Then, F is P-Donsker. For any f in F, let us define $\mathbb{P}_n f$, $\mathbb{P} f$ and $\mathbb{G}_n f$ as

$$\mathbb{P}_n f = \frac{1}{n} \sum_{i=1}^{n} f(X_i),\ \mathbb{P} f = \mathbb{E}[f(X)],\ \mathbb{G}_n f = \sqrt{n}(\mathbb{P}_n f - \mathbb{P} f).$$

Let us also suppose that for all f, $\mathbb{P} f^2 < \delta^2$ and $\|f\|_\infty < \boldsymbol{M}$ and that the random elements $X_1, X_2, \cdots$ are Borel-measurable. Then, we have

$$\mathbb{E} \|\mathbb{G}\|_F = \mathcal{O}(1),$$

where $\|\mathbb{G}\|_F = \sup_{f \in F} |\mathbb{G}_n f|$.

5.B.2 Proof of Proposition 5.2

Proof. The optimal solution $\boldsymbol{U}$ for (5.7) is given by the first order optimality condition:

$$\frac{\partial \tilde{h}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{U})}{\partial \boldsymbol{U}} = \boldsymbol{U} + \lambda_3(\boldsymbol{U}\boldsymbol{Y}^\top - \boldsymbol{D}^\top)\boldsymbol{Y} = 0,$$

by which we have

$$\begin{aligned}\tilde{\boldsymbol{U}} &= \boldsymbol{D}^\top \boldsymbol{Y}\left(\frac{1}{\lambda_3}\boldsymbol{I}_n + \boldsymbol{Y}^\top\boldsymbol{Y}\right)^{-1} \\ &= \boldsymbol{D}^\top\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{Y}.\end{aligned}$$

Note that $\tilde{\boldsymbol{u}}_i$ is the ith column of $\tilde{\boldsymbol{U}}$. So for each $i \in [n]$,

$$\tilde{\boldsymbol{u}}_i = \boldsymbol{D}^\top\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{y}_i.$$

Also, we have

$$\begin{aligned}\boldsymbol{Y}\tilde{\boldsymbol{U}}^\top &= \boldsymbol{Y}\boldsymbol{Y}^\top\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{D} \\ &= \left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top - \frac{1}{\lambda_3}\boldsymbol{I}_d\right)\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{D} \\ &= \boldsymbol{D} - \frac{1}{\lambda_3}\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{D}.\end{aligned}$$

Thus,

$$h(\boldsymbol{D}; \boldsymbol{Y}) = \frac{1}{2}\left\|\boldsymbol{D}^\top\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{Y}\right\|_F^2 + \frac{1}{2\lambda_3}\left\|\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{D}\right\|_F^2.$$

□

5.B.3 Proof of Proposition 5.3

Proof. Let us consider the optimization problem of solving $\boldsymbol{v}$ and $\boldsymbol{e}$. As the trivial solution $\{\boldsymbol{v}_t', \boldsymbol{e}_t'\} = \{\mathbf{0}, \mathbf{0}\}$ is feasible, we have

$$\tilde{\ell}_1(\boldsymbol{D}_{t-1}, \boldsymbol{v}_t', \boldsymbol{e}_t'; \boldsymbol{z}_t) = \lambda_2 \left\|\boldsymbol{z}_t\right\|_1 .$$

Therefore, the optimal solution should satisfy:

$$\frac{\lambda_1}{2} \left\|\boldsymbol{z}_t - \boldsymbol{D}_{t-1}\boldsymbol{v}_t^* - \boldsymbol{e}_t^*\right\|^2 + \frac{1}{2} \left\|\boldsymbol{v}_t^*\right\|^2 + \lambda_2 \left\|\boldsymbol{e}_t^*\right\|_1 \leq \lambda_2 \left\|\boldsymbol{z}_t\right\|_1 ,$$

which implies

$$\frac{1}{2} \left\|\boldsymbol{v}_t^*\right\|^2 \leq \lambda_2 \left\|\boldsymbol{z}_t\right\|_1 , \ \lambda_2 \left\|\boldsymbol{e}_t^*\right\|_1 \leq \lambda_2 \left\|\boldsymbol{z}_t\right\|_1 .$$

Since $\boldsymbol{z}_t$ is uniformly bounded (Assumption $(A1)$) and λ_2 does not grow with t, $\boldsymbol{v}_t^*$ and $\boldsymbol{e}_t^*$ are both uniformly bounded from above.

To examine the uniform boundedness for $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$, note that

$$\begin{aligned}
\frac{1}{t}\boldsymbol{A}_t &= \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{v}_i^* (\boldsymbol{v}_i^*)^\top, \\
\frac{1}{t}\boldsymbol{B}_t &= \frac{1}{t}\sum_{i=1}^{t} (\boldsymbol{z}_i - \boldsymbol{e}_i^*) (\boldsymbol{v}_i^*)^\top.
\end{aligned}$$

Since for each i, $\boldsymbol{v}_i^*$, $\boldsymbol{e}_i^*$ and $\boldsymbol{z}_i$ are uniformly bounded from above, $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$ are uniformly upper bounded.

Now we derive the bound for $\boldsymbol{D}_t$ and $\boldsymbol{M}_t$. We inductively show that both the sequences $\{\boldsymbol{D}_t\}_{t\geq 1}$ and $\{\boldsymbol{M}_t\}_{t\geq 1}$ are uniformly bounded. First, let us denote the upper bound of $\frac{1}{t}\boldsymbol{B}_t$ by C_2, i.e.,

$$\left\|\frac{1}{t}\boldsymbol{B}_t\right\| \leq \mathrm{C}_2, \ \forall\, t \geq 1. \tag{5.26}$$

Also, Assumption $(A2)$ indicates that there exists an absolute constant C_3, such that

$$\left\| \frac{1}{t} \boldsymbol{A}_t \right\| \geq \mathrm{C}_3, \ \forall \, t \geq 1. \tag{5.27}$$

Now suppose that for all $1 \leq i \leq t-1$, it holds for some absolute constant $\mathrm{C}_1 > \mathrm{C}_2/\mathrm{C}_3$ that

$$\|\boldsymbol{D}_i\| \leq \mathrm{C}_1, \quad \|\boldsymbol{M}_i\| \leq \mathrm{C}_1.$$

Using the closed form solution of $\boldsymbol{u}_t^*$, we have

$$\begin{aligned}
\boldsymbol{M}_t &= \boldsymbol{M}_{t-1} + \boldsymbol{y}_t (\boldsymbol{u}_t^*)^\top \\
&= \boldsymbol{M}_{t-1} + \|\boldsymbol{y}_t\|^2 \left(\|\boldsymbol{y}_t\|^2 + \frac{1}{\lambda_3} \right)^{-1} (\boldsymbol{D}_{t-1} - \boldsymbol{M}_{t-1}) \\
&= \frac{\|\boldsymbol{y}_t\|^2}{\|\boldsymbol{y}_t\|^2 + \frac{1}{\lambda_3}} \boldsymbol{D}_{t-1} + \frac{\lambda_3^{-1}}{\|\boldsymbol{y}_t\|^2 + \frac{1}{\lambda_3}} \boldsymbol{M}_{t-1}.
\end{aligned}$$

Hence,

$$\|\boldsymbol{M}_t\| \leq \frac{\|\boldsymbol{y}_t\|^2}{\|\boldsymbol{y}_t\|^2 + \lambda_3^{-1}} \|\boldsymbol{D}_{t-1}\| + \frac{\lambda_3^{-1}}{\|\boldsymbol{y}_t\|^2 + \lambda_3^{-1}} \|\boldsymbol{M}_{t-1}\| \leq \mathrm{C}_1.$$

Now using the closed form solution of $\boldsymbol{D}_t$, we have

$$\boldsymbol{D}_t = (\lambda_1 \boldsymbol{B}_t + \lambda_3 \boldsymbol{M}_t)(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}_d)^{-1} = \left(\frac{\lambda_1}{t} \boldsymbol{B}_t + \frac{\lambda_3}{t} \boldsymbol{M}_t \right) \left(\frac{\lambda_1}{t} \boldsymbol{A}_t + \frac{\lambda_3}{t} \boldsymbol{I}_d \right)^{-1}.$$

Combining the above with (5.26) and (5.27) gives us

$$\|\boldsymbol{D}_t\| \leq \left(\lambda_1 \mathrm{C}_2 + \frac{\lambda_3}{t} \mathrm{C}_1 \right) \left(\lambda_1 \mathrm{C}_3 + \frac{\lambda_3}{t} \right)^{-1} = \frac{\lambda_3}{\lambda_1 \mathrm{C}_3} \frac{\mathrm{C}_1 - \mathrm{C}_2/\mathrm{C}_3}{t + \frac{\lambda_3}{\lambda_1 \mathrm{C}_3}} + \frac{\mathrm{C}_2}{\mathrm{C}_3}.$$

It turns out that the maximum of the right hand side is attained at $t = 1$ due to our earlier choice $\mathrm{C}_1 > \mathrm{C}_2/\mathrm{C}_3$. Hence, we have

$$\|\boldsymbol{D}_t\| \leq \mathrm{C}_1.$$

The induction is complete.

By examining the closed form of $\boldsymbol{u}_t^*$, and note that we have proved the uniform boundedness of $\boldsymbol{D}_t$ and $\boldsymbol{M}_t$, we conclude that $\boldsymbol{u}_t^*$ is uniformly upper bounded. □

Corollary 5.13. *Assume same conditions as in Proposition 5.3. Further suppose that λ_1 does not grow with t. Then, for all $t \geq 1$,*

1. *$\tilde{\ell}(\boldsymbol{D}_t, \boldsymbol{v}_t^*, \boldsymbol{e}_t^*; \boldsymbol{z}_t)$ and $\ell(\boldsymbol{D}_t; \boldsymbol{z}_t)$ are uniformly bounded from above.*
2. *$\frac{1}{t}\tilde{h}(\boldsymbol{D}_t, \boldsymbol{U}_{1:t}^*; \boldsymbol{Y}_{1:t})$ is uniformly upper bounded where $\boldsymbol{U}_{1:t}^* = (\boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_t^*)$ and $\boldsymbol{Y}_{1:t} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$.*
3. *The surrogate function $g_t(\boldsymbol{D}_t)$ defined in* (5.19) *is uniformly upper bounded and Lipschitz.*

Proof. To show Claim 1, we just need to examine the definition of $\tilde{\ell}(\boldsymbol{D}_t, \boldsymbol{v}_t^*, \boldsymbol{e}_t^*; \boldsymbol{z}_t)$ (see Eq. (5.6)) and notice that $\boldsymbol{z}_t$, $\boldsymbol{D}_t$, $\boldsymbol{v}_t^*$ and $\boldsymbol{e}_t^*$ are all uniformly bounded. This implies that $\tilde{\ell}(\boldsymbol{D}_t, \boldsymbol{v}_t^*, \boldsymbol{e}_t^*; \boldsymbol{z}_t)$ is uniformly bounded and so is $\ell(\boldsymbol{D}_t; \boldsymbol{z}_t)$. Likewise, we prove that $\frac{1}{t}\tilde{h}(\boldsymbol{D}_t, \boldsymbol{U}_{1:t}^*; \boldsymbol{Y}_{1:t})$ is uniformly bounded. The uniform boundedness of $g_t(\boldsymbol{D}_t)$ follows immediately.

To show that $g_t(\boldsymbol{D})$ is Lipschitz, we show that the gradient of $g_t(\boldsymbol{D})$ is uniformly bounded for all $\boldsymbol{D} \in \mathcal{D}$.

$$\begin{aligned}\|\nabla g_t(\boldsymbol{D})\|_F &= \left\|\lambda_1 \boldsymbol{D}\left(\frac{1}{t}\boldsymbol{A}_t + \frac{\lambda_3}{t}\boldsymbol{I}_d\right) - \frac{\lambda_1}{t}\boldsymbol{B}_t - \frac{\lambda_3}{t}\boldsymbol{M}_t\right\|_F \\ &\leq \lambda_1 \|\boldsymbol{D}\|_F \left(\left\|\frac{1}{t}\boldsymbol{A}_t\right\|_F + \left\|\frac{\lambda_3}{t}\boldsymbol{I}_d\right\|_F\right) + \lambda_1 \left\|\frac{1}{t}\boldsymbol{B}_t\right\|_F + \left\|\frac{\lambda_3}{t}\boldsymbol{M}_t\right\|_F.\end{aligned}$$

Notice that each term on the right hand side of the inequality is uniformly bounded from above and λ_1 does not grow with t. Thus the gradient of $g_t(\boldsymbol{D})$ is uniformly bounded, implying that $g_t(\boldsymbol{D})$ is Lipschitz. □

Proposition 5.14. *Let $\boldsymbol{D} \in \mathcal{D}$ and denote the minimizer of $\tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z})$ as:*

$$\{\boldsymbol{v}', \boldsymbol{e}'\} = \arg\min_{\boldsymbol{v},\boldsymbol{e}} \tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z}).$$

Then, the function $\ell(\boldsymbol{D};\boldsymbol{z})$ is continuously differentiable and

$$\nabla_{\boldsymbol{D}}\ell(\boldsymbol{D};\boldsymbol{z}) = (\boldsymbol{D}\boldsymbol{v}' + \boldsymbol{e}' - \boldsymbol{z})(\boldsymbol{v}')^\top.$$

Furthermore, $\ell(\boldsymbol{D};\boldsymbol{z})$ is uniformly Lipschitz.

Proof. By fixing $\boldsymbol{z}$, the function $\tilde{\ell}$ can be seen as a mapping:

$$\mathbb{R}^{r+d} \times \mathcal{D} \to \mathbb{R}$$
$$([\boldsymbol{v};\ \boldsymbol{e}], \boldsymbol{D}) \mapsto \tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z}).$$

It is easy to show that for all $[\boldsymbol{v};\ \boldsymbol{e}] \in \mathbb{R}^{r+d}$, $\tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z})$ is differentiable with respect to $\boldsymbol{D}$. Also $\tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z})$ is continuous on $\mathbb{R}^{r+d} \times \mathcal{D}$ and so is its gradient $\nabla_{\boldsymbol{D}}\tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z}) = (\boldsymbol{D}\boldsymbol{v} + \boldsymbol{e} - \boldsymbol{z})\boldsymbol{v}^\top$. For all $\boldsymbol{D} \in \mathcal{D}$, since $\tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z})$ is strongly convex w.r.t. $\boldsymbol{v}$ and $\boldsymbol{e}$, it has a unique minimizer $\{\boldsymbol{v}', \boldsymbol{e}'\}$. Thus Lemma 5.11 applies and we prove that $\ell(\boldsymbol{D};\boldsymbol{z})$ is differentiable in $\boldsymbol{D}$ and

$$\nabla_{\boldsymbol{D}}\ell(\boldsymbol{D};\boldsymbol{z}) = (\boldsymbol{D}\boldsymbol{v}' + \boldsymbol{e}' - \boldsymbol{z})(\boldsymbol{v}')^\top.$$

Since every term in $\nabla_{\boldsymbol{D}}\ell(\boldsymbol{D};\boldsymbol{z})$ is uniformly bounded (Assumption $(A1)$ and Proposition 5.3), we conclude that the gradient of $\ell(\boldsymbol{D};\boldsymbol{z})$ is uniformly bounded, implying that $\ell(\boldsymbol{z}, \boldsymbol{D})$ is uniformly Lipschitz w.r.t. $\boldsymbol{D}$. □

Corollary 5.15. *Let $f_t(\boldsymbol{D})$ be the empirical loss function defined in* (5.9)*. Then $f_t(\boldsymbol{D})$ is uniformly bounded from above and Lipschitz for all $t \geq 1$ and $\boldsymbol{D} \in \mathcal{D}$.*

Proof. Since $\ell(\boldsymbol{D};\boldsymbol{z})$ can be uniformly bounded (Corollary 5.13), we only need to show that $\frac{1}{t}h(\boldsymbol{D};\boldsymbol{Y})$ is uniformly bounded, where $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_t)$. Note that we have derived the form for $h(\boldsymbol{D};\boldsymbol{Y})$ in Proposition 5.2:

$$\frac{1}{t}h(\boldsymbol{D};\boldsymbol{Y}) = \frac{1}{2t}\sum_{i=1}^{t}\left\|\boldsymbol{D}^\top\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{y}_i\right\|^2 + \frac{1}{2\lambda_3 t}\left\|\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{D}\right\|_F^2.$$

Since every term in the above equation can be uniformly bounded, $\frac{1}{t}h(\boldsymbol{D};\boldsymbol{Z}_t)$ is uniformly bounded and so is $f_t(\boldsymbol{D})$.

To show that $f_t(\boldsymbol{D})$ is uniformly Lipschitz, it amounts to prove that its gradient can be uniformly bounded from above. Using Proposition 5.14, we have

$$\begin{aligned}\nabla f_t(\boldsymbol{D}) &= \frac{1}{t}\sum_{i=1}^{t}\nabla \ell(\boldsymbol{D};\boldsymbol{z}_i) + \frac{1}{t}\nabla h(\boldsymbol{D};\boldsymbol{Z}_t) \\ &= \frac{1}{t}\sum_{i=1}^{t}(\boldsymbol{D}\boldsymbol{v}_i^* + \boldsymbol{e}_i^* - \boldsymbol{z}_i)(\boldsymbol{v}_i^*)^\top \\ &\quad + \frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{y}_i\boldsymbol{y}_i^\top\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{D} \\ &\quad + \frac{\lambda_3}{t}\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-2}\boldsymbol{D}.\end{aligned}$$

Then the Frobenius norm of $\nabla f_t(\boldsymbol{D})$ can be bounded by:

$$\begin{aligned}\|\nabla f_t(\boldsymbol{D})\|_F \le \frac{1}{t}\sum_{i=1}^{t}\|\boldsymbol{D}\boldsymbol{v}_i^* + \boldsymbol{e}_i^* - \boldsymbol{z}_i\|\cdot\|\boldsymbol{v}_i^*\| \\ + \frac{1}{t}\sum_{i=1}^{t}\left\|\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\right\|_F^2\cdot\|\boldsymbol{y}_i\|^2\cdot\|\boldsymbol{D}\|_F \\ + \frac{\lambda_3}{t}\left\|\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\right\|_F^2\cdot\|\boldsymbol{D}\|_F.\end{aligned}$$

One can easily check that the right hand side of the inequality is uniformly bounded from above. Thus $\|\nabla f_t(\boldsymbol{D})\|_F$ is uniformly bounded, implying that $f_t(\boldsymbol{D})$ is uniformly Lipschitz. □

5.B.4 Proof of P-Donsker

Proposition 5.16. *Let* $\tilde{f}_t(\boldsymbol{D}) = \frac{1}{t}\sum_{i=1}^{t}\ell(\boldsymbol{D};\boldsymbol{z}_i)$. *Then we have*

$$\mathbb{E}[\sqrt{t}\left\|\tilde{f}_t - \mathbb{E}_{\boldsymbol{z}}[\ell(\boldsymbol{D};\boldsymbol{z})]\right\|_\infty] = \mathcal{O}(1).$$

Proof. Let us consider $\{\ell(\boldsymbol{D};\boldsymbol{z})\}$ as a set of measurable functions indexed by $\boldsymbol{D}\in\mathcal{D}$. As we showed in Proposition 5.3, $\mathcal{D}$ is a compact set. Also, we have proved that $\ell(\boldsymbol{D};\boldsymbol{z})$ is uniformly Lipschitz over $\boldsymbol{D}$ (Proposition 5.14). Thus, $\{\ell(\boldsymbol{D};\boldsymbol{z})\}$ is P-Donsker (see the definition in Lemma 5.12). Furthermore, as $\ell(\boldsymbol{D};\boldsymbol{z})$ is non-negative and its magnitude is uniformly upper bounded (Corollary 5.13), so is $\ell^2(\boldsymbol{D};\boldsymbol{z})$. Hence we have $\mathbb{E}_{\boldsymbol{z}}[\ell^2(\boldsymbol{D};\boldsymbol{z})]\le c$ for some absolute constant c. Note

that we have verified all the hypotheses in Lemma 5.12. Hence the proof is complete. □

5.B.5 Proof of Theorem 5.5

Proof. Note that $g_t(\boldsymbol{D}_t)$ can be viewed as a stochastic positive process since every term in $g_t(\boldsymbol{D}_t)$ is non-negative and the samples are drawn randomly. We define for all $t \geq 1$

$$\psi_t := g_t(\boldsymbol{D}_t).$$

To show the convergence of ψ_t, we need to bound the difference of ψ_{t+1} and ψ_t:

$$\begin{aligned}
\psi_{t+1} - \psi_t &= g_{t+1}(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t) \\
&= g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + g_{t+1}(\boldsymbol{D}_t) - g_t(\boldsymbol{D}_t) \\
&= g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{1}{t+1}\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - \frac{1}{t+1}\tilde{g}_t(\boldsymbol{D}_t) \\
&\quad + \left[\frac{1}{t+1} \sum_{i=1}^{t+1} \frac{1}{2} \|\boldsymbol{u}_i^*\|^2 + \frac{\lambda_3}{2(t+1)} \|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\|_F^2 \right. \\
&\quad \left. - \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} \|\boldsymbol{u}_i^*\|^2 - \frac{\lambda_3}{2t} \|\boldsymbol{D}_t - \boldsymbol{M}_t\|_F^2 \right] \\
&= g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{\tilde{f}_t(\boldsymbol{D}_t) - \tilde{g}_t(\boldsymbol{D}_t)}{t+1} + \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - \tilde{f}_t(\boldsymbol{D}_t)}{t+1} \\
&\quad + \left[\frac{1}{t+1} \sum_{i=1}^{t+1} \frac{1}{2} \|\boldsymbol{u}_i^*\|^2 + \frac{\lambda_3}{2(t+1)} \|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\|_F^2 \right. \\
&\quad \left. - \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} \|\boldsymbol{u}_i^*\|^2 - \frac{\lambda_3}{2t} \|\boldsymbol{D}_t - \boldsymbol{M}_t\|_F^2 \right]. \qquad (5.28)
\end{aligned}$$

Here,

$$\tilde{g}_t(\boldsymbol{D}_t) = \frac{1}{t} \sum_{i=1}^{t} \tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}_i, \boldsymbol{e}_i; \boldsymbol{z}_i),$$

and

$$\tilde{f}_t(\boldsymbol{D}_t) = \frac{1}{t} \sum_{i=1}^{t} \ell(\boldsymbol{D}_t; \boldsymbol{z}_i).$$

First, we bound the four terms in the brackets of (5.28). We have

$$\frac{1}{t+1}\sum_{i=1}^{t+1}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2-\frac{1}{t}\sum_{i=1}^{t}\left\|\boldsymbol{u}_i^*\right\|^2=\frac{-1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2+\frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}^*\right\|^2$$
$$\leq\frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}^*\right\|^2, \tag{5.29}$$

and

$$\begin{aligned}
&\frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{D}_t-\boldsymbol{M}_{t+1}\right\|_F^2-\frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t-\boldsymbol{M}_t\right\|_F^2\\
=&\frac{-\lambda_3}{2t(t+1)}\left\|\boldsymbol{D}_t-\boldsymbol{M}_t\right\|_F^2+\frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{z}_{t+1}(\boldsymbol{u}_{t+1}^*)^\top\right\|_F^2\\
&-\frac{\lambda_3}{t+1}\operatorname{Tr}\left((\boldsymbol{D}_t-\boldsymbol{M}_t)^\top\boldsymbol{z}_{t+1}(\boldsymbol{u}_{t+1}^*)^\top\right)\\
=&\frac{-\lambda_3}{2t(t+1)}\left\|\boldsymbol{D}_t-\boldsymbol{M}_t\right\|_F^2+\frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{z}_{t+1}(\boldsymbol{u}_{t+1}^*)^\top\right\|_F^2\\
&-\frac{\lambda_3}{t+1}\left(\left\|\boldsymbol{z}_{t+1}\right\|^2+\frac{1}{\lambda_3}\right)\left\|\boldsymbol{u}_{t+1}^*\right\|^2\\
\leq&\frac{1}{t+1}\left(\frac{\lambda_3}{2}\left\|\boldsymbol{z}_{t+1}(\boldsymbol{u}_{t+1}^*)^\top\right\|_F^2-\left(\lambda_3\left\|\boldsymbol{z}_{t+1}\right\|^2+1\right)\left\|\boldsymbol{u}_{t+1}^*\right\|^2\right)\\
\leq&\frac{1}{t+1}\left(-\frac{\lambda_3}{2}\left\|\boldsymbol{z}_{t+1}\right\|^2\left\|\boldsymbol{u}_{t+1}^*\right\|^2-\left\|\boldsymbol{u}_{t+1}^*\right\|^2\right),
\end{aligned}\tag{5.30}$$

where the first equality is derived by the fact that $\boldsymbol{M}_{t+1}=\boldsymbol{M}_t+\boldsymbol{z}_{t+1}(\boldsymbol{u}_{t+1}^*)^\top$, and the second equality is derived by the closed form solution of $\boldsymbol{u}_{t+1}^*$ (see (5.18)).

Combining (5.29) and (5.30), we know that

$$\begin{aligned}
&\frac{1}{t+1}\sum_{i=1}^{t+1}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2-\frac{1}{t}\sum_{i=1}^{t}\left\|\boldsymbol{u}_i^*\right\|^2\\
&+\frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{D}_t-\boldsymbol{M}_{t+1}\right\|_F^2-\frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t-\boldsymbol{M}_t\right\|_F^2\\
\leq&\frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}^*\right\|^2+\frac{1}{t+1}\Big(-\frac{\lambda_3}{2}\left\|\boldsymbol{z}_{t+1}\right\|^2\left\|\boldsymbol{u}_{t+1}^*\right\|^2-\left\|\boldsymbol{u}_{t+1}^*\right\|^2\Big)\\
=&\frac{1}{t+1}\left(-\frac{\lambda_3}{2}\left\|\boldsymbol{z}_{t+1}\right\|^2\left\|\boldsymbol{u}_{t+1}^*\right\|^2-\frac{1}{2}\left\|\boldsymbol{u}_{t+1}^*\right\|^2\right)\leq 0.
\end{aligned}$$

Therefore,

$$\begin{aligned}\psi_{t+1} - \psi_t &\leq g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{1}{t+1}\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - \frac{1}{t+1}\tilde{g}_t(\boldsymbol{D}_t) \\ &= g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{\tilde{f}_t(\boldsymbol{D}_t) - \tilde{g}_t(\boldsymbol{D}_t)}{t+1} + \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - \tilde{f}_t(\boldsymbol{D}_t)}{t+1} \\ &\leq \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - \tilde{f}_t(\boldsymbol{D}_t)}{t+1},\end{aligned}$$

where the last inequality holds because $\boldsymbol{D}_{t+1}$ is the minimizer of $g_{t+1}(\boldsymbol{D})$ and $\tilde{g}_t(\boldsymbol{D})$ is a surrogate function of $\tilde{f}_t(\boldsymbol{D})$.

Let $\mathcal{F}_t$ be the filtration of the past information. We take the expectation on the above equation conditional on $\mathcal{F}_t$:

$$\begin{aligned}\mathbb{E}[\psi_{t+1} - \psi_t \mid \mathcal{F}_t] &\leq \frac{\mathbb{E}[\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) \mid \mathcal{F}_t] - \tilde{f}_t(\boldsymbol{D}_t)}{t+1} \\ &\leq \frac{f(\boldsymbol{D}_t) - \tilde{f}_t(\boldsymbol{D}_t)}{t+1} \\ &\leq \frac{\left\| f - \tilde{f}_t \right\|_\infty}{t+1}.\end{aligned}$$

From Proposition 5.16, we know

$$\mathbb{E}\Big[\left\| f - \tilde{f}_t \right\|_\infty \Big] = O\left(\frac{1}{\sqrt{t}}\right).$$

Thus,

$$\mathbb{E}\big[\mathbb{E}[\psi_{t+1} - \psi_t \mid \mathcal{F}_t]^+\big] = \mathbb{E}\big[\max\{\mathbb{E}[\psi_{t+1} - \psi_t \mid \mathcal{F}_t], 0\}\big] \leq \frac{c}{\sqrt{t}(t+1)}, \tag{5.31}$$

where c is some constant.

Now let us define the index set

$$\mathcal{T} = \Big\{ t : \ t \geq 1, \mathbb{E}\big[\psi_{t+1} - \psi_t \mid \mathcal{F}_t\big] > 0 \Big\},$$

and the indicator function

$$\delta_t = \begin{cases} 1, & \text{if } t \in \mathcal{T}, \\ 0, & \text{otherwise.} \end{cases}$$

It follows that

$$\begin{aligned} \sum_{t=1}^{\infty} \mathbb{E}[\delta_t(\psi_{t+1} - \psi_t)] &= \sum_{t \in \mathcal{T}} \mathbb{E}[\psi_{t+1} - \psi_t] \\ &= \sum_{t \in \mathcal{T}} \mathbb{E}[\mathbb{E}[\psi_{t+1} - \psi_t \mid \mathcal{F}_t]] \\ &= \sum_{t=1}^{\infty} \mathbb{E}[\mathbb{E}[\psi_{t+1} - \psi_t \mid \mathcal{F}_t]^+] \\ &\leq +\infty, \end{aligned}$$

where the last inequality holds in view of (5.31).

Thus, Lemma 5.4 applies. That is, $\{g_t(\boldsymbol{D}_t)\}_{t \geq 1}$ is a quasi-martingale and converges almost surely. In addition,

$$\sum_{t=1}^{\infty} \left| \mathbb{E}\left[\psi_{t+1} - \psi_t \mid \mathcal{F}_t\right] \right| < +\infty, \ a.s.$$

□

5.B.6 Proof of Proposition 5.9

Proof. According the strong convexity of $g_t(\boldsymbol{D})$ (Assumption $(A2)$), we have,

$$g_t(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t) \geq \frac{\beta_0}{2} \left\| \boldsymbol{D}_{t+1} - \boldsymbol{D}_t \right\|_F^2, \tag{5.32}$$

On the other hand,

$$\begin{aligned} g_t(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t) &= g_t(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_{t+1}) + g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) \\ &\quad + g_{t+1}(\boldsymbol{D}_t) - g_t(\boldsymbol{D}_t) \\ &\leq g_t(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_{t+1}) + g_{t+1}(\boldsymbol{D}_t) - g_t(\boldsymbol{D}_t). \end{aligned} \tag{5.33}$$

Note that the inequality is derived by the fact that $g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) \leq 0$, as $\boldsymbol{D}_{t+1}$ is the minimizer of $g_{t+1}(\boldsymbol{D})$. Let

$$G_t(\boldsymbol{D}) = g_t(\boldsymbol{D}) - g_{t+1}(\boldsymbol{D}). \tag{5.34}$$

By a simple calculation, we obtain the gradient of $G_t(\boldsymbol{D})$:

$$\begin{aligned}
\nabla G_t(\boldsymbol{D}) &= \nabla g_t(\boldsymbol{D}) - \nabla g_{t+1}(\boldsymbol{D}) \\
&= \frac{1}{t}\Big[\boldsymbol{D}\left(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}_d\right) - \left(\lambda_1 \boldsymbol{B}_t + \lambda_3 \boldsymbol{M}_t\right)\Big] \\
&\quad - \frac{1}{t+1}\Big[\boldsymbol{D}(\lambda_1 A_{t+1} + \lambda_3 \boldsymbol{I}_d) - (\lambda_1 B_{t+1} + \lambda_3 \boldsymbol{M}_{t+1})\Big] \\
&= \frac{1}{t}\Bigg[\boldsymbol{D}\left(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}_d - \frac{\lambda_1 t}{t+1} A_{t+1} - \frac{\lambda_3 t}{t+1}\boldsymbol{I}_d\right) \\
&\quad + \frac{\lambda_1 t}{t+1} B_{t+1} - \lambda_1 \boldsymbol{B}_t + \frac{\lambda_3 t}{t+1}\boldsymbol{M}_{t+1} - \lambda_3 \boldsymbol{M}_t\Bigg] \\
&= \frac{1}{t}\Bigg[\boldsymbol{D}\left(\frac{\lambda_1}{t+1} A_{t+1} - \lambda_1 \boldsymbol{v}_{t+1}\boldsymbol{v}_{t+1}^\top + \frac{\lambda_3}{t+1}\boldsymbol{I}_d\right) \\
&\quad + \lambda_1(\boldsymbol{z}_{t+1} - \boldsymbol{e}_{t+1})\boldsymbol{v}_{t+1}^\top - \frac{\lambda_1}{t+1} B_{t+1} + \lambda_3 \boldsymbol{z}_{t+1}\boldsymbol{u}_{t+1}^\top - \frac{\lambda_3}{t+1}\boldsymbol{M}_{t+1}\Bigg]
\end{aligned}$$

So the upper bound of the Frobenius norm of $\nabla G_t(\boldsymbol{D})$ follows immediately:

$$\begin{aligned}
&\left\|\nabla G_t(\boldsymbol{D})\right\|_F \\
&\leq \frac{1}{t}\Bigg[\left\|\boldsymbol{D}\right\|_F\left(\lambda_1\left\|\frac{A_{t+1}}{t+1}\right\|_F + \lambda_1\left\|\boldsymbol{v}_{t+1}\boldsymbol{v}_{t+1}^\top\right\|_F + \frac{\lambda_3}{t+1}\left\|\boldsymbol{I}_d\right\|_F\right) \\
&\quad + \lambda_1\left\|(\boldsymbol{z}_{t+1} - \boldsymbol{e}_{t+1})\boldsymbol{v}_{t+1}^\top\right\|_F + \lambda_1\left\|\frac{B_{t+1}}{t+1}\right\|_F + \lambda_3\left\|\boldsymbol{z}_{t+1}\boldsymbol{u}_{t+1}^\top\right\|_F + \frac{\lambda_3}{t+1}\left\|\boldsymbol{M}_{t+1}\right\|_F\Bigg] \\
&= \frac{1}{t}\Bigg[\left\|\boldsymbol{D}\right\|_F\left(\lambda_1\left\|\frac{A_{t+1}}{t+1}\right\|_F + \lambda_1\left\|\boldsymbol{v}_{t+1}\boldsymbol{v}_{t+1}^\top\right\|_F\right) + \lambda_1\left\|(\boldsymbol{z}_{t+1} - \boldsymbol{e}_{t+1})\boldsymbol{v}_{t+1}^\top\right\|_F \\
&\quad + \lambda_1\left\|\frac{B_{t+1}}{t+1}\right\|_F + \lambda_3\left\|\boldsymbol{z}_{t+1}\boldsymbol{u}_{t+1}^\top\right\|_F\Bigg] + \frac{\lambda_3}{t(t+1)}\Big[\left\|\boldsymbol{I}_d\right\|_F + \left\|\boldsymbol{M}_{t+1}\right\|_F\Big].
\end{aligned}$$

We know from Proposition 5.3 that all the terms in the above equation are uniformly bounded from above. Thus, there exist constants c_1, c_2 and c_3, such that

$$\left\|\nabla G_t(\boldsymbol{D})\right\|_F \leq \frac{1}{t}\left(c_1\left\|\boldsymbol{D}\right\|_F + c_2\right) + \frac{c_3}{t}. \tag{5.35}$$

According to the first order Taylor expansion,

$$\begin{aligned}
& G_t(\boldsymbol{D}_{t+1}) - G_t(\boldsymbol{D}_t) \\
= & \operatorname{Tr}\left(\left(\boldsymbol{D}_{t+1} - \boldsymbol{D}_t\right)^\top \nabla G_t\left(\rho\boldsymbol{D}_t + (1-\rho)\,\boldsymbol{D}_{t+1}\right)\right) \\
\leq & \left\|\boldsymbol{D}_{t+1} - \boldsymbol{D}_t\right\|_F \cdot \left\|\nabla G_t\left(\rho\boldsymbol{D}_t + (1-\rho)\,\boldsymbol{D}_{t+1}\right)\right\|_F,
\end{aligned}$$

where ρ is some scalar between 0 and 1. According to Proposition 5.3, $\boldsymbol{D}_t$ and $\boldsymbol{D}_{t+1}$ are uniformly bounded, indicating that $\rho\boldsymbol{D}_t + (1-\rho)\,\boldsymbol{D}_{t+1}$ is uniformly bounded. In view of (5.35), there exists a constant c_4, such that

$$\left\|\nabla G_t\left(\alpha\boldsymbol{D}_t + (1-\alpha)\,\boldsymbol{D}_{t+1}\right)\right\|_F \leq \frac{c_4}{t},$$

resulting in

$$G_t(\boldsymbol{D}_{t+1}) - G_t(\boldsymbol{D}_t) \leq \frac{c_4}{t} \cdot \left\|\boldsymbol{D}_{t+1} - \boldsymbol{D}_t\right\|_F.$$

Combining (5.32), (5.33) and the above equation, we have

$$\left\|\boldsymbol{D}_{t+1} - \boldsymbol{D}_t\right\|_F = \frac{2c_4}{\beta_0 t}.$$

□

5.B.7 Proof of Theorem 5.6

Proof. Recall that $\tilde{f}_t(\boldsymbol{D}) = \frac{1}{t}\sum_{i=1}^{t} \ell(\boldsymbol{D}; \boldsymbol{z}_i)$ and $\tilde{g}_t(\boldsymbol{D}) = \frac{1}{t}\sum_{i=1}^{t} \tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}_i^*, \boldsymbol{e}_i^*; \boldsymbol{z}_i)$. Define

$$
\begin{aligned}
b_t &:= g_t(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t) \\
&= \tilde{g}_t(\boldsymbol{D}_t) - \tilde{f}_t(\boldsymbol{D}_t) + \left[\frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 + \frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t - \boldsymbol{M}_t\right\|_F^2\right. \\
&\quad \left. - \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{D}_t^\top\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{z}_i\right\|^2 - \frac{1}{2\lambda_3}\left\|\left(\frac{1}{\lambda_3}\boldsymbol{I}_d + \boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{D}_t\right\|_F^2\right] \\
&= \tilde{g}_t(\boldsymbol{D}_t) - \tilde{f}_t(\boldsymbol{D}_t) + q_t(\boldsymbol{D}_t),
\end{aligned}
$$

where $q_t(\boldsymbol{D}_t)$ denotes the four terms in the brackets. Using (5.28), we have

$$
\begin{aligned}
\frac{b_t}{t+1} &= \frac{\tilde{g}_t(\boldsymbol{D}_t) - \tilde{f}_t(\boldsymbol{D}_t)}{t+1} + \frac{q_t(\boldsymbol{D}_t)}{t+1} \\
&= g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{\ell(\boldsymbol{D}_t; \boldsymbol{z}_{t+1}) - \tilde{f}_t(\boldsymbol{D}_t)}{t+1} + \psi_t - \psi_{t+1} \\
&\quad + \left[\frac{q_t(\boldsymbol{D}_t)}{t+1} + \frac{1}{t+1}\sum_{i=1}^{t+1}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 + \frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\right\|_F^2\right. \\
&\quad \left. - \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 - \frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t - \boldsymbol{M}_t\right\|_F^2\right].
\end{aligned}
$$

Note that it always holds that for some constant c,

$$
\begin{aligned}
\frac{q_t(\boldsymbol{D}_t)}{t+1} &\le \frac{1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 + \frac{\lambda_3}{2t(t+1)}\left\|\boldsymbol{D}_t - \boldsymbol{M}_t\right\|_F^2 \\
&\le \frac{1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 + \frac{c}{2t(t+1)},
\end{aligned}
$$

where the second inequality is due to the fact that $\boldsymbol{D}_t$ and $\boldsymbol{M}_t$ are both uniformly bounded (see Proposition 5.3).

On the other hand, from (5.29) we know that

$$
\frac{1}{t+1}\sum_{i=1}^{t+1}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 - \frac{1}{t}\sum_{i=1}^{t}\left\|\boldsymbol{u}_i^*\right\|^2 = \frac{-1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 + \frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}^*\right\|^2,
$$

while (5.30) implies

$$\frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{D}_t-\boldsymbol{M}_{t+1}\right\|_F^2-\frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t-\boldsymbol{M}_t\right\|_F^2\leq\frac{1}{t+1}\left(-\frac{\lambda_3}{2}\left\|\boldsymbol{z}_{t+1}\right\|^2\left\|\boldsymbol{u}_{t+1}^*\right\|^2-\left\|\boldsymbol{u}_{t+1}^*\right\|^2\right).$$

Combining these pieces, we have

$$\begin{aligned}
&\frac{q_t(\boldsymbol{D}_t)}{t+1}+\frac{1}{t+1}\sum_{i=1}^{t+1}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2+\frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{D}_t-\boldsymbol{M}_{t+1}\right\|_F^2\\
&-\frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2-\frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t-\boldsymbol{M}_t\right\|_F^2\\
\leq&\frac{c}{2t(t+1)}+\frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}^*\right\|^2+\frac{1}{t+1}\left(-\frac{\lambda_3}{2}\left\|\boldsymbol{z}_{t+1}\right\|^2\left\|\boldsymbol{u}_{t+1}^*\right\|^2-\left\|\boldsymbol{u}_{t+1}^*\right\|^2\right)\\
=&\frac{c}{2t(t+1)}-\frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}^*\right\|^2-\frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{z}_{t+1}\right\|^2\left\|\boldsymbol{u}_{t+1}^*\right\|^2\\
\leq&\frac{c}{2t(t+1)}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\frac{b_t}{t+1}&\leq g_{t+1}(\boldsymbol{D}_{t+1})-g_{t+1}(\boldsymbol{D}_t)+\frac{\ell(\boldsymbol{D}_t;\boldsymbol{z}_{t+1})-\tilde{f}_t(\boldsymbol{D}_t)}{t+1}\\
&\quad+\psi_t-\psi_{t+1}+\frac{c}{2t(t+1)}\\
&\leq\frac{\ell(\boldsymbol{D}_t;\boldsymbol{z}_{t+1})-\tilde{f}_t(\boldsymbol{D}_t)}{t+1}+\psi_t-\psi_{t+1}+\frac{c}{2t(t+1)},
\end{aligned}$$

where we use the fact that $\boldsymbol{D}_{t+1}$ minimizes $g_{t+1}(\boldsymbol{D})$ in the second inequality and we denote $\psi_t=g_t(\boldsymbol{D}_t)$. By taking the expectation over $\boldsymbol{z}$ conditional on the past filtration $\mathcal{F}_t$, we have

$$\frac{b_t}{t+1}\leq\frac{c_1}{\sqrt{t}(t+1)}+\left|\mathbb{E}[\psi_t-\psi_{t+1}\mid\mathcal{F}_t]\right|+\frac{c}{2t(t+1)},$$

which is an immediate result from Proposition 5.16. Thereby,

$$\sum_{t=1}^{\infty}\frac{b_t}{t+1}\leq\sum_{t=1}^{\infty}\frac{c_1}{\sqrt{t}(t+1)}+\sum_{t=1}^{\infty}\left|\mathbb{E}[\psi_t-\psi_{t+1}\mid\mathcal{F}_t]\right|+\sum_{t=1}^{\infty}\frac{c}{2t(t+1)}<+\infty.$$

Here, the last inequality is derived by applying (5.24).

Next, we examine the difference between b_{t+1} and b_t:

$$\begin{aligned}
|b_{t+1} - b_t| = & \left|g_{t+1}(\boldsymbol{D}_{t+1}) - f_{t+1}(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t) + f_t(\boldsymbol{D}_t)\right| \\
\leq & \left|g_{t+1}(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_{t+1})\right| + \left|g_t(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t)\right| \\
& + \left|f_{t+1}(\boldsymbol{D}_{t+1}) - f_t(\boldsymbol{D}_{t+1})\right| + \left|f_t(\boldsymbol{D}_{t+1}) - f_t(\boldsymbol{D}_t)\right|. \qquad (5.36)
\end{aligned}$$

For the first term on the right hand side, we have

$$\begin{aligned}
& \left|g_{t+1}(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_{t+1})\right| \\
= & \Big|\tilde{g}_{t+1}(\boldsymbol{D}_{t+1}) - \tilde{g}_t(\boldsymbol{D}_{t+1}) + \frac{1}{t+1}\sum_{i=1}^{t+1}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 - \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 \\
& + \frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{D}_{t+1} - \boldsymbol{M}_{t+1}\right\|_F^2 - \frac{\lambda_3}{2t}\left\|\boldsymbol{D}_{t+1} - \boldsymbol{M}_t\right\|_F^2\Big| \\
= & \Big|\tilde{g}_{t+1}(\boldsymbol{D}_{t+1}) - \tilde{g}_t(\boldsymbol{D}_{t+1}) - \frac{1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 - \frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}^*\right\|^2 \\
& - \frac{\lambda_3}{2t(t+1)}\left\|\boldsymbol{D}_{t+1} - \boldsymbol{M}_t\right\|_F^2 - \frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{z}_{t+1}(\boldsymbol{u}_{t+1}^*)^\top\right\|_F^2\Big| \\
\leq & \left|\tilde{g}_{t+1}(\boldsymbol{D}_{t+1}) - \tilde{g}_t(\boldsymbol{D}_{t+1})\right| + \frac{1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i^*\right\|^2 + \frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}^*\right\|^2 \\
& + \frac{\lambda_3}{2t(t+1)}\left\|\boldsymbol{D}_{t+1} - \boldsymbol{M}_t\right\|_F^2 + \frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{z}_{t+1}(\boldsymbol{u}_{t+1}^*)^\top\right\|_F^2 \\
\overset{\zeta_1}{\leq} & \left|\tilde{g}_{t+1}(\boldsymbol{D}_{t+1}) - \tilde{g}_t(\boldsymbol{D}_{t+1})\right| + \frac{c_1}{t+1} \\
= & \left|\frac{1}{t+1}\ell(\boldsymbol{D}_{t+1}; \boldsymbol{z}_{t+1}) - \frac{1}{t+1}\tilde{g}_t(\boldsymbol{D}_{t+1})\right| + \frac{c_1}{t+1} \\
\overset{\zeta_2}{\leq} & \frac{c_2}{t+1},
\end{aligned}$$

where c_1 and c_2 are some uniform constants. Note that ζ_1 holds because all the $\boldsymbol{u}_t^*$, $\boldsymbol{D}_{t+1}$, $\boldsymbol{M}_t$ and $\boldsymbol{z}_{t+1}$ are uniformly bounded (see Proposition 5.3), and ζ_2 holds because $\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_{t+1})$ and $\tilde{g}_t(\boldsymbol{D}_{t+1})$ are uniformly bounded (see Corollary 5.13).

For the third term on the right hand side of (5.36), we can similarly show that

$$\begin{aligned}|f_{t+1}(\boldsymbol{D}_{t+1}) - f_t(\boldsymbol{D}_{t+1})| &\leq \left|\tilde{f}_{t+1}(\boldsymbol{D}_{t+1}) - \tilde{f}_t(\boldsymbol{D}_{t+1})\right| + \frac{c_3}{t+1} \\ &= \left|\frac{1}{t+1}\ell(\boldsymbol{D}_{t+1}; \boldsymbol{z}_{t+1}) - \frac{1}{t+1}\tilde{f}_t(\boldsymbol{D}_{t+1})\right| + \frac{c_3}{t+1} \\ &\overset{\zeta_3}{\leq} \frac{c_4}{t+1},\end{aligned}$$

where c_3 and c_4 are some uniform constants, and ζ_3 holds as $\ell(\boldsymbol{D}_{t+1}; \boldsymbol{z}_{t+1})$ and $\tilde{f}_t(\boldsymbol{D}_{t+1})$ are both uniformly bounded (see Corollary 5.15).

Using Corollary 5.13 and Corollary 5.15, we know that both $g_t(\boldsymbol{D})$ and $f_t(\boldsymbol{D})$ are uniformly Lipschitz. That is, there exist uniform constants κ_1, κ_2, such that

$$\begin{aligned}|g_t(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t)| \leq \kappa_1 \left\|\boldsymbol{D}_{t+1} - \boldsymbol{D}_t\right\|_F &\overset{\zeta_4}{\leq} \frac{\kappa_3}{t+1}, \\ |f_t(\boldsymbol{D}_{t+1}) - f_t(\boldsymbol{D}_t)| \leq \kappa_2 \left\|\boldsymbol{D}_{t+1} - \boldsymbol{D}_t\right\|_F &\overset{\zeta_5}{\leq} \frac{\kappa_4}{t+1}.\end{aligned}$$

Here, ζ_4 and ζ_5 are derived by applying Proposition 5.9 and κ_3 and κ_4 are some uniform constants. Finally, we have a bound for (5.36):

$$|b_{t+1} - b_t| \leq \frac{\kappa_0}{t+1},$$

where κ_0 is some uniform constant.

By applying Lemma 5.8, we conclude that $\{b_t\}_{t\geq 1}$ converges to zero. That is,

$$\lim_{t\to+\infty} g_t(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t) = 0.$$

Since we have proved in Theorem 5.5 that $g_t(\boldsymbol{D}_t)$ converges almost surely, we conclude that $f_t(\boldsymbol{D}_t)$ converges almost surely to the same limit of $g_t(\boldsymbol{D}_t)$. □

5.B.8 Proof of Theorem 5.10

We need a technical result to prove Theorem 5.10.

Proposition 5.17. *Let $f(\boldsymbol{D})$ be the expected loss function which is defined in* (5.13)*. Then, $f(\boldsymbol{D})$*

is continuously differentiable and $\nabla f(\boldsymbol{D}) = \mathbb{E}_{\boldsymbol{z}}[\nabla_{\boldsymbol{D}} \ell(\boldsymbol{D}; \boldsymbol{z})]$. *Moreover,* $\nabla f(\boldsymbol{D})$ *is uniformly Lipschitz on* $\mathcal{D}$.

Proof. We have shown in Proposition 5.14 that $\ell(\boldsymbol{D}; \boldsymbol{z})$ is continuously differentiable, $f(\boldsymbol{D})$ is also continuously differentiable and we have $\nabla f(\boldsymbol{D}) = \mathbb{E}_{\boldsymbol{z}}[\nabla_{\boldsymbol{D}} \ell(\boldsymbol{D}; \boldsymbol{z})]$.

Next, we prove the Lipschitz of $\nabla f(\boldsymbol{D})$. Let $\boldsymbol{v}'(\boldsymbol{D}'; \boldsymbol{z}')$ and $\boldsymbol{e}'(\boldsymbol{D}'; \boldsymbol{z}')$ be the minimizer of $\tilde{\ell}(\boldsymbol{D}', \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z}')$. Since $\tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z})$ has a unique minimum and is continuous in $\boldsymbol{D}$, $\boldsymbol{v}$, $\boldsymbol{e}$ and $\boldsymbol{z}$, $\boldsymbol{v}'(\boldsymbol{D}'; \boldsymbol{z}')$ and $\boldsymbol{e}'(\boldsymbol{D}'; \boldsymbol{z}')$ is continuous in $\boldsymbol{D}$ and $\boldsymbol{z}$.

Let $\mathit{\Lambda} = \{j \mid \boldsymbol{e}'_j \neq 0\}$. According the first order optimality condition, we know that

$$\frac{\partial \tilde{\ell}(\boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}; \boldsymbol{z})}{\partial \boldsymbol{e}} = 0,$$

which implies

$$\lambda_1(\boldsymbol{z}' - \boldsymbol{D}'\boldsymbol{v}' - \boldsymbol{e}') \in \lambda_2 \partial \left\|\boldsymbol{e}'\right\|_1.$$

Hence,

$$\left|(\boldsymbol{z}' - \boldsymbol{D}'\boldsymbol{v}' - \boldsymbol{e}')_j\right| = \frac{\lambda_2}{\lambda_1},\ \forall j \in \mathit{\Lambda}.$$

Since $\boldsymbol{z} - \boldsymbol{D}\boldsymbol{v} - \boldsymbol{e}$ is continuous in $\boldsymbol{z}$ and $\boldsymbol{D}$, there exists an open neighborhood $\mathcal{V}$, such that for all $(\boldsymbol{z}'', \boldsymbol{D}'') \in \mathcal{V}$, if $j \notin \mathit{\Lambda}$, then $|(\boldsymbol{z}'' - \boldsymbol{D}''\boldsymbol{v}'' - \boldsymbol{e}'')_j| < \frac{\lambda_2}{\lambda_1}$ and $\boldsymbol{e}''_j = \boldsymbol{0}$. That is, the support set of $\boldsymbol{e}'$ will not change.

Let us denote $\boldsymbol{H} = [\boldsymbol{D}\ \boldsymbol{I}_d]$, $\boldsymbol{r} = [\boldsymbol{v}^\top\ \boldsymbol{e}^\top]^\top$ and define the function

$$\tilde{\ell}(\boldsymbol{H}_{\mathit{\Lambda}}, \boldsymbol{r}_{\mathit{\Lambda}}; \boldsymbol{z}) = \frac{\lambda_1}{2}\left\|\boldsymbol{z} - \boldsymbol{H}_{\mathit{\Lambda}}\boldsymbol{r}_{\mathit{\Lambda}}\right\|^2 + \frac{1}{2}\left\|[\boldsymbol{I}_d\ \boldsymbol{0}]\boldsymbol{r}_{\mathit{\Lambda}}\right\|^2 + \lambda_2\left\|[\boldsymbol{0}\ \boldsymbol{I}_{|\mathit{\Lambda}|}]\boldsymbol{r}_{\mathit{\Lambda}}\right\|_1.$$

Above, $\boldsymbol{r}_{\mathit{\Lambda}} = [\boldsymbol{v}^\top\ \boldsymbol{e}_{\mathit{\Lambda}}^\top]^\top$, and accordingly for $\boldsymbol{H}_{\mathit{\Lambda}}$. Since $\tilde{\ell}(\boldsymbol{D}_{\mathit{\Lambda}}, \boldsymbol{r}_{\mathit{\Lambda}}; \boldsymbol{z})$ is strongly convex with respect to $\boldsymbol{r}_{\mathit{\Lambda}}$, there exists a uniform constant κ_1, such that for all $\boldsymbol{r}''_{\mathit{\Lambda}}$,

$$\tilde{\ell}(H'_{\mathit{\Lambda}}, \boldsymbol{r}''_{\mathit{\Lambda}}; \boldsymbol{z}') - \tilde{\ell}(\boldsymbol{H}'_{\mathit{\Lambda}}, \boldsymbol{r}'_{\mathit{\Lambda}}; \boldsymbol{z}') \geq \kappa_1\left\|\boldsymbol{r}''_{\mathit{\Lambda}} - \boldsymbol{r}'_{\mathit{\Lambda}}\right\|^2 = \kappa_1\left(\left\|\boldsymbol{v}'' - \boldsymbol{v}'\right\|^2 + \left\|\boldsymbol{e}''_{\mathit{\Lambda}} - \boldsymbol{e}'_{\mathit{\Lambda}}\right\|^2\right). \quad (5.37)$$

On the other hand,

$$\begin{aligned}
&\tilde{\ell}(\boldsymbol{H}'_{\Lambda}, \boldsymbol{r}''_{\Lambda}; \boldsymbol{z}') - \tilde{\ell}(\boldsymbol{H}'_{\Lambda}, \boldsymbol{r}'_{\Lambda}; \boldsymbol{z}') \\
=& \tilde{\ell}(\boldsymbol{H}'_{\Lambda}, \boldsymbol{r}''_{\Lambda}; \boldsymbol{z}') - \tilde{\ell}(\boldsymbol{H}''_{\Lambda}, \boldsymbol{r}''_{\Lambda}; \boldsymbol{z}'') + \tilde{\ell}(\boldsymbol{H}''_{\Lambda}, \boldsymbol{r}''_{\Lambda}; \boldsymbol{z}'') - \tilde{\ell}(\boldsymbol{D}'_{\Lambda}, \boldsymbol{r}'_{\Lambda}; \boldsymbol{z}') \\
\leq& \tilde{\ell}(\boldsymbol{H}'_{\Lambda}, \boldsymbol{r}''_{\Lambda}; \boldsymbol{z}') - \tilde{\ell}(\boldsymbol{H}''_{\Lambda}, \boldsymbol{r}''_{\Lambda}; \boldsymbol{z}'') + \tilde{\ell}(\boldsymbol{H}''_{\Lambda}, \boldsymbol{r}'_{\Lambda}; \boldsymbol{z}'') - \tilde{\ell}(\boldsymbol{H}'_{\Lambda}, \boldsymbol{r}'_{\Lambda}; \boldsymbol{z}'), \qquad (5.38)
\end{aligned}$$

where the last inequality holds because $\boldsymbol{r}''$ is the minimizer of $\tilde{\ell}(\boldsymbol{H}'', \boldsymbol{r}; \boldsymbol{z}'')$.

We shall prove that $\tilde{\ell}(\boldsymbol{H}'_{\Lambda}, \boldsymbol{r}_{\Lambda}; \boldsymbol{z}') - \tilde{\ell}(\boldsymbol{H}''_{\Lambda}, \boldsymbol{r}_{\Lambda}; \boldsymbol{z}'')$ is Lipschitz w.r.t. $\boldsymbol{r}$, which implies the Lipschitz of $\boldsymbol{v}'(\boldsymbol{D}; \boldsymbol{z})$ and $\boldsymbol{e}'(\boldsymbol{D}; \boldsymbol{z})$. By algebra, we have

$$\begin{aligned}
\nabla_{\boldsymbol{r}} \left(\tilde{\ell}(\boldsymbol{H}'_{\Lambda}, \boldsymbol{r}_{\Lambda}; \boldsymbol{z}') - \tilde{\ell}(\boldsymbol{H}''_{\Lambda}, \boldsymbol{r}_{\Lambda}; \boldsymbol{z}'') \right) = \lambda_1 \Big[& \boldsymbol{H}'^{\top}_{\Lambda} (\boldsymbol{H}'_{\Lambda} - \boldsymbol{H}''_{\Lambda}) + (\boldsymbol{H}'_{\Lambda} - \boldsymbol{H}''_{\Lambda})^{\top} \boldsymbol{H}''_{\Lambda} \\
& + \boldsymbol{H}'^{\top}_{\Lambda} (\boldsymbol{z}'' - \boldsymbol{z}') + (\boldsymbol{H}''_{\Lambda} - \boldsymbol{H}'_{\Lambda})^{\top} \boldsymbol{z}'' \Big].
\end{aligned}$$

Note that $\|\boldsymbol{H}'_{\Lambda}\|_F$, $\|\boldsymbol{H}''_{\Lambda}\|_F$ and $\boldsymbol{z}''$ are all uniformly bounded by Assumption $(A1)$ and Proposition 5.3. Hence, there exists uniform constants c_1 and c_2, such that

$$\left\| \nabla_{\boldsymbol{r}} \left(\tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_{\Lambda}, \boldsymbol{r}_{\Lambda}) - \tilde{\ell}(\boldsymbol{z}'', \boldsymbol{H}''_{\Lambda}, \boldsymbol{r}_{\Lambda}) \right) \right\| \leq c_1 \left\| \boldsymbol{H}'_{\Lambda} - \boldsymbol{H}''_{\Lambda} \right\|_F + c_2 \left\| \boldsymbol{z}' - \boldsymbol{z}'' \right\|,$$

which implies that $\tilde{\ell}(\boldsymbol{H}'_{\Lambda}, \boldsymbol{r}_{\Lambda}; \boldsymbol{z}') - \tilde{\ell}(\boldsymbol{H}''_{\Lambda}, \boldsymbol{r}_{\Lambda}; \boldsymbol{z}'')$ is Lipschitz w.r.t $\boldsymbol{r}_{\Lambda}$ where the Lipschitz coefficient $c(\boldsymbol{H}'_{\Lambda}, \boldsymbol{H}''_{\Lambda}, \boldsymbol{z}', \boldsymbol{z}'') = c_1 \|\boldsymbol{H}'_{\Lambda} - \boldsymbol{H}''_{\Lambda}\|_F + c_2 \|\boldsymbol{z}' - \boldsymbol{z}''\|$. Combining this fact with (5.37) and (5.38), we obtain

$$\kappa_1 \left\| \boldsymbol{r}''_{\Lambda} - \boldsymbol{r}'_{\Lambda} \right\|^2 \leq c(\boldsymbol{H}'_{\Lambda}, \boldsymbol{H}''_{\Lambda}, \boldsymbol{z}', \boldsymbol{z}'') \left\| \boldsymbol{r}''_{\Lambda} - \boldsymbol{r}'_{\Lambda} \right\|.$$

Therefore, $\boldsymbol{r}(\boldsymbol{D}; \boldsymbol{z})$ is Lipschitz and so are $\boldsymbol{v}(\boldsymbol{D}; \boldsymbol{z})$ and $\boldsymbol{e}(\boldsymbol{D}; \boldsymbol{z})$. Note that according to Proposition 5.14,

$$\begin{aligned}
\nabla f(\boldsymbol{D}') - \nabla f(\boldsymbol{D}'') = \mathbb{E}_{\boldsymbol{z}} \Big[& (\boldsymbol{H}' \boldsymbol{r}' - \boldsymbol{z})(\boldsymbol{v}')^{\top} - (\boldsymbol{H}'' \boldsymbol{r}'' - \boldsymbol{z})(\boldsymbol{v}'')^{\top} \Big] \\
= \mathbb{E}_{\boldsymbol{z}} \Big[& \boldsymbol{H}' \boldsymbol{r}' (\boldsymbol{v}' - \boldsymbol{v}'')^{\top} + (\boldsymbol{H}' - \boldsymbol{H}'') \boldsymbol{r}' (\boldsymbol{v}'')^{\top} \\
& + \boldsymbol{H}'' (\boldsymbol{r}' - \boldsymbol{r}'') (\boldsymbol{v}'')^{\top} + \boldsymbol{z} (\boldsymbol{v}'' - \boldsymbol{v}')^{\top} \Big].
\end{aligned}$$

Thus,

$$
\begin{aligned}
&\left\|\nabla f(\boldsymbol{D}') - \nabla f(\boldsymbol{D}'')\right\|_F \\
\overset{\zeta_1}{\leq}\ &\mathbb{E}_{\boldsymbol{z}}\Big[\left\|\boldsymbol{H}'\boldsymbol{r}'\right\|\left\|\boldsymbol{v}' - \boldsymbol{v}''\right\| + \left\|\boldsymbol{H}' - \boldsymbol{H}''\right\|_F\left\|\boldsymbol{r}'\boldsymbol{v}''^{\top}\right\|_F \\
&\quad + \left\|\boldsymbol{H}''\right\|_F\left\|\boldsymbol{r}' - \boldsymbol{r}''\right\|\left\|\boldsymbol{v}''\right\| + \left\|\boldsymbol{z}\right\|\left\|\boldsymbol{v}' - \boldsymbol{v}''\right\|\Big] \\
\overset{\zeta_2}{\leq}\ &\mathbb{E}_{\boldsymbol{z}}\Big[\left(\gamma_1 + \gamma_2\left\|\boldsymbol{z}\right\|\right)\left\|\boldsymbol{H}' - \boldsymbol{H}''\right\|_F\Big] \\
\overset{\zeta_3}{\leq}\ &\gamma_0\left\|\boldsymbol{D}' - \boldsymbol{D}''\right\|_F,
\end{aligned}
$$

where γ_0, γ_1 and γ_2 are all uniform constants. Here, ζ_1 holds due to the convexity of $\|\cdot\|_F$. ζ_2 is derived by using the result that $\boldsymbol{r}(\boldsymbol{H};\boldsymbol{z})$ and $\boldsymbol{v}(\boldsymbol{H};\boldsymbol{z})$ are both Lipschitz and $\boldsymbol{H}'$, $\boldsymbol{H}''$, $\boldsymbol{r}'$, $\boldsymbol{r}''$, $\boldsymbol{v}'$ and $\boldsymbol{v}''$ are all uniformly bounded. ζ_3 holds because $\boldsymbol{z}$ is uniformly bounded and $\|\boldsymbol{H}' - \boldsymbol{H}''\|_F = \|\boldsymbol{D}' - \boldsymbol{D}''\|_F$. Thus, we complete the proof. □

Proof. (Proof of Theorem 5.10) Since $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$ are uniformly bounded (Proposition 5.3), there exist sub-sequences of $\{\frac{1}{t}\boldsymbol{A}_t\}$ and $\{\frac{1}{t}\boldsymbol{B}_t\}$ that converge to $\boldsymbol{A}_\infty$ and $\boldsymbol{B}_\infty$ respectively. Then $\boldsymbol{D}_t$ will converge to $\boldsymbol{D}_\infty$. Let $\boldsymbol{W}$ be an arbitrary matrix in $\mathbb{R}^{d\times r}$ and $\{h_k\}_{k\geq 1}$ be any positive sequence that converges to zero.

As $g_t(\boldsymbol{D})$ is a surrogate function of $f_t(\boldsymbol{D})$, for all t and k, we have

$$g_t(\boldsymbol{D}_t + h_k\boldsymbol{W}) \geq f_t(\boldsymbol{D}_t + h_k\boldsymbol{W}).$$

Let t tend to infinity, and note that $f(\boldsymbol{D}) = \lim_{t\to\infty} f_t(\boldsymbol{D})$, we have

$$g_\infty(\boldsymbol{D}_\infty + h_k\boldsymbol{W}) \geq f(\boldsymbol{D}_\infty + h_k\boldsymbol{W}).$$

Note that the Lipschitz property of $\nabla f(\boldsymbol{D})$ indicates that the second derivative of $f(\boldsymbol{D})$ is uniformly bounded. By a simple calculation, we can also show that it also holds for $g_t(\boldsymbol{D})$. This fact implies that we can take the first order Taylor expansion for both $g_t(\boldsymbol{D})$ and $f(\boldsymbol{D})$ even when t tends to

infinity (because the second order derivatives of them always exist). That is,

$$\mathrm{Tr}\left(h_k \boldsymbol{W}^\top \nabla g_\infty(\boldsymbol{D}_\infty)\right) + o(h_k \left\|\boldsymbol{W}\right\|_F) \geq \mathrm{Tr}\left(h_k \boldsymbol{W}^\top \nabla f(\boldsymbol{D}_\infty)\right) + o(h_k \left\|\boldsymbol{W}\right\|_F).$$

Above, we use the fact that $\lim_{t\to\infty} g_t(\boldsymbol{D}_t) - f(\boldsymbol{D}_t) = 0$ as implied by Corollary 5.7. By multiplying $\frac{1}{h_k\|\boldsymbol{W}\|_F}$ on both sides and note that $\{h_k\}_{k\geq 1}$ is a positive sequence, it follows that

$$\mathrm{Tr}\left(\frac{1}{\left\|\boldsymbol{W}\right\|_F}\boldsymbol{W}^\top \nabla g_\infty(\boldsymbol{D}_\infty)\right) + \frac{o(h_k \left\|\boldsymbol{W}\right\|_F)}{h_k \left\|\boldsymbol{W}\right\|_F} \geq \mathrm{Tr}\left(\frac{1}{\left\|\boldsymbol{W}\right\|_F}\boldsymbol{W}^\top \nabla f(\boldsymbol{D}_\infty)\right) + \frac{o(h_k \left\|\boldsymbol{W}\right\|_F)}{h_k \left\|\boldsymbol{W}\right\|_F}.$$

Now let k go to infinity, we obtain

$$\mathrm{Tr}\left(\frac{1}{\left\|\boldsymbol{W}\right\|_F}\boldsymbol{W}^\top \nabla g_\infty(\boldsymbol{D}_\infty)\right) \geq \mathrm{Tr}\left(\frac{1}{\left\|\boldsymbol{W}\right\|_F}\boldsymbol{W}^\top \nabla f(\boldsymbol{D}_\infty)\right).$$

Note that this inequality holds for any matrix $\boldsymbol{W} \in \mathbb{R}^{d\times r}$, so we actually have

$$\nabla g_\infty(\boldsymbol{D}_\infty) = \nabla f(\boldsymbol{D}_\infty).$$

As $\boldsymbol{D}_\infty$ is the minimizer of $g_\infty(\boldsymbol{D})$, we have

$$\nabla f(\boldsymbol{D}_\infty) = \nabla g_\infty(\boldsymbol{D}_\infty) = \boldsymbol{0}.$$

The proof is complete. □

Chapter 6

Estimation from Quantized Data

6.1 Background

Many practical problems can be formulated in principle as recovering an incomplete matrix from a small portion of its components, known as matrix completion. For instance, in the *Netflix Prize* competition, the underlying matrix consists of movie ratings from a variety of users, and the task is to predict the taste of the users for their unrated movies (i.e., missing entries). This problem has been studied for a decade, and the matrix factorization framework was proposed as an early answer [138]. In the seminal work [36], it was shown that if the singular vectors of the matrix to be recovered are dense enough and the observed entries are sampled uniformly random, then with high probability, a simple nuclear-norm based minimization program guarantees exact recovery.

Inspired by the elegant work of [36], a plethora of theoretical results exist that study the problem from different aspects. A partial list of the follow-up work includes: improving the sample complexity (i.e., parameter dependence) [38, 80, 66, 42], addressing structured noise [35, 81, 84, 44], developing fast provable algorithms [72, 77], mitigating memory cost [133, 11], to name just a few. Orthogonal to these work where the observed entries are real-valued, [47] considered the problem in the 1-bit setup. That is, given a target low-rank matrix which is real-valued, one only sees some sign patterns ($+1$ or -1) determined by the true matrix. The goal, however, is still to recover the real-valued matrix by using as few samples as possible.

The 1-bit setting is of broad interest for the machine learning community. From the theoretical perspective, it immediately raises the challenge that a straightforward observation model makes the

problem ill-posed. Suppose that the binary patterns are obtained by taking the sign of the entries of the true matrix. Then even for a rank-one matrix $\boldsymbol{M} = \boldsymbol{u}\boldsymbol{v}^\top$ where $\boldsymbol{u}$ and $\boldsymbol{v}$ are column vectors, one can freely modify the magnitude of the elements of $\boldsymbol{u}$ and $\boldsymbol{v}$ without changing the sign patterns of $\boldsymbol{M}$. The second issue coming up with the 1-bit setting is a tractable recovery paradigm. Since the sign function is not convex, one cannot tailor the nuclear-norm based convex program [36] to this case. Another concern is the loss of estimation accuracy owing to quantization, and a precise characterization of the trade-off between bits and sample size. Related to the sign patterns, it is also interesting to ask if there is a provable algorithm that is tolerant to noise.

In [47], they answered the first two questions by showing that, a nuclear-norm constrained convex program guarantees exact recovery from binary measurements if the observations are generated from a distribution parameterized by the true matrix. [15] derived the statistical error rate of multi-bit quantization which gives a partial answer to the third question. We in this chapter tackle the question of robustness: **(a)** can we exactly recover the matrix in polynomial time if the observations are flipped with some probability close to $1/2$; **(b)** if yes, how many samples suffice and is this sample complexity optimal.

Our motivation is two-fold. For practitioners, realistic data are usually discrete. For instance, the data matrix of the social network that represents whether two individuals are friends or not is binary. Sometimes the data are intended to be quantized, due to memory or communication limit. Additionally, it is easier to get quantized/binary feedback data from users as opposed to real-valued data. For instance, Netflix recently changed its rating system that only requires the user to say a "thumbs up" or "thumbs down". The system then has to process this feedback and predict a real value (a percentage value that the user will like a new movie) for the missing entry. On the other hand, there is a large body of work studying the robustness of original matrix completion while little is known for the 1-bit case. In the 1-bit setting, the sign flipping noise is no longer additive which poses specific challenges for theoretical analysis.

6.1.1 Contributions

We offer a positive answer to the noisy 1-bit matrix completion problem. In particular, we consider the following noise model: for each binary observation, it is flipped with probability $\tau \in [0, 1/2)$ where τ itself is a random variable. We assume that we have the knowledge of the distribution of τ in

order to construct an estimator. Yet, surprisingly we show that τ affects the estimation only through its mean. Formally, we prove that for any rank-r matrix $\boldsymbol{M} \in \mathbb{R}^{d_1 \times d_2}$ that satisfies mild conditions, a nuclear-norm constrained maximum likelihood estimator exactly recovers $\boldsymbol{M}$, in the sense that the estimation error vanishes when the sample size is $\mathcal{O}\left(\mathrm{poly}\left(1-2\mathbb{E}[\tau]\right)^{-2} r(d_1+d_2)\log(d_1 d_2)\right)$. We also establish a lower bound on the statistical error, showing that the sample complexity we obtained is near-optimal in some regimes.

6.1.2 Related Work

Matrix completion is closely related to compressed sensing [50] where the goal is to recover a sparse vector from its compressed linear measurements. It is now well-understood that if the sensing matrix satisfies the restricted isometry property [37], then either convex programs like basis pursuit [41] and Lasso [140, 146] or greedy algorithms like orthogonal matching pursuit [115, 142] or iterative hard thresholding [20] can be used for sparse recovery. Encouraged by the success of compressed sensing, a large body of work was devoted to the nuclear-norm based convex optimization for low-rank matrix recovery, in view of the analogy between the ℓ_1 norm and the nuclear norm [58, 120, 40]. However, the essential difference is that the sampling operator in compressed sensing is Gaussian, while for matrix completion it is a deterministic zero-one matrix $\boldsymbol{e}_i\boldsymbol{e}_j^\top$, where $\boldsymbol{e}_i$ is the ith canonical basis and likewise for $\boldsymbol{e}_j$. In this light, theoretical results in compressed sensing cannot be transferred to the matrix completion problem directly [117].

In compressed sensing, the 1-bit setting has received a broad attention due to [25]. There is a variety of appealing work contributed to this emerging field [67, 70, 65], while recently [116] gave an optimal sample complexity that ensures exact recovery of the direction of the signal. It is very interesting to contrast such a result to the matrix completion problem, where we recall that in the matrix case, even the direction (i.e., $\boldsymbol{u}\boldsymbol{v}^\top/(\|\boldsymbol{u}\|\cdot\|\boldsymbol{v}\|)$) cannot be recovered from the knowledge of $\mathrm{sign}\left(\boldsymbol{u}\boldsymbol{v}^\top\right)$. This again suggests discrepancy between compressed sensing and matrix completion. Very recently, the statistical trade-off between the sample size and bit depth of compressed sensing was investigated in [135], and a guaranteed estimator of the magnitude of the signal was proposed in [83]. The trade-off of quantized matrix completion was also tackled in [15], but a full picture is still missing.

Of specific interest to 1-bit setting is the sign flipping noise. Such a kind of noise has been widely

studied in the learning theory community for more than a decade [100, 7, 156], in the context of learning halfspaces and binary classification. However, the target vector therein is a general object, i.e., without the sparsity structure. A unified analysis was presented recently in [6], showing possible improvement on noisy 1-bit compressed sensing using tools from learning theory.

Despite these promising results in 1-bit compressed sensing and learning theory, it turns out that the robustness of 1-bit matrix completion is not well-understood until now. Though these two problems are inherently linked, it has been recognized that extra efforts have to be made in the matrix case. In this work, we take a step to study symmetric noise, where the noise has the same distribution over the observed entries. This is a popular noise model that was also considered in [116] in the context of 1-bit compressed sensing.

6.1.3 Notation

Suppose d_1 and d_2 are two positive integers. We write $[d_1] \times [d_2]$ for the index set $\big\{(i,j) : 1 \le i \le d_1, 1 \le j \le d_2\big\}$. For a finite set Ω, we slightly abuse the notation to denote its cardinality by $|\Omega|$. Throughout the chapter, f and g are reserved for particular functions. Hence, f' and g' should be interpreted as the derivative evaluated at some point. Finally, the sign function $\operatorname{sign}(x)$ outputs $+1$ if $x \ge 0$ and outputs -1 otherwise. For a matrix X, $\operatorname{sign}(X)$ operates in an entry-wise manner. The indicator function is denoted by $\mathbf{1}_{\{E\}}$, which equals one if the event E is true and zero otherwise.

6.2 Problem Setup

In this section, we formulate the problem. Recall that $\boldsymbol{M} \in \mathbb{R}^{d_1 \times d_2}$ is the underlying low-rank matrix that we aim to recover, and $\Omega \subset [d_1] \times [d_2]$ is a subset that indexing the observed components. In conventional matrix completion [36], one observes m_{ij} for $(i,j) \in \Omega$. Though it seems natural to consider the 1-bit matrix completion problem as a recovery from $\operatorname{sign}(\boldsymbol{M}_\Omega)$, Davenport et al. pointed out that it is not possible even when the matrix $\boldsymbol{M}$ has rank one [47]. The good news is that if we add noise (e.g., Gaussian, logistic) before quantization, it is tractable to solve the problem. Formally, the observation model considered in [47] is as follows: for all $(i,j) \in \Omega$, we observe

$$y_{ij} = \operatorname{sign}(m_{ij} + z_{ij}), \tag{6.1}$$

where $\{z_{ij}\}$ are i.i.d. random noise. With a proper choice of a differentiable function $f : \mathbb{R} \to [0, 1]$, the above is equivalent to the following probabilistic model:

$$y_{ij} = \begin{cases} +1, \text{ with probability } f(m_{ij}), \\ -1, \text{ with probability } 1 - f(m_{ij}). \end{cases} \tag{6.2}$$

In fact, we can set $f(x) = \Pr(z_{11} + x \geq 0)$ for which the model (6.1) reduces to the model (6.2). Conversely, given the function $f(x)$, we may think of $\{z_{ij}\}$ as i.i.d. random noise with cumulative distribution function $F(x) := \Pr(z_{11} < x) = 1 - f(-x)$. In this way, (6.2) reduces to (6.1).

In this chapter, we will mainly consider the model (6.2), which is viewed as a noiseless probabilistic model. With this in mind, we are in the position to introduce the *noisy* probabilistic model. Our central interest is the random sign flipping. That is, in place of observing y_{ij} as in (6.2), we have

$$y'_{ij} = \delta_{ij} y_{ij}, \ \forall \ (i, j) \in \Omega, \tag{6.3}$$

where $\{\delta_{ij}\}$ are i.i.d. random variables such that

$$\delta_{ij} = \begin{cases} +1, \text{ with probability } 1 - \tau, \\ -1, \text{ with probability } \tau. \end{cases} \tag{6.4}$$

Above, τ itself might be a random variable but we impose $0 \leq \tau < 1/2$ to prevent model ambiguity. Note that $\tau = 0$ corresponds to the noiseless model studied in [47, 15]. The model (6.3) together with (6.4) indicate that for each element belonging to Ω, with probability τ the sign is flipped. Note that our assumption on τ is more general than [116, 135] which treat τ as a deterministic parameter.

It is worth mentioning that a more general noise model is that each δ_{ij} is parameterized by τ_{ij}, where $\{\tau_{ij}\}$ may differ from each other but subject to the constraint $0 \leq \tau_{ij} \leq \tau < 1/2$ for some parameter τ. This is known as bounded noise (a.k.a. Massart noise) [100] that has received a broad attention in learning theory [5, 7]. Extension to such kind of noise is an interesting future work.

6.2.1 Assumptions

Before presenting our estimator for $\boldsymbol{M}$, we need a few assumptions.

$(A1)$ Given $n > 0$, each component (i, j) is included in Ω with probability $\frac{n}{d_1 d_2}$. Hence, $\mathbb{E}\,|\Omega| = n$.

$(A2)$ The maximum absolute value of $\boldsymbol{M}$ is upper bounded by a parameter α, i.e., $\|\boldsymbol{M}\|_\infty \leq \alpha$.

$(A3)$ $\boldsymbol{M}$ lies in a nuclear-norm ball with radius $\alpha\sqrt{rd_1d_2}$ where r is the rank of $\boldsymbol{M}$.

Note that $(A1)$ assumes a Bernoulli sampling scheme for Ω which is more convenient than the uniform sampling. In fact, the equivalence between these two sampling models was pointed out in [38, 34]. The second assumption essentially excludes the case that $\boldsymbol{M}$ is too spiky. Otherwise, the recovery of $\boldsymbol{M}$ is ill-posed [106, 117]. Finally, $(A3)$ acts as a convex surrogate to the exact rank constraint $\mathrm{rank}\,(\boldsymbol{M}) \leq r$. To see this, we note that by algebra, the following holds:

$$\|\boldsymbol{M}\|_* \leq \sqrt{r}\,\|\boldsymbol{M}\|_F \leq \alpha\sqrt{rd_1d_2}.$$

As we will illustrate later, $(A3)$ also allows us to approximate $\boldsymbol{M}$ by solving a *convex* program.

Under these assumptions, we propose to solve the following problem in order to approximate $\boldsymbol{M}$:

$$\begin{aligned} &\max_{\boldsymbol{X}} L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}), \\ &\text{s.t. } \|\boldsymbol{X}\|_\infty \leq \alpha,\ \|\boldsymbol{X}\|_* \leq \alpha\sqrt{rd_1d_2}. \end{aligned} \tag{6.5}$$

Above, the objective function $L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})$ is given as follows:

$$L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) = \sum_{(i,j)\in\Omega} \Big[\mathbf{1}_{\{y'_{ij}=1\}} \log g(x_{ij}) + \mathbf{1}_{\{y'_{ij}=-1\}} \log\left(1 - g(x_{ij})\right)\Big], \tag{6.6}$$

where $g(x)$ is the function such that for every $(i, j) \in \Omega$, y'_{ij} equals 1 with probability $g(m_{ij})$. In this light, it is not hard to see that $L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})$ is the log-likelihood function and the optimum of (6.5) is a maximum likelihood estimator (MLE). In addition, we remark that the two constraints in (6.5) are due to our assumptions $(A2)$ and $(A3)$.

It remains to characterize the function $g(x)$ which is a crucial component of (6.5). Note that in

view of (6.3) and (6.4), we have the following conditional probability:

$$\Pr\left(y'_{ij} = 1 \mid \tau\right) = (1-\tau)f(m_{ij}) + \tau(1 - f(m_{ij})). \tag{6.7}$$

Thus, depending on the distribution of τ, $g(x)$ is computed in a different manner.

τ is discrete. In this case, let us suppose that the random variable τ takes value in $(\tau_1, \tau_2, \ldots, \tau_s)$ with corresponding probability $(p_1, p_2, \ldots, p_s)$. It then follows that

$$\Pr\left(y'_{ij} = 1\right) = \sum_{k=1}^{s} \Pr\left(y'_{ij} = 1, \tau = \tau_k\right) = \sum_{k=1}^{s} p_k \Pr\left(y'_{ij} = 1 \mid \tau = \tau_k\right).$$

Hence, letting

$$g(x) = \sum_{k=1}^{s} p_k \left((1-\tau_k)f(x) + \tau_k(1 - f(x))\right) = f(x)\mathbb{E}[1-2\tau] + \mathbb{E}[\tau]. \tag{6.8}$$

gives $\Pr(y'_{ij} = 1) = g(m_{ij})$ as desired.

τ is continuous. Suppose that the probability density function of τ is $h_\tau(\cdot)$. Then by simple calculation, it can be shown that

$$g(x) = \int_t h_\tau(t)\Big[(1-t)f(x) + t(1 - f(x))\Big]dt = f(x)\mathbb{E}[1-2\tau] + \mathbb{E}[\tau], \tag{6.9}$$

which is identical to the discrete case. Therefore, it turns out that the random flipping noise (6.4) affects the recovery only through the mean.

6.3 Main Results

Our main results characterize the statistical rate of the MLE produced by (6.6). There are two important quantities we need in the theoretical analysis, as described below:

$$\rho_\gamma^+ = \sup_{|x|\leq\gamma} \frac{|g'(x)|}{g(x)(1-g(x))}, \quad \rho_\gamma^- = \sup_{|x|\leq\gamma} \frac{g(x)(1-g(x))}{(g'(x))^2}. \tag{6.10}$$

By some algebra, it is not hard to see that the quantity ρ_γ^+ is essentially the Lipschitz constant of the likelihood function $L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})$. The other quantity ρ_γ^- is not associated with the curvature explicitly.

However, there is still some intuitive explanation on why this quantity enters our analysis. Indeed, presume that $g(x)$ is bounded from below in the interval $[-\gamma, \gamma]$. As $g'(x)$ approaches zero, we find that ρ_γ^- tends to infinity since

$$\frac{\mathrm{C}}{(g'(x))^2} \leq \frac{g(x)(1-g(x))}{(g'(x))^2} \leq \frac{1}{2(g'(x))^2}$$

for some constant C. In view of (6.9), this in turn suggests that either the function $f(x)$ is quite flat in the interval or $\mathbb{E}[\tau]$ is close to $1/2$, making it difficult to distinguish the entries of $\boldsymbol{M}$.

6.3.1 Upper Bound

With these notions on hand, we state our first result which upper bounds the error of the solution of (6.5) for the recovery of $\boldsymbol{M}$.

Theorem 6.1 (Upper Bound)**.** *Assume* $(A1)$*,* $(A2)$ *and* $(A3)$*. Suppose that the observation model follows* (6.3)*. Denote* $\widehat{\boldsymbol{M}}$ *the optimum of* (6.5)*. Then, with probability at least* $1 - \mathrm{C}_1/(d_1+d_2)$*, we have*

$$\frac{1}{d_1 d_2} \left\| \widehat{\boldsymbol{M}} - \boldsymbol{M} \right\|_F^2 \leq \psi_\alpha \sqrt{\frac{r(d_1+d_2)}{n}},$$

provided that $n \geq (d_1+d_2)\log(d_1 d_2)$*. Above,* $\psi_\alpha = \mathrm{C}_2 \rho_\alpha^+ \rho_\alpha^-$.

The theorem implies that as soon as we randomly sample $n \geq \psi_\alpha^2 r(d_1+d_2)\log(d_1 d_2)$ entries, the estimation error vanishes and exact recovery is achieved. Note that, the dependence on the matrix rank r and the dimension (d_1, d_2) is optimal up to a logarithmic factor.

The theorem also suggests that the random flipping noise τ affects the recovery through the quantity ψ_α, which is multiplicative. For concreteness, we give estimates of the quantity ψ_α for several prevalent choices of $f(x)$. The connection between $g(x)$ and $f(x)$ (see (6.9)) immediately indicates the form of ψ_α. In the following we write $a := \mathbb{E}[\tau]$ for brevity.

- Logistic regression: $f(x) = e^x/(1+e^x)$. We have

$$\rho_\alpha^+ = 1, \quad \rho_\alpha^- = \frac{(1+e^\alpha)^2}{(1-2a)^2 e^{2\alpha}} \left((1-e^\alpha)a + e^\alpha\right)\left((e^\alpha - 1)a + 1\right).$$

Therefore, if we treat the parameter α as a constant, say $e^{\alpha} = 2$, it follows that

$$\psi_{\alpha} = \mathcal{O}\left(\frac{(a+1)(2-a)}{(1-2a)^2}\right) = \mathcal{O}\left(\frac{1}{(1-2a)^2}\right),$$

where the second equivalence follows by investigating the asymptotic behavior when a approaches $1/2$ from below. The above quickly implies that the sample size

$$n = \mathcal{O}\left((1-2a)^{-4} r(d_1+d_2)\log(d_1 d_2)\right)$$

suffices for exact recovery even when nearly half of the entries are flipped (i.e., $a = 1/2 - \epsilon$ for some small quantity ϵ).

- Probit regression: $f(x) = \Phi(x/\sigma)$. This corresponds with the scenario where $\{z_{ij}\}$ in (6.1) are Gaussian with variance σ^2. We have

$$\rho_{\alpha}^{+} \leq \frac{4}{(1-2a)\sigma}\left(\frac{\alpha}{\sigma}+1\right), \quad \rho_{\alpha}^{-} \leq \frac{\pi\sigma^2}{(1-2a)^2}\exp(\alpha^2/(2\sigma^2)).$$

This gives an upper bound of ψ_{α} as follows:

$$\psi_{\alpha} \leq \mathcal{O}\left(\frac{\alpha+\sigma}{(1-2a)^3}\exp\left(\frac{\alpha^2}{2\sigma^2}\right)\right).$$

It is not hard to see that there exists a threshold $\sigma^* > \alpha$ that minimizes the right-hand side above, hence is a heuristically optimal choice. When $\sigma < \sigma^*$, one can increase the variance to obtain a better error bound. This is not surprising since on one spectrum, if the variance is too small, the model (6.1) reduces to $y_{ij} = \operatorname{sign}(m_{ij})$ for which recovery is not possible [47]. On the other extreme, if σ is too large, then the function $f'(x)$ (and hence $g'(x)$) becomes flat, which makes recovery challenging as we have discussed earlier.

In the regime where the parameter α is a constant, we obtain the sample complexity $n = \mathcal{O}\left((1-2a)^{-6} r(d_1+d_2)\log(d_1 d_2)\right)$. This is worse than the logistic case. Given that these two models are quite similar to each other, we conjecture that it might be an artifact of our analysis.

- Laplacian: $f'(x) = -\frac{1}{2b}\exp(-|x|/b)$. We have

$$\rho_\alpha^+ = \frac{2(1-2a)}{b}, \quad \rho_\alpha^- = \frac{b^2}{(1-2a)^2}\left(2(e^{\alpha/b}-1)a+1\right) \times \left(2(1-e^{\alpha/b})a + 2e^{\alpha/b} - 1\right),$$

which yields

$$\psi_\alpha = \mathcal{O}\left(\frac{1}{1-2a}\right),$$

provided that the parameters α and b are constants, e.g., $e^{\alpha/b} = 2$. This gives us the sample complexity $n = \mathcal{O}\left((1-2a)^{-2} r(d_1+d_2)\log(d_1 d_2)\right)$ which is better than the logistic case. It is also worth mentioning that like probit regression, there exists an optimal choice of the parameter b that minimizes the upper bound of the statistical error, though we do not pursue it here.

6.3.2 Lower Bound

Our second theorem provides a lower bound on the statistical error for the recovery of $\boldsymbol{M}$. It asserts that under the observation model (6.3) and sampling scheme $(A1)$, we can always find an instance $\boldsymbol{M}$ satisfying $(A2)$ and $(A3)$, such that with a non-trivial probability (say, $3/4$), any algorithm requires as many samples as Theorem 6.1 suggests for the sake of recovering $\boldsymbol{M}$.

Theorem 6.2 (Lower Bound)**.** *Fix the parameters α, r d_1 and d_2 with $\alpha \geq 1$ and $r \geq 16$. Suppose that $\alpha^2 r \max\{d_1, d_2\} \geq \mathrm{C}_0$ for some absolute constant C_0. Further suppose that $g'(x)$ is non-increasing for $x > 0$. Let Ω by an arbitrary index set with $|\Omega| = n$ and assume the noisy observation model* (6.3)*. Then there exists $\boldsymbol{M}$ satisfying $(A2)$ and $(A3)$ such that for any algorithm, with probability at least $3/4$, its output $\widehat{\boldsymbol{M}}$ satisfies*

$$\frac{1}{d_1 d_2}\left\|\widehat{\boldsymbol{M}} - \boldsymbol{M}\right\|_F^2 \geq \min\left\{\mathrm{C}_1, \mathrm{C}_2 \phi_\alpha \sqrt{\frac{r\max\{d_1,d_2\}}{n}}\right\},$$

provided that the right-hand side is larger than $r\alpha^2/\min\{d_1,d_2\}$. Above, $\phi_\alpha = \alpha(\rho^-_{0.75\alpha})^{1/2}$.

A few remarks are in order. First and foremost, it is shown that $n = \mathcal{O}\left(\phi_\alpha^2 r d_2\right)$ samples are necessary for exact recovery where we assume $d_1 \leq d_2$ without loss of generality. The dependence on the rank r and matrix dimension (d_1, d_2) matches the upper bound of Theorem 6.1 (up to a

logarithmic factor), justifying the optimality (provided that α is a constant). Regarding the noise parameter $\mathbb{E}[\tau]$ contained in ϕ_α, it is not hard to see that for all the choices of $f(x)$ (i.e., logistic, probit, laplacian), our lower bound implies that n should be proportional to $(1-2\mathbb{E}[\tau])^{-2}$, indicating a room for improvement of the upper bound in the logistic and probit cases (the upper bound for the laplacian case we established is optimal).

Table 6.1: **Upper and lower bounds on the sample complexity in the regime where α is a constant.**

$f(x)$	Upper bound	Lower bound
Logistic	$\mathcal{O}\left((1-2\mathbb{E}[\tau])^{-4} r(d_1+d_2)\log(d_1 d_2)\right)$	$\mathcal{O}\left((1-2\mathbb{E}[\tau])^{-2} r\max\{d_1,d_2\}\right)$
Probit	$\mathcal{O}\left((1-2\mathbb{E}[\tau])^{-6} r(d_1+d_2)\log(d_1 d_2)\right)$	$\mathcal{O}\left((1-2\mathbb{E}[\tau])^{-2} r\max\{d_1,d_2\}\right)$
Laplacian	$\mathcal{O}\left((1-2\mathbb{E}[\tau])^{-2} r(d_1+d_2)\log(d_1 d_2)\right)$	$\mathcal{O}\left((1-2\mathbb{E}[\tau])^{-2} r\max\{d_1,d_2\}\right)$

Now let us investigate the conditions in Theorem 6.2. Note that we did not optimize the constants. For example, the condition $r \geq 16$ can be relaxed to, e.g., $r \geq 4$. The condition $\alpha^2 r d_2 \geq \mathrm{C}_0$ is easy to satisfy, especially in the high-dimensional regime where $r, d_2 \to \infty$. In fact, this condition is invoked only for a proof technique. We also point out that it is very mild to assume $g'(x)$ is decreasing in $\mathbb{R}^+$. Such a requirement amounts to impose that the probability density function has a non-increasing tail, which holds for the popular statistical models in Section 6.3.1. Finally, when the rank $r \leq \mathcal{O}\left(d_1/\alpha^2\right)$, the right-hand side of the inequality in the theorem is always larger than $r\alpha^2/d_1$. It turns out that under the setting $\alpha = \mathcal{O}(1)$, the lower bound holds even when the matrix rank is of the same order of the dimension. We summarize the established bounds in Table 6.1.

6.4 Proof Sketch

Our proof technique follows from [47] but tailored to the noisy situation. We will consider the following centralized loss function:

$$\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) := L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - L_{\Omega,\boldsymbol{Y}'}(0).$$

To prove Theorem 6.1, we show the following crucial lemma.

Lemma 6.3. *Let the set $\mathcal{S}$ be*

$$\mathcal{S} = \left\{ \boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2} : \|\boldsymbol{X}\|_* \leq \alpha \sqrt{r d_1 d_2} \right\}.$$

Write

$$G_{\Omega, \boldsymbol{Y}'} = \sup_{\boldsymbol{X} \in \mathcal{S}} \left| \bar{L}_{\Omega, \boldsymbol{Y}'}(\boldsymbol{X}) - \mathbb{E}\bar{L}_{\Omega, \boldsymbol{Y}'}(\boldsymbol{X}) \right|, \quad \bar{G} = \alpha \rho_\alpha^+ \sqrt{r} \sqrt{n(d_1 + d_2) + d_1 d_2 \log(d_1 d_2)}.$$

Then it follows that

$$\Pr\left(G_{\Omega, \boldsymbol{Y}'} \geq \mathrm{C}_0 \bar{G} \right) \leq \frac{\mathrm{C}_1}{d_1 + d_2},$$

for some absolute constants C_0 *and* C_1.

Recall that the likelihood function we defined in (6.6), which is not averaged by n. Hence, the above lemma suggests that for n large enough, the shifted loss $\frac{1}{n}\bar{L}_{\Omega, \boldsymbol{Y}'}(\boldsymbol{X})$ concentrates around its expectation with the rate $\mathcal{O}\left(1/\sqrt{n}\right)$.

On the other hand, by algebra we can show that

$$D(g(\boldsymbol{M}) || g(\widehat{\boldsymbol{M}})) \leq \frac{2}{n} G_{\Omega, \boldsymbol{Y}'},$$

where the left-hand side is the KL divergence, which is bounded from below by the averaged least-squares loss:

$$\frac{1}{d_1 d_2} \left\| \widehat{\boldsymbol{M}} - \boldsymbol{M} \right\|_F^2 \leq 8 \rho_\alpha^+ D(g(\boldsymbol{M}) || g(\widehat{\boldsymbol{M}})).$$

This immediately implies Theorem 6.1 after some rearrangements.

The lower bound follows from standard information theoretic arguments. To be more detailed, we construct a set of matrices that satisfying $(A2)$ and $(A3)$ but the discrepancy between the members of this set is large in terms of Frobenius norm. We then show that for any true matrix $\boldsymbol{M}$ coming from this set, it is not easy for any recovery algorithm to output a solution that is quite close to it. This suggests a lower bound as stated in Theorem 6.2. See Appendix 6.B for the full proof.

6.5 Numerical Study

We complement our theoretical findings by performing a comprehensive set of simulations. In particular, our focus is on how the estimation error changes with the sample size n and the random sign flipping noise. We first elaborate the experimental settings.

Data. For simplicity, we set $d_1 = d_2 = d$, where $d = 200$ if not specified. We randomly generate the true real-valued matrix $\boldsymbol{M} \in \mathbb{R}^{d \times d}$ such that it has rank r and $\|\boldsymbol{M}\|_\infty \leq 1$. To be more concrete, we construct two matrices $U, V \in \mathbb{R}^{d \times r}$ where the entries are drawn i.i.d. from a uniform distribution on the interval $[-1, 1]$. The low-rank matrix $\boldsymbol{M}$ is then given by the product $UV^\top$ followed by a normalization (that is, $\boldsymbol{M} \leftarrow \boldsymbol{M} / \|\boldsymbol{M}\|_\infty$). Given a sample size n, the index set Ω is picked uniformly random such that $|\Omega| = n$. The noisy observation $\boldsymbol{Y}'_\Omega$ depends on the choice of $f(x)$ and the flipping parameter τ (see (6.3)). Here, we choose the probit regression for $f(x)$, i.e., $f(x) = \Phi(x/\sigma)$, the cumulative density function of Gaussian distribution. We further use the default value $\sigma = 0.3$ as suggested by [47].

Evaluation. We measure the discrepancy between the recovered matrix $\widehat{\boldsymbol{M}}$ and the true matrix $\boldsymbol{M}$ by the mean squared error (MSE). We will also report the relative error, which normalizes the MSE with $\|\boldsymbol{M}\|_F^2 / d^2$.

Other settings. The solver for the convex program (6.5) is publicly available at Davenport's homepage. We follow their default settings of the solver. Each experiment to be showed are conducted for 5 trials, and we report the averaged MSE and relative error over these trials.

6.5.1 Deterministic τ

Our first empirical study focuses on the error curve against sample size when τ is fixed as a scalar (so there is no randomness in τ). We point out that though τ is a deterministic quantity, the flipping noise is still random. Such a noise model was studied in the context of 1-bit compressed sensing [116, 135]. We set $\tau = 0.2$ which means for all $(i, j) \in \Omega$, the component y_{ij} is flipped with probability 0.2. We plot the curves of MSE and relative error in Figure 6.1 where we also vary the rank r from 1 to 10. Note that a larger rank indicates a more complicated problem, hence we need to draw more observations to achieve a low error, as illustrated in this figure. Also note that in the right panels, the x-axis is $1/\sqrt{n}$ (n is the sample size), and we find that the statistical error scales approximately

linear with it, which matches our theoretical prediction.

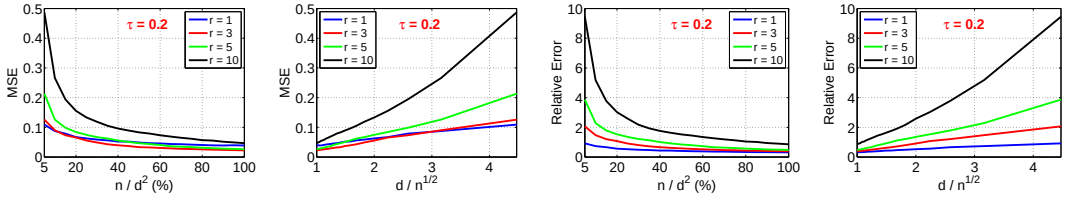


Figure 6.1: **Estimation error against sample size under fixed τ.** The x-axis is properly normalized by a constant for a better view. The statistical error is approximately linear with $1/\sqrt{n}$.

Then we fix the rank $r = 3$, and tune the parameter τ from 0 to 0.4. Note that τ is still a deterministic quantity. For each value of τ, we plot the error curves in Figure 6.2. It is not hard to see that the recovery becomes challenging when the data are grossly corrupted. For example, from the bottom-left panel, we observe that when 40 percents of the entries are observed, the relative error increases from 0.25 to 1 as τ changes from 0 to 0.3. Another notable aspect of Figure 6.2 is that, though the sample size is nearly linear with $1/\sqrt{n}$, the slopes of these lines are different from each other. This is actually implied by our theorems, which state that the error is proportional to $n^{-1/2}\,\mathrm{poly}\left(1/(1-2\tau)^2\right)$.

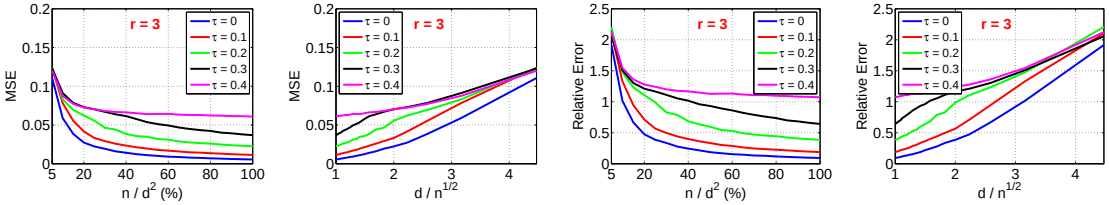


Figure 6.2: **Estimation error against sample size under fixed rank.**

6.5.2 Random τ

Now we investigate the situation where τ itself is a random variable. A remarkable implication of our theoretical analysis is that the random variable τ affects the recovery only through its mean. We verify this by randomly generating 3 different distributions for τ, say $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$. For each distribution $\mathcal{D}_i$, τ takes value from $\{\tau_{ik}\}_{k=1}^4$ with corresponding probability $\{p_{ik}\}_{k=1}^4$. The configuration $\{\tau_{ik}, p_{ik}\}_{k=1}^4$ is generated randomly, but subject to the constraints that **(i)** each τ_{ik} lies in the interval $[0, 1/2)$; **(ii)** $\mathbb{E}[\tau] = 0.2$; and **(iii)** $\sum_{k=1}^4 p_{ik} = 1$ for a given i. Then for each distribution $\mathcal{D}_i$, we manually corrupt the clean data $\boldsymbol{Y}_\Omega$ and run the solver to obtain an estimate.

The results are recorded in Figure 6.3 where we use the logarithmic scale for the y-axis to magnify the difference for the curves of these distributions. Even by doing so, we find that the three curves are almost lying on top of each other, which verifies our theoretical finding that the statistical error only depends on the mean of τ.

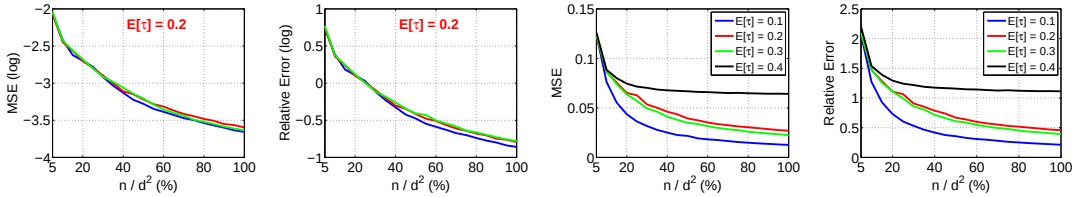


Figure 6.3: **Estimation error against sample size under the same and different noise expectation.** We observe that the statistical error depends on τ only through its mean.

Finally, we generate 4 distributions with different mean values using the same scheme just discussed. We illustrate the results in the last two panels of Figure 6.3. These curves again show that the sign flipping noise poses challenges for exact recovery. Careful readers may also compare it with Figure 6.2, in particular the right panels therein. It is not hard to see that for each configuration of the mean of τ, the curves in these two figures are quite similar.

6.6 Conclusion

In this chapter, we have introduced the noisy 1-bit matrix completion model, where each observed entry is flipped with some probability controlled by a random variable $\tau \in [0, 1/2)$. It has been shown that under rather mild conditions on the sampling scheme and the true matrix, a simple maximum likelihood estimator guarantees exact recovery. Along with our analysis, we have established a somewhat surprising result that the random variable τ enters the sample complexity only through its mean. When the binary data are generated from a Laplacian distribution, we have demonstrated that the upper bound matches the lower bound (up to a logarithmic factor). For logistic and Gaussian distributions, the lower bound implies potential room for improvements.

6.A Technical Lemmas

Lemma 6.4 (Theorem 1.1 in [127]). *There exists a constant K such that, for any n, m any $h \leq 2\log\max\{m,n\}$ and any $m \times n$ matrix $\boldsymbol{A} = (a_{ij})$ where a_{ij} are i.i.d. symmetric random variables, the following inequality holds:*

$$\max\left\{\mathbb{E}\max_{1\leq i\leq m}\|a_{i\cdot}\|^h, \mathbb{E}\max_{1\leq j\leq n}\left\|a_{\cdot j}^h\right\|\right\} \leq \mathbb{E}\|\boldsymbol{A}\|^h \leq K\left(\mathbb{E}\max_{1\leq i\leq m}\|a_{i\cdot}\|^h + \mathbb{E}\max_{1\leq j\leq n}\left\|a_{\cdot j}^h\right\|\right).$$

Lemma 6.5 (Symmetrization, Lemma 6.3 in [86]). *Let $F : \mathbb{R}_+ \to \mathbb{R}_+$ be convex. Then, for any finite sequence $\{t_i\}$ of independent mean zero random variables in B such that for every i $\mathbb{E}\big[F(\|t_i\|)\big] < \infty$, then*

$$\mathbb{E}\left[F\left(\frac{1}{2}\left\|\sum \xi_i t_i\right\|\right)\right] \leq \mathbb{E}\Big[F\Big(\Big\|\sum t_i\Big\|\Big)\Big] \leq \mathbb{E}\Big[F\Big(2\Big\|\sum \xi_i t_i\Big\|\Big)\Big],$$

where $\{\xi_i\}$ are i.i.d. Rademacher random variables.

Lemma 6.6 (Contraction, Theorem 4.12 in [86]). *Let $F : \mathbb{R}_+ \to \mathbb{R}_+$ be convex and increasing. Let $\psi_i : \mathbb{R} \to \mathbb{R}$ be contraction such that $\psi_i(0) = 0$. Then it holds that*

$$\mathbb{E}\left[F\left(\frac{1}{2}\sup_{t_1,\ldots t_N}\left|\sum_{i=1}^N \xi_i\psi_i(t_i)\right|\right)\right] \leq \mathbb{E}\left[F\left(\sup_{t_1,\ldots t_N}\left|\sum_{i=1}^N \xi_i t_i\right|\right)\right],$$

where $\{\xi_i\}$ are i.i.d. Rademacher random variables.

Lemma 6.7 (Lemma 2 in [47]). *Let f be a differentiable function and assume that*

$$\max\left\{\|\boldsymbol{M}\|_\infty, \left\|\widehat{\boldsymbol{M}}\right\|_\infty\right\} \leq \alpha.$$

Then

$$d_H^2\left(f(\boldsymbol{M}), f(\widehat{\boldsymbol{M}})\right) \geq \inf_{|x|\leq\alpha}\frac{(f'(x))^2}{8f(x)(1-f(x))}\frac{\left\|\boldsymbol{M}-\widehat{\boldsymbol{M}}\right\|_F^2}{d_1 d_2}.$$

Lemma 6.8 (Lemma 4 in [47]). *Suppose that $x, y \in (0,1)$. Then*

$$D(x||y) \leq \frac{(x-y)^2}{y(1-y)}.$$

Lemma 6.9 (Lemma 3 in [47]). *Let $\mathcal{K}$ be the set of matrices that satisfy $(A2)$ and $(A3)$. Let $0 < \nu \leq 1$ be a scalar such that $r(\nu)^{-2}$ is an integer that is not larger than d_1. Then there exists a subset $\mathcal{X} \subset \mathcal{K}$ with the following properties:*

1. $|\mathcal{X}| \geq \exp\left(\frac{rd_2}{16\nu^2}\right)$.
2. $\forall \boldsymbol{X} \in \mathcal{X}$, $|x_{ij}| = \alpha\nu$.
3. $\forall \boldsymbol{X}, \widetilde{\boldsymbol{X}} \in \mathcal{X}$ *with* $\boldsymbol{X} \neq \widetilde{\boldsymbol{X}}$, $\left\|\boldsymbol{X} - \widetilde{\boldsymbol{X}}\right\|_F^2 > \frac{1}{2}\alpha^2\nu^2 d_1 d_2$.

6.B Proofs

Recall the observation model: $\boldsymbol{M} \in \mathbb{R}^{d_1 \times d_2}$ is the true low-rank matrix and $\Omega \subset [d_1] \times [d_2]$ is the entries we observed. $\boldsymbol{Y} \in \mathbb{R}^{d_1 \times d_2}$ is the binary matrix determined by $\boldsymbol{M}$: for all $(i,j) \in \Omega$,

$$y_{ij} = \begin{cases} +1, \text{ with probability } f(m_{ij}), \\ -1, \text{ with probability } 1 - f(m_{ij}). \end{cases}$$

In the setting of symmetric noise, the observation $y'_{ij} = \delta_{ij} y_{ij}$ where δ_{ij} are i.i.d. and

$$\delta_{ij} = \begin{cases} +1, \text{ with probability } 1-\tau, \\ -1, \text{ with probability } \tau, \end{cases}$$

where $\tau \in (0, 1/2)$ itself can be a random variable. Therefore, conditioning on τ, we observe

$$\Pr\left(y'_{ij} = 1 \mid \tau\right) = (1-\tau) f(m_{ij}) + \tau(1 - f(m_{ij})).$$

Case 1. If τ is a discrete random variable, say

$$\Pr(\tau = \tau_k) = p_k,\ 1 \le k \le s,$$

then it is easy to see that

$$\begin{aligned}\Pr\left(y'_{ij} = 1\right) &= \sum_{k=1}^{s} \Pr\left(y'_{ij} = 1,\ \tau = \tau_k\right) \\ &= \sum_{k=1}^{s} \Pr\left(y'_{ij} = 1 \mid \tau = \tau_k\right) \cdot \Pr\left(\tau = \tau_k\right) \\ &= \sum_{k=1}^{s} p_k \Big[(1-\tau_k) f(m_{ij}) + \tau_k (1 - f(m_{ij}))\Big].\end{aligned}$$

Denote

$$g(x) = \sum_{k=1}^{s} p_k \Big[(1-\tau_k) f(x) + \tau_k (1 - f(x))\Big] = (1 - 2\mathbb{E}[\tau]) f(x) + \mathbb{E}[\tau].$$

We have

$$y'_{ij} = \begin{cases} +1, \text{ with probability } g(m_{ij}), \\ -1, \text{ with probability } 1 - g(m_{ij}). \end{cases}$$

Case 2. If τ is a continuous random variable with probability density function (pdf) $h_\tau(t)$, then we have

$$\begin{aligned}\Pr\left(y'_{ij} = 1\right) &= \int_t h_{\boldsymbol{Y},\tau}(y'_{ij} = 1, t) dt \\ &= \int_t h_{\boldsymbol{Y}|\tau}(y'_{ij} = 1 \mid t) h_\tau(t) dt \\ &= \int_t h_\tau(t) \Big[(1-t) f(m_{ij}) + t(1 - f(m_{ij}))\Big] dt,\end{aligned}$$

where $h_{\boldsymbol{Y},\tau}(y, t)$ is the joint pdf of y_{ij} and τ, and $h_{\boldsymbol{Y}|\tau}(y \mid t)$ is the conditional pdf. Thus, define

$$g(x) = \int_t h_\tau(t) \Big[(1-t) f(x) + t(1 - f(x))\Big] dt = (1 - 2\mathbb{E}[\tau]) f(x) + \mathbb{E}[\tau].$$

We again have

$$y'_{ij} = \begin{cases} +1, \text{ with probability } g(m_{ij}), \\ -1, \text{ with probability } 1 - g(m_{ij}). \end{cases}$$

Hence, the maximum likelihood estimator is given as follows:

$$\widehat{\boldsymbol{M}} = \arg\max_{\boldsymbol{X}} \ L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}), \quad \text{s. t. } \|\boldsymbol{X}\|_* \leq \alpha\sqrt{rd_1d_2}, \ \|\boldsymbol{X}\|_\infty \leq \gamma,$$

where

$$L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) := \sum_{(i,j)\in\Omega} \left(\mathbf{1}_{\{y_{ij}=1\}} \log g(x_{ij}) + \mathbf{1}_{\{y_{ij}=-1\}} \log(1 - g(x_{ij})) \right).$$

For the sake of a principled analysis, we will treat $g(x)$ as a general function at this point. Associated with the function $g(x)$ are two quantities:

$$\rho_\gamma^+ := \sup_{|x|\leq\gamma} \frac{|g'(x)|}{g(x)(1-g(x))}, \quad \rho_\gamma^- := \sup_{|x|\leq\gamma} \frac{g(x)(1-g(x))}{(g'(x))^2}.$$

We will use several kinds of distances in the proof. The first one is Hellinger distance that is given by

$$d_H^2(p,q) := (\sqrt{p} - \sqrt{q})^2 + (\sqrt{1-p} + \sqrt{1-q})^2, \quad \forall\, 0 \leq p,\ q \leq 1.$$

Extending it to the matrix, we write

$$d_H^2(P,Q) := \frac{1}{d_1d_2} \sum_{i,j} d_H^2(p_{ij}, q_{ij}),$$

where $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{R}^{d_1\times d_2}$ and the entries therein are between 0 and 1.

For two probability distributions $\mathcal{P}$ and $\mathcal{Q}$ on a finite set A, the Kullback-Leibler (KL) diver-

gence is defined as

$$D(\mathcal{P}||\mathcal{Q}) = \sum_{x\in A} \mathcal{P}(x) \log \frac{\mathcal{P}(x)}{\mathcal{Q}(x)}.$$

With a slight abuse, we write for two scalars $p, q \in [0,1]$

$$D(p||q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q},$$

and for two matrices $\boldsymbol{P}, \boldsymbol{Q} \in [0,1]^{d_1 \times d_2}$,

$$D(\boldsymbol{P}||\boldsymbol{Q}) = \frac{1}{d_1 d_2} \sum_{i,j} D(p_{ij}||q_{ij}).$$

Throughout the proof, we will work with a shifted MLE, i.e.

$$\begin{aligned}
&\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) := L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - L_{\Omega,\boldsymbol{Y}'}(0) && (6.11)\\
&= \sum_{(i,j)\in\Omega} \left(\mathbf{1}_{\{y_{ij}=1\}} \log \frac{g(x_{ij})}{g(0)} + \mathbf{1}_{\{y_{ij}=-1\}} \log \frac{1-g(x_{ij})}{1-g(0)} \right)\\
&= \sum_{i,j} \mathbf{1}_{\{(i,j)\in\Omega\}} \left(\mathbf{1}_{\{y_{ij}=1\}} \log \frac{g(x_{ij})}{g(0)} + \mathbf{1}_{\{y_{ij}=-1\}} \log \frac{1-g(x_{ij})}{1-g(0)} \right). && (6.12)
\end{aligned}$$

6.B.1 Proof of Lemma 6.3

Proof. Using the Markov's inequality, we have for any $\theta > 0$,

$$\begin{aligned}
&\Pr\left(\sup_{\boldsymbol{X}\in\mathcal{S}} \left| \bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - \mathbb{E}\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) \right| \geq \mathrm{C}_0 \alpha \rho_\gamma^+ \sqrt{r} \sqrt{n(d_1+d_2) + d_1 d_2 \log(d_1 d_2)} \right)\\
&\qquad \leq \frac{\mathbb{E}\left[\sup_{\boldsymbol{X}\in\mathcal{S}} \left| \bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - \mathbb{E}\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) \right|^\theta \right]}{\left(\mathrm{C}_0 \alpha \rho_\gamma^+ \sqrt{r} \sqrt{n(d_1+d_2) + d_1 d_2 \log(d_1 d_2)} \right)^\theta}. \qquad (6.13)
\end{aligned}$$

We bound the numerator above. Recall that

$$\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) = \sum_{i,j} \mathbf{1}_{\{(i,j)\in\Omega\}} \left(\mathbf{1}_{\{y'_{ij}=1\}} \log \frac{g(x_{ij})}{g(0)} + \mathbf{1}_{\{y'_{ij}=-1\}} \log \frac{1-g(x_{ij})}{1-g(0)} \right).$$

Let the random variable

$$\tilde{t}_{ij} = \mathbf{1}_{\{(i,j)\in\Omega\}} \left(\mathbf{1}_{\{y'_{ij}=1\}} \log \frac{g(x_{ij})}{g(0)} + \mathbf{1}_{\{y'_{ij}=-1\}} \log \frac{1-g(x_{ij})}{1-g(0)} \right),$$

and let

$$t_{ij} = \tilde{t}_{ij} - \mathbb{E}\tilde{t}_{ij}.$$

Then

$$\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - \mathbb{E}\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) = \sum_{i,j} t_{ij}.$$

Note that $\{t_{ij}\}$ are i.i.d. random variables with zero mean. The function $F(t) = \sup t^\theta$ is convex for $\theta \geq 1$, and $\mathbb{E}F(|t_{ij}|)$ is finite for all $(i,j) \in [d_1] \times [d_2]$. Hence, we can apply Lemma 6.5 to obtain

$$\begin{aligned}
&\mathbb{E}\Big[\sup_{\boldsymbol{X}\in\mathcal{S}} \left|\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - \mathbb{E}\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})\right|^\theta \Big] \\
&\leq 2^\theta \mathbb{E}\left[\sup_{\boldsymbol{X}\in\mathcal{S}} \left| \sum_{i,j} \xi_{ij} \mathbf{1}_{\{(i,j)\in\Omega\}} \left(\mathbf{1}_{\{y'_{ij}=1\}} \log \frac{g(x_{ij})}{g(0)} + \mathbf{1}_{\{y'_{ij}=-1\}} \log \frac{1-g(x_{ij})}{1-g(0)} \right) \right|^\theta \right],
\end{aligned}$$

where $\{\xi_{ij}\}$ are i.i.d. Rademacher random variables. Now observe that due to the construction of ρ_γ^+, both $\frac{1}{\rho_\gamma^+} \log \frac{g(x)}{g(0)}$ and $\frac{1}{\rho_\gamma^+} \log \frac{1-g(x)}{1-g(0)}$ are contractions and vanish at $x = 0$. Thereby, using Lemma 6.6 we have

$$\begin{aligned}
&\mathbb{E}\Big[\sup_{\boldsymbol{X}\in\mathcal{S}} \left|\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - \mathbb{E}\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})\right|^\theta \Big] \\
&\leq (4\rho_\gamma^+)^\theta \mathbb{E}\left[\sup_{\boldsymbol{X}\in\mathcal{S}} \left| \sum_{i,j} \xi_{ij} \mathbf{1}_{\{(i,j)\in\Omega\}} \left(\mathbf{1}_{\{y'_{ij}=1\}} x_{ij} - \mathbf{1}_{\{y'_{ij}=-1\}} x_{ij} \right) \right|^\theta \right] \\
&= (4\rho_\gamma^+)^\theta \mathbb{E}\left[\sup_{\boldsymbol{X}\in\mathcal{S}} \left| \sum_{i,j} \xi_{ij} \mathbf{1}_{\{(i,j)\in\Omega\}} y'_{ij} x_{ij} \right|^\theta \right].
\end{aligned}$$

With a simple algebra, we have

$$\begin{aligned}\Pr(\xi_{ij}y'_{ij}=1) &= \Pr(\xi_{ij}=1, y'_{ij}=1) + \Pr(\xi_{ij}=-1, y'_{ij}=-1)\\ &= \frac{1}{2}\left(\Pr(y'_{ij}=1)+\Pr(y'_{ij}=-1)\right) = \frac{1}{2},\end{aligned}$$

which implies that the distribution of $\xi_{ij}y'_{ij}$ is the same as that of ξ_{ij} for all $(i,j) \in [d_1]\times[d_2]$. Thus, by denoting Δ_Ω the matrix such that its (i,j)-th element is 1 if $(i,j)\in\Omega$ and 0 otherwise, and $\Xi = (\xi_{ij})$, it follows that

$$\begin{aligned}\mathbb{E}\Big[\sup_{\boldsymbol{X}\in\mathcal{S}}\left|\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - \mathbb{E}\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})\right|^\theta\Big] &\le (4\rho_\gamma^+)^\theta\mathbb{E}\left[\sup_{\boldsymbol{X}\in\mathcal{S}}\left|\sum_{i,j}\xi_{ij}\mathbf{1}_{\{(i,j)\in\Omega\}}x_{ij}\right|^\theta\right]\\ &= (4\rho_\gamma^+)^\theta\mathbb{E}\Big[\sup_{\boldsymbol{X}\in\mathcal{S}}\left|\langle\Delta_\Omega\circ\Xi,\boldsymbol{X}\rangle\right|^\theta\Big]\\ &\le (4\rho_\gamma^+)^\theta\mathbb{E}\Big[\sup_{\boldsymbol{X}\in\mathcal{S}}\left\|\Delta_\Omega\circ\Xi\right\|^\theta\left\|\boldsymbol{X}\right\|_*^\theta\Big]\\ &\le (\alpha\sqrt{rd_1d_2})^\theta\mathbb{E}\Big[\left\|\Delta_\Omega\circ\Xi\right\|^\theta\Big]. \qquad (6.14)\end{aligned}$$

Above, the last inequality follows from the nuclear norm constraint we imposed in the MLE estimator. Note that the (i,j)-th entry of the matrix $\Delta_\Omega\circ\Xi$ is given by $\mathbf{1}_{\{(i,j)\in\Omega\}}\xi_{ij}$, which are i.i.d. symmetric random variables. Thus, Lemma 6.4 implies that

$$\begin{aligned}\mathbb{E}\Big[\left\|\Delta_\Omega\circ\Xi\right\|^\theta\Big] &\le \mathrm{C}\left(\mathbb{E}\max_{1\le i\le d_1}\left(\sum_{j=1}^{d_2}(\xi_{ij}\Delta_{ij})^2\right)^{\theta/2} + \mathbb{E}\max_{1\le j\le d_2}\left(\sum_{i=1}^{d_1}(\xi_{ij}\Delta_{ij})^2\right)^{\theta/2}\right)\\ &= \mathrm{C}\left(\mathbb{E}\max_{1\le i\le d_1}\left(\sum_{j=1}^{d_2}\Delta_{ij}\right)^{\theta/2} + \mathbb{E}\max_{1\le j\le d_2}\left(\sum_{i=1}^{d_1}\Delta_{ij}\right)^{\theta/2}\right). \qquad (6.15)\end{aligned}$$

Fix i. By Bernstein's inequality, for all $t>0$,

$$\Pr\left(\left|\sum_{j=1}^{d_2}\left(\Delta_{ij}-\frac{n}{d_1d_2}\right)\right|>t\right) \le 2\exp\left(\frac{-t^2/2}{n/d_1+t/3}\right).$$

When $t \geq \frac{6n}{d_1}$, the above reduces to

$$\Pr\left(\left|\sum_{j=1}^{d_2}\left(\Delta_{ij} - \frac{n}{d_1 d_2}\right)\right| > t\right) \leq 2\exp(-t).$$

Suppose that $W_1, \ldots, W_{d_1}$ are i.i.d. exponential random variables with probability density $\exp(-t)$. Then it follows that

$$\Pr\left(\left|\sum_{j=1}^{d_2}\left(\Delta_{ij} - \frac{n}{d_1 d_2}\right)\right| > t\right) \leq 2\Pr(W_i \geq t).$$

On the other hand, we have

$$\begin{aligned}
&\left(\mathbb{E}\max_{1\leq i\leq d_1}\left(\sum_{j=1}^{d_2}\Delta_{ij}\right)^{\theta/2}\right)^{1/\theta} \\
&\leq \sqrt{\frac{n}{d_1}} + \left(\mathbb{E}\max_{1\leq i\leq d_1}\left|\sum_{j=1}^{d_2}(\Delta_{ij} - \frac{n}{d_1 d_2})\right|^{\theta/2}\right)^{1/\theta} \\
&\overset{\zeta_1}{=} \sqrt{\frac{n}{d_1}} + \left(\int_0^{+\infty}\Pr\left(\max_{1\leq i\leq d_1}\left|\sum_{j=1}^{d_2}(\Delta_{ij} - \frac{n}{d_1 d_2})\right|^{\theta/2} \geq t\right)dt\right)^{1/\theta} \\
&\leq \sqrt{\frac{n}{d_1}} + \left(\left(\frac{6n}{d_1}\right)^{\theta/2} + \int_{(6n/d_1)^{\theta/2}}^{+\infty}\Pr\left(\max_{1\leq i\leq d_1}\left|\sum_{j=1}^{d_2}(\Delta_{ij} - \frac{n}{d_1 d_2})\right|^{\theta/2} \geq t\right)dt\right)^{1/\theta} \\
&\leq \sqrt{\frac{n}{d_1}} + \left(\left(\frac{6n}{d_1}\right)^{\theta/2} + 2\int_{(6n/d_1)^{\theta/2}}^{+\infty}\Pr\left(\max_{1\leq i\leq d_1} W_i^{\theta/2} \geq t\right)dt\right)^{1/\theta} \\
&\overset{\zeta_2}{\leq} \sqrt{\frac{n}{d_1}} + \left(\left(\frac{6n}{d_1}\right)^{\theta/2} + 2\mathbb{E}\max_{1\leq i\leq d_1} W_i^{\theta/2}\right)^{1/\theta} \\
&\leq (1+\sqrt{6})\sqrt{\frac{n}{d_1}} + 2^{1/\theta}\left(\mathbb{E}\max_{1\leq i\leq d_1} W_i^{\theta/2}\right)^{1/\theta}.
\end{aligned}$$

Here, ζ_1 and ζ_2 use the identity $\mathbb{E}x = \int_0^{+\infty} \Pr(x \geq t)dt$ for any positive random variable x. It remains to bound $\mathbb{E}\max_{1\leq i\leq d_1} W_i^{\theta/2}$. Using the fact that W_i is exponential, we have

$$\begin{aligned}\mathbb{E}\max_{1\leq i\leq d_1} W_i^{\theta/2} &\leq \left|\max_{1\leq i\leq d_1} W_i - \log d_1\right|^{\theta/2} + \log^{\theta/2} d_1 \\ &\leq 2((\theta/2)!) + \log^{\theta/2} d_1 \leq 2(\theta/2)^{\theta/2} + \log^{\theta/2} d_1,\end{aligned}$$

where we apply Stirling's approximation in the last inequality. Thus,

$$2^{1/\theta}\left(\mathbb{E}\max_{1\leq i\leq d_1} W_i^{\theta/2}\right)^{1/\theta} \leq 2^{1/\theta}\left(\sqrt{\log d_1} + 2^{1/\theta}\sqrt{\theta/2}\right).$$

Picking $\theta = 2\log(d_1+d_2)$ gives

$$2^{1/\theta}\left(\mathbb{E}\max_{1\leq i\leq d_1} W_i^{\theta/2}\right)^{1/\theta} \leq (2+\sqrt{2})\sqrt{\log(d_1+d_2)}.$$

Putting pieces together, we obtain

$$\left(\mathbb{E}\max_{1\leq i\leq d_1}\left(\sum_{j=1}^{d_2}\Delta_{ij}\right)^{\theta/2}\right)^{1/\theta} \leq (1+\sqrt{6})\sqrt{\frac{n}{d_1}} + (2+\sqrt{2})\sqrt{\log(d_1+d_2)}.$$

Likewise, we can show that

$$\left(\mathbb{E}\max_{1\leq j\leq d_2}\left(\sum_{i=1}^{d_1}\Delta_{ij}\right)^{\theta/2}\right)^{1/\theta} \leq (1+\sqrt{6})\sqrt{\frac{n}{d_2}} + (2+\sqrt{2})\sqrt{\log(d_1+d_2)}.$$

Note that $\sqrt{x}$ is a concave function. Hence, Jensen's inequality implies that (6.15) can be bounded as follows:

$$\begin{aligned}\left(\mathbb{E}\Big[\,\|\Delta_\Omega \circ \Xi\|^\theta\,\Big]\right)^{1/\theta} &\leq \mathrm{C}^{1/\theta}\left((1+\sqrt{6})\sqrt{\frac{2n(d_1+d_2)}{d_1d_2}} + (2+\sqrt{2})\sqrt{\log(d_1+d_2)}\right) \\ &\leq \mathrm{C}^{1/\theta}2(1+\sqrt{6})\sqrt{\frac{n(d_1+d_2)+d_1d_2\log(d_1+d_2)}{d_1d_2}}.\end{aligned}$$

Plugging this back to (6.14), we have

$$\left(\mathbb{E}\Big[\sup_{\boldsymbol{X}\in\mathcal{S}}\left|\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})-\mathbb{E}\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})\right|^{\theta}\Big]\right)^{1/\theta}$$
$$\leq \mathrm{C}^{1/\theta}8(1+\sqrt{6})\alpha\rho_{\gamma}^{+}\sqrt{r}\sqrt{n(d_1+d_2)+d_1d_2\log(d_1+d_2)}.$$

Therefore, (6.13) is upper bounded by

$$\mathrm{C}\left(\frac{8(1+\sqrt{6})}{\mathrm{C}_0}\right)^{2\log(d_1+d_2)}\leq\frac{\mathrm{C}}{d_1+d_2},$$

as soon as we choose $\mathrm{C}_0\geq 8(1+\sqrt{6})\sqrt{e}$. □

Proposition 6.10. *Assume same conditions as in Theorem 6.1 but with a slightly more general assumption that* $\|\boldsymbol{M}\|_{\infty}\leq\gamma$ *in place of* $\|\boldsymbol{M}\|_{\infty}\leq\alpha$*. Then, with probability at least* $1-\mathrm{C}_1/(d_1+d_2)$*, the follows holds:*

$$d_H^2(g(\widehat{\boldsymbol{M}}),g(\boldsymbol{M}))\leq\mathrm{C}_2\rho_{\gamma}^{+}\alpha\sqrt{\frac{r(d_1+d_2)}{n}}\sqrt{1+\frac{(d_1+d_2)\log(d_1d_2)}{n}},$$

where C_1 *and* C_2 *are absolute constants.*

Proof. For any matrix $\boldsymbol{X}\in\mathbb{R}^{d_1\times d_2}$, we have

$$\begin{aligned}
&\mathbb{E}\big[\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})-\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{M})\big] &(6.16)\\
=&\,\mathbb{E}\big[L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})-L_{\Omega,\boldsymbol{Y}'}(\boldsymbol{M})\big]\\
=&\,\mathbb{E}\left[\sum_{i,j}\mathbf{1}_{\{(i,j)\in\Omega\}}\left(\mathbf{1}_{\{y'_{ij}=1\}}\log\frac{g(x_{ij})}{g(m_{ij})}+\mathbf{1}_{\{y'_{ij}=-1\}}\log\frac{1-g(x_{ij})}{1-g(m_{ij})}\right)\right]\\
=&\,\mathbb{E}\left[\sum_{i,j}\frac{n}{d_1d_2}\left(g(m_{ij})\log\frac{g(x_{ij})}{g(m_{ij})}+(1-g(m_{ij}))\log\frac{1-g(x_{ij})}{1-g(m_{ij})}\right)\right]\\
=&\,-nD(g(\boldsymbol{M})||g(\boldsymbol{X})). &(6.17)
\end{aligned}$$

On the other hand, for the optimum $\widehat{\boldsymbol{M}}$, it holds that

$$\begin{aligned}\bar{L}_{\Omega,\boldsymbol{Y}'}(\widehat{\boldsymbol{M}}) - \bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{M}) &= \mathbb{E}\big[\bar{L}_{\Omega,\boldsymbol{Y}'}(\widehat{\boldsymbol{M}}) - \bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{M})\big] + \Big(\bar{L}_{\Omega,\boldsymbol{Y}'}(\widehat{\boldsymbol{M}}) - \mathbb{E}\big[\bar{L}_{\Omega,\boldsymbol{Y}'}(\widehat{\boldsymbol{M}})\big]\Big) \\ &\quad + \big(\mathbb{E}\big[\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{M}) - \bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{M})\big) \\ &\leq \mathbb{E}\big[\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - \bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{M})\big] + 2\sup_{\boldsymbol{X}\in\mathcal{S}}\big|\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - \mathbb{E}\big[\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})\big]\big|,\end{aligned}$$

where we recall that $\mathcal{S}$ was defined in Lemma 6.3. Since $\widehat{\boldsymbol{M}}$ also maximizes $\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})$, we obtain

$$-\mathbb{E}\big[\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - \bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{M})\big] \leq 2\sup_{\boldsymbol{X}\in\mathcal{S}}\big|\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X}) - \mathbb{E}\big[\bar{L}_{\Omega,\boldsymbol{Y}'}(\boldsymbol{X})\big]\big|.$$

This together with (6.16) and Lemma 6.3 imply that

$$D(g(\boldsymbol{M})||g(\widehat{\boldsymbol{M}})) \leq 2\mathrm{C}_0\alpha_0\rho_\gamma^+\sqrt{\frac{r(d_1+d_2)}{n}}\sqrt{1+\frac{(d_1+d_2)\log(d_1d_2)}{n}}$$

holds with probability at least $1-\mathrm{C}_1/(d_1+d_2)$. Since the Hellinger distance is upper bounded by the KL divergence, we complete the proof. □

6.B.2 Proof of Theorem 6.1

Proof. Theorem 6.1 follows immediately from Prop. 6.10 and Lemma 6.7. □

6.B.3 Proof of Theorem 6.2

Proof. Without loss of generality, suppose that $d_1 \leq d_2$. Let

$$\epsilon^2 = \min\left\{\frac{1}{1024}, \mathrm{C}\alpha\sqrt{\frac{\rho_{0.75\alpha}^- r d_2}{n}}\right\}.$$

Pick

$$\frac{4\sqrt{2}\epsilon}{\alpha} \leq \nu \leq \frac{8\epsilon}{\alpha}.$$

It is easy to see that

$$\frac{r\alpha^2}{64\epsilon^2} \leq \frac{r}{\nu^2} \leq \frac{r\alpha^2}{32\epsilon^2}.$$

The length of this interval is $\frac{r\alpha^2}{64\epsilon}$, which is larger than 1 since $\alpha \geq 1$, $r \geq 16$ and $\epsilon^2 \leq 1/1024$. Hence, it is possible to pick a proper ν such that $\frac{r}{\nu^2}$ is an integer. Also, the assumption that $\epsilon^2 \geq O(r\alpha^2/d_1)$ suggests $r/\nu^2 \leq d_1$. Hence we have found an appropriate ν for Lemma 6.9.

Let $\mathcal{X}'_{\alpha/2,\nu}$ be a set that satisfies all the properties in Lemma 6.9 with parameter $\alpha/2$. Let

$$\mathcal{X} = \left\{ \boldsymbol{X}' + \alpha \left(1 - \frac{\nu}{2}\right) \boldsymbol{U} : \boldsymbol{X}' \in \mathcal{X}'_{\alpha/2,\nu} \right\},$$

where all the entries of $\boldsymbol{U}$ equal one.

First, we verify that each component in $\mathcal{X}$ satisfies $(A2)$ and $(A3)$. It is easy to see that for any $\boldsymbol{X} \in \mathcal{X}$, $|x_{ij}|$ either equals α or $(1-\nu)\alpha$, i.e., $\|\boldsymbol{X}\|_\infty \leq \alpha$ since $\nu < 1$. In addition,

$$\left\| \boldsymbol{X}' + \alpha \left(1 - \frac{\nu}{2}\right) \boldsymbol{U} \right\|_* \leq \left\| \boldsymbol{X}' \right\|_* + \alpha \left(1 - \frac{\nu}{2}\right) \|\boldsymbol{U}\|_* \leq \frac{\alpha}{2}\sqrt{r d_1 d_2} + \alpha \left(1 - \frac{\nu}{2}\right) \|\boldsymbol{U}\|_F .$$

Since $\nu \in (0,1)$ and $r \geq 16$, we have $2 - \nu \leq \sqrt{r}$, which together with $\|\boldsymbol{U}\|_F = \sqrt{d_1 d_2}$ imply that $\|\boldsymbol{X}\|_* \leq \alpha\sqrt{r d_1 d_2}$ for all $\boldsymbol{X} \in \mathcal{X}$.

We prove the theorem by showing that its converse is false. That is, suppose that there exists an algorithm such that for any $\boldsymbol{M} \in \mathcal{X}$ (which satisfies $(A2)$ and $(A3)$), with probability at least $1/4$, its output $\widehat{\boldsymbol{X}}$ satisfies

$$\frac{1}{d_1 d_2} \left\| \widehat{\boldsymbol{X}} - \boldsymbol{M} \right\|_F^2 < \epsilon^2. \tag{6.18}$$

Let $\boldsymbol{X}^* \in \mathcal{X}$ be the closest member to $\widehat{\boldsymbol{X}}$. For any $\widetilde{\boldsymbol{X}} \neq \boldsymbol{M} \in \mathcal{X}$, it follows that

$$\left\| \widetilde{\boldsymbol{X}} - \widehat{\boldsymbol{X}} \right\|_F \geq \left\| \widetilde{\boldsymbol{X}} - \boldsymbol{M} \right\|_F - \left\| \widehat{\boldsymbol{X}} - \boldsymbol{M} \right\|_F > 2\epsilon\sqrt{d_1 d_2} - \epsilon\sqrt{d_1 d_2} = \epsilon\sqrt{d_1 d_2}, \tag{6.19}$$

where the last inequality follows from (6.18) and the fact that for any $\boldsymbol{X}, \widetilde{\boldsymbol{X}} \in \mathcal{X}$,

$$\left\|\boldsymbol{X} - \widetilde{\boldsymbol{X}}\right\|_F^2 \geq \frac{\alpha^2\nu^2 d_1 d_2}{8} \geq 4d_1 d_2 \epsilon^2.$$

The first inequality above uses the third property in Lemma 6.9 and the second inequality follows from our choice of ν.

On the other hand, since $\boldsymbol{X}^*$ is the closest one to $\widehat{\boldsymbol{X}}$, we have

$$\left\|\boldsymbol{X}^* - \widehat{\boldsymbol{X}}\right\|_F \leq \left\|\boldsymbol{M} - \widehat{\boldsymbol{X}}\right\|_F \leq \epsilon\sqrt{d_1 d_2}. \tag{6.20}$$

Combining (6.19) and (6.20), we obtain

$$\left\|\boldsymbol{X}^* - \widehat{\boldsymbol{X}}\right\|_F < \left\|\widetilde{\boldsymbol{X}} - \widehat{\boldsymbol{X}}\right\|_F, \ \forall\, \widetilde{\boldsymbol{X}} \neq \boldsymbol{M},$$

which implies $\boldsymbol{X}^* = \boldsymbol{M}$. Since (6.18) holds with probability at least $1/4$,

$$\Pr\left(\boldsymbol{X}^* \neq \boldsymbol{M}\right) \leq \frac{3}{4}. \tag{6.21}$$

From a variant of Fano's inequality,

$$\Pr(\boldsymbol{X}^* \neq \boldsymbol{M}) \geq 1 - \frac{1 + d_1 d_2 \max_{\boldsymbol{X} \neq \widetilde{\boldsymbol{X}}} D(\boldsymbol{Y}'_\Omega | \boldsymbol{X} \,||\, \boldsymbol{Y}'_\Omega | \widetilde{\boldsymbol{X}})}{\log |\mathcal{X}|} \tag{6.22}$$

Denote

$$D = d_1 d_2 D(\boldsymbol{Y}'_\Omega | \boldsymbol{X} \,||\, \boldsymbol{Y}'_\Omega | \widetilde{\boldsymbol{X}}) = \sum_{(i,j)\in\Omega} D(y'_{ij} | x_{ij} \,||\, y'_{ij} | \tilde{x}_{ij}).$$

For each $(i,j) \in \Omega$, $D(y'_{ij}|x_{ij} \,||\, y'_{ij}|\tilde{x}_{ij})$ is either 0, $D(g(\alpha)||g(\alpha'))$ or $D(g(\alpha)||g(\alpha'))$, where $\alpha' = (1-\nu)\alpha$ and we recall that x_{ij}, $\tilde{x}_{ij}$ can only take value from $\{\alpha, \alpha'\}$. It thus follows from Lemma 6.8 that

$$D(y'_{ij}|x_{ij} \,||\, y'_{ij}|\tilde{x}_{ij}) \leq \frac{(g(\alpha) - g(\alpha'))^2}{g(\alpha')(1 - g(\alpha'))},$$

since $\alpha' < \alpha$. Now using the mean value theorem, we obtain

$$D \leq n(g'(\theta))^2 \frac{(\alpha - \alpha')^2}{g(\alpha')(1 - g(\alpha'))}, \text{ for some } \theta \in [\alpha', \alpha].$$

As we assumed that $g'(x)$ is decreasing in $(0, +\infty)$, we get

$$D \leq \frac{n(\nu\alpha)^2}{\rho^-_{\alpha'}} \leq \frac{64n\epsilon^2}{\rho^-_{\alpha'}}.$$

Due to the construction, the cardinality of $\mathcal{X}$ equals to that of $\mathcal{X}'_{\alpha/2,\nu}$. Hence, combining (6.21) and (6.22), we can show

$$\frac{1}{4} \leq \frac{D+1}{\log|\mathcal{X}|} \leq \frac{16\nu^2}{rd_2}\left(\frac{64n\epsilon^2}{\rho^-_{\alpha'}} + 1\right) \leq \frac{1024\epsilon^2}{\alpha^2 rd_2}\left(\frac{64n\epsilon^2}{\rho^-_{\alpha'}} + 1\right). \tag{6.23}$$

Note that when $64n\epsilon^2 \leq \rho^-_{\alpha'}$, we have

$$\frac{1}{4} \leq 1024\frac{2048\epsilon^2}{\alpha^2 rd_2},$$

implying $\alpha^2 rd_2 \leq 8$ due to the definition of ϵ. This contradicts our assumption that $\alpha^2 rd_2 \geq \mathrm{C}_0$ if we specify $\mathrm{C}_0 > 8$.

When $64n\epsilon^2 > \rho^-_{\alpha'}$, then (6.23) suggests

$$\frac{1}{4} \leq \frac{1024 \times 128 \times n\epsilon^4}{\rho^-_{\alpha'}\alpha^2 rd_2},$$

which gives

$$\epsilon^2 > \frac{\alpha\sqrt{\rho^-_{\alpha'}}}{1024}\sqrt{\frac{rd_2}{n}}.$$

Picking $\mathrm{C}_2 = 1/1024$ in the definition of ϵ and noting $\rho^-_{\alpha'} \geq \rho^-_{0.75\alpha}$ yields a contradiction. Therefore, (6.18) fails to hold with probability at least $3/4$. □

Bibliography

[1] Radosław Adamczak, Alexander E. Litvak, Alain Pajor, and Nicole Tomczak-Jaegermann. Restricted isometry property of matrices with independent columns and neighborly polytopes by random sampling. *Constructive Approximation*, 34(1):61–88, 2011.

[2] Alekh Agarwal, Sahand Negahban, and Martin J. Wainwright. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482, 2012.

[3] Matej Artač, Matjaž Jogan, and Aleš Leonardis. Incremental pca for on-line visual learning and recognition. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 781–784, 2002.

[4] Haim Avron, Satyen Kale, Shiva Prasad Kasiviswanathan, and Vikas Sindhwani. Efficient and practical stochastic subgradient descent for nuclear norm regularization. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[5] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In *Proceedings of the 28th Conference on Learning Theory*, pages 167–190, 2015.

[6] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Proceedings of the 29th Conference on Learning Theory*, pages 152–192, 2016.

[7] Pranjal Awasthi, Maria-Florina Balcan, and Philip M. Long. The power of localization for efficiently learning linear separators with noise. *Journal of the ACM*, 63(6):50:1–50:27, 2017.

[8] Bubacarr Bah and Jared Tanner. Improved bounds on restricted isometry constants for gaussian matrices. *SIAM Journal on Matrix Analysis Applications*, 31(5):2882–2898, 2010.

[9] Bubacarr Bah and Jared Tanner. Bounds of restricted isometry constants in extreme asymptotics: Formulae for Gaussian matrices. *Linear Algebra and its Applications*, 441:88–109, 2014.

[10] Sohail Bahmani, Bhiksha Raj, and Petros T. Boufounos. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research*, 14(1):807–841, 2013.

[11] Maria-Florina Balcan and Hongyang Zhang. Noise-tolerant life-long matrix completion via adaptive sampling. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 2955–2963, 2016.

[12] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael B. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[13] Pierre C. Bellec, Guillaume Lecué, and Alexandre B. Tsybakov. Slope meets lasso: improved oracle bounds and optimality. *CoRR*, abs/1605.08651, 2016.

[14] Dimitri P. Bertsekas. *Nonlinear Programming*. Massachusetts: Athena Scientific, 1999.

[15] Sonia A. Bhaskar. Probabilistic low-rank matrix completion from quantized measurements. *Journal of Machine Learning Research*, 17(60):1–34, 2016.

[16] Srinadh Bhojanapalli, Anastasios Kyrillidis, and Sujay Sanghavi. Dropping convexity for faster semi-definite optimization. In *Proceedings of the 29th Conference on Learning Theory*, pages 530–582, 2016.

[17] Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705–1732, 2009.

[18] Jeffrey D. Blanchard and Jared Tanner. Performance comparisons of greedy algorithms in compressed sensing. *Numerical Linear Algebra with Applications*, 22(2):254–282, 2015.

[19] Thomas Blumensath and Mike E. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, 2008.

[20] Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

[21] J. Frédéric Bonnans and Alexander Shapiro. Optimization problems with perturbations: A guided tour. *SIAM Review*, 40(2):228–264, 1998.

[22] Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9), 1998.

[23] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems*, pages 161–168, 2007.

[24] Jean-Luc Bouchot, Simon Foucart, and Pawel Hitczenko. Hard thresholding pursuit algorithms: number of iterations. *Applied and Computational Harmonic Analysis*, 41(2):412–435, 2016.

[25] Petros Boufounos and Richard G. Baraniuk. 1-bit compressive sensing. In *Proceedings of the 42nd Annual Conference on Information Sciences and Systems*, pages 16–21, 2008.

[26] Samuel Burer and Renato D. C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

[27] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[28] T. Tony Cai and Lie Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, 57(7):4680–4688, 2011.

[29] Tony T. Cai, Lie Wang, and Guangwu Xu. New bounds for restricted isometry constants. *IEEE Transactions on Information Theory*, 56(9):4388–4394, 2010.

[30] Tony T. Cai and Anru Zhang. Sharp RIP bound for sparse signal and low-rank matrix recovery. *Applied and Computational Harmonic Analysis*, 35(1):74–93, 2013.

[31] Tony T. Cai and Wen-Xin Zhou. A max-norm constrained minimization approach to 1-bit matrix completion. *Journal of Machine Learning Research*, 14(1):3619–3647, 2013.

[32] Tony T. Cai and Wen-Xin Zhou. Matrix completion via max-norm constrained optimization. *Electronic Journal of Statistics*, 10(1):1493–1525, 2016.

[33] Emmanuel J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

[34] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11:1–11:37, 2011.

[35] Emmanuel J. Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[36] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

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

[38] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

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

[40] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[41] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[42] Yudong Chen. Incoherence-optimal matrix completion. *IEEE Transactions on Information Theory*, 61(5):2909–2923, 2015.

[43] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. *Journal of Machine Learning Research*, 15(1):2213–2238, 2014.

[44] Yudong Chen, Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Matrix completion with column manipulation: Near-optimal sample-robustness-rank tradeoffs. *IEEE Transactions on Information Theory*, 62(1):503–526, 2016.

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

[46] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[47] Mark A. Davenport, Yaniv Plan, Ewout van den Berg, and Mary Wootters. 1-bit matrix completion. *Information and Inference*, 3(3):189–223, 2014.

[48] Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*, pages 1646–1654, 2014.

[49] David L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

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

[51] David L. Donoho, Michael Elad, and Vladimir N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, 2006.

[52] David L. Donoho, Iain Johnstone, and Andrea Montanari. Accurate prediction of phase transitions in compressed sensing via a connection to minimax denoising. *IEEE Transactions on Information Theory*, 59(6):3396–3433, 2013.

[53] David L. Donoho and Jared Tanner. Precise undersampling theorems. *Proceedings of the IEEE*, 98(6):913–924, 2010.

[54] John C. Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009.

[55] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[56] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, 2009.

[57] Brian Eriksson, Laura Balzano, and Robert D. Nowak. High-rank matrix completion and subspace clustering with missing data. *CoRR*, abs/1112.5629, 2011.

[58] Maryam Fazel, Haitham Hindi, and Stephen P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, pages 4734–4739, 2001.

[59] Jiashi Feng, Huan Xu, and Shuicheng Yan. Online robust PCA via stochastic optimization. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 404–412, 2013.

[60] Simon Foucart. Hard thresholding pursuit: An algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

[61] Simon Foucart. Sparse recovery algorithms: Sufficient conditions in terms of restricted isometry constants. In *Approximation Theory XIII: San Antonio 2010*, pages 65–77. Springer, New York, NY, 2012.

[62] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser, 2013.

[63] Rina Foygel, Nathan Srebro, and Ruslan Salakhutdinov. Matrix reconstruction with the local max norm. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems*, pages 944–952, 2012.

[64] Rong Ge, Jason D. Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 2973–2981, 2016.

[65] Sivakant Gopi, Praneeth Netrapalli, Prateek Jain, and Aditya V. Nori. One-bit compressed sensing: Provable support and vector recovery. In *Proceedings of the 30th International Conference on Machine Learning*, pages 154–162, 2013.

[66] David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

[67] Ankit Gupta, Robert D. Nowak, and Benjamin Recht. Sample complexity for 1-bit compressed sensing and sparse classification. In *Proceedings of the IEEE International Symposium on Information Theory*, pages 1553–1557, 2010.

[68] Benjamin D. Haeffele, Eric Young, and René Vidal. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *Proceedings of the 31st International Conference on Machine Learning*, pages 2007–2015, 2014.

[69] Cho-Jui Hsieh and Peder A. Olsen. Nuclear norm minimization via active subspace selection. In *Proceedings of the 31st International Conference on Machine Learning*, pages 575–583, 2014.

[70] Laurent Jacques, Jason N. Laska, Petros T. Boufounos, and Richard G. Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Transactions on Information Theory*, 59(4):2082–2102, 2013.

[71] Martin Jaggi and Marek Sulovský. A simple algorithm for nuclear norm regularized problems. In *Proceedings of the 27th International Conference on Machine Learning*, pages 471–478, 2010.

[72] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th Annual Symposium on the Theory of Computing*, pages 665–674, 2013.

[73] Prateek Jain, Ambuj Tewari, and Inderjit S. Dhillon. Orthogonal matching pursuit with replacement. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems*, pages 1215–1223, 2011.

[74] Prateek Jain, Ambuj Tewari, and Inderjit S. Dhillon. Partial hard thresholding. *IEEE Transactions on Information Theory*, 63(5):3029–3038, 2017.

[75] Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional M-estimation. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*, pages 685–693, 2014.

[76] Ali Jalali and Nathan Srebro. Clustering using max-norm constrained optimization. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[77] Chi Jin, Sham M. Kakade, and Praneeth Netrapalli. Provable efficient online matrix completion via non-convex stochastic gradient descent. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 4520–4528, 2016.

[78] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 315–323, 2013.

[79] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[80] Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

[81] Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11:2057–2078, 2010.

[82] Olga Klopp. Noisy low-rank matrix completion with general sampling distribution. *Bernoulli*, 20(1):282–303, 2014.

[83] Karin Knudson, Rayan Saab, and Rachel Ward. One-bit compressive sensing with norm estimation. *IEEE Transactions on Information Theory*, 62(5):2748–2758, 2016.

[84] Jean Lafond. Low rank matrix completion with exponential family noise. In *Proceedings of the 28th Conference on Learning Theory*, pages 1224–1243, 2015.

[85] John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, 2009.

[86] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer-Verlag Berlin Heidelberg, 1991.

[87] Jason D. Lee, Ben Recht, Ruslan Salakhutdinov, Nathan Srebro, and Joel A. Tropp. Practical large-scale optimization for max-norm regularization. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems*, pages 1297–1305, 2010.

[88] Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Proceedings of the 29th Conference on Learning Theory*, pages 1246–1257, 2016.

[89] Ping Li, Cun-Hui Zhang, and Tong Zhang. Compressed counting meets compressed sensing. In *Proceedings of The 27th Conference on Learning Theory*, pages 1058–1077, 2014.

[90] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *CoRR*, abs/1009.5055, 2010.

[91] Guangcan Liu and Ping Li. Recovery of coherent data via low-rank dictionary pursuit. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*, pages 1206–1214, 2014.

[92] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[93] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th International Conference on Machine Learning*, pages 663–670, 2010.

[94] Po-Ling Loh and Martin J. Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *The Annals of Statistics*, 40(3):1637–1664, 2012.

[95] Po-Ling Loh and Martin J. Wainwright. Regularized m-estimators with nonconvexity: statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16:559–616, 2015.

[96] Po-Ling Loh and Martin J. Wainwright. Support recovery without incoherence: A case for nonconvex regularization. *The Annals of Statistics*, 45(6):2455–2482, 2017.

[97] Zongming Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013.

[98] Julien Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 2283–2291, 2013.

[99] Julien Mairal, Francis R. Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[100] Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, pages 2326–2366, 2006.

[101] Qun Mo. A sharp restricted isometry constant bound of orthogonal matching pursuit. *CoRR*, abs/1501.01708, 2015.

[102] Qun Mo and Yi Shen. A remark on the restricted isometry property in orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 58(6):3654–3656, 2012.

[103] Deanna Needell and Joel A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[104] Deanna Needell and Roman Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):310–316, 2010.

[105] Sahand Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, pages 1348–1356, 2009.

[106] Sahand Negahban and Martin J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13:1665–1697, 2012.

[107] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Springer US, 2004.

[108] Behnam Neyshabur, Yury Makarychev, and Nathan Srebro. Clustering, hamming embedding, generalized LSH and the max norm. In *Proceedings of the 25th International Conference on Algorithmic Learning Theory*, pages 306–320, 2014.

[109] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 15th Annual Conference on Neural Information Processing Systems*, pages 849–856, 2001.

[110] Nam H. Nguyen, Deanna Needell, and Tina Woolf. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *CoRR*, abs/1407.0088, 2014.

[111] Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23):3311–3325, 1997.

[112] Francesco Orabona, Andreas Argyriou, and Nathan Srebro. PRISMA: PRoximal Iterative SMoothing Algorithm. *CoRR*, abs/1206.2372, 2012.

[113] Stanley Osher, Feng Ruan, Jiechao Xiong, Yuan Yao, and Wotao Yin. Sparse recovery via differential inclusions. *Applied and Computational Harmonic Analysis*, 41(2):436–469, 2016.

[114] Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

[115] Yagyensh C. Pati, Ramin Rezaiifar, and Perinkulam S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Conference Record of The 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44, 1993.

[116] Yaniv Plan and Roman Vershynin. Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach. *IEEE Transactions on Information Theory*, 59(1):482–494, 2013.

[117] Yaniv Plan, Roman Vershynin, and Elena Yudovina. High-dimensional estimation with geometric constraints. *Information and Inference*, 6(1):1–40, 2017.

[118] Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Restricted eigenvalue properties for correlated gaussian designs. *Journal of Machine Learning Research*, 11:2241–2259, 2010.

[119] Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Transactions on Information Theory*, 57(10):6976–6994, 2011.

[120] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[121] Jason D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 713–719, 2005.

[122] Peter Richtárik and Martin Takác. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[123] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, 1958.

[124] Nicolas Le Roux, Mark W. Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems*, pages 2672–2680, 2012.

[125] Mark Rudelson and Shuheng Zhou. Reconstruction from anisotropic random measurements. *IEEE Transactions on Information Theory*, 59(6):3434–3447, 2013.

[126] Ruslan Salakhutdinov and Nathan Srebro. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems*, pages 2056–2064, 2010.

[127] Yoav Seginer. The expected norm of random matrices. *Combinatorics, Probability & Computing*, 9(2):149–166, 2000.

[128] Jie Shen and Ping Li. Learning structured low-rank representation via matrix factorization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 500–509, 2016.

[129] Jie Shen and Ping Li. On the iteration complexity of support recovery via hard thresholding pursuit. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3115–3124, 2017.

[130] Jie Shen and Ping Li. Partial hard thresholding: Towards a principled analysis of support recovery. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 3127–3137, 2017.

[131] Jie Shen and Ping Li. A tight bound of hard thresholding. *Journal of Machine Learning Research*, 18(208):1–42, 2018.

[132] Jie Shen, Ping Li, and Huan Xu. Online low-rank subspace clustering by basis dictionary pursuit. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 622–631, 2016.

[133] Jie Shen, Huan Xu, and Ping Li. Online optimization for max-norm regularization. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*, pages 1718–1726, 2014.

[134] Jie Shen, Huan Xu, and Ping Li. Online optimization for max-norm regularization. *Machine Learning*, 106(3):419–457, 2017.

[135] Martin Slawski and Ping Li. Linear signal recovery from b-bit-quantized linear measurements: precise analysis of the trade-off between bit depth and number of measurements. *CoRR*, abs/1607.02649, 2016.

[136] Mahdi Soltanolkotabi and Emmanuel J. Candès. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40:2195–2238, 2012.

[137] Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J. Candès. Robust subspace clustering. *The Annals of Statistics*, 42(2):669–699, 2014.

[138] Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakkola. Maximum-margin matrix factorization. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems*, pages 1329–1336, 2004.

[139] Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 545–560, 2005.

[140] Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[141] Joel A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[142] Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[143] Joel A. Tropp and Stephen J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

[144] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge University Press, 2000.

[145] René Vidal. A tutorial on subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2010.

[146] Martin J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009.

[147] Huahua Wang and Arindam Banerjee. Randomized block coordinate descent for online and stochastic optimization. *CoRR*, abs/1407.0107, 2014.

[148] Jian Wang, Suhyuk Kwon, Ping Li, and Byonghyo Shim. Recovery of sparse signals via generalized orthogonal matching pursuit: A new analysis. *IEEE Transactions on Signal Processing*, 64(4):1076–1089, 2016.

[149] Jian Wang and Byonghyo Shim. On the recovery limit of sparse signals using orthogonal matching pursuit. *IEEE Transactions on Signal Processing*, 60(9):4973–4976, 2012.

[150] Yu-Xiang Wang, Huan Xu, and Chenlei Leng. Provable subspace clustering: When LRR meets SSC. In *Proceedings of 27th Annual Conference on Neural Information Processing Systems*, pages 64–72, 2013.

[151] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

[152] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[153] Huan Xu, Constantine Caramanis, and Shie Mannor. Principal component analysis with contaminated data: The high dimensional case. In *Proceedings of the 23rd Conference on Learning Theory*, pages 490–502, 2010.

[154] Huan Xu, Constantine Caramanis, and Shie Mannor. Outlier-robust PCA: the high-dimensional case. *IEEE Transactions on Information Theory*, 59(1):546–572, 2013.

[155] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, 2012.

[156] Songbai Yan and Chicheng Zhang. Revisiting perceptron: Efficient and label-optimal learning of halfspaces. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 1056–1066, 2017.

[157] Ming Yuan and Yi Lin. On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):143–161, 2007.

[158] Xiao-Tong Yuan, Ping Li, and Tong Zhang. Exact recovery of hard thresholding pursuit. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 3558–3566, 2016.

[159] Xiao-Tong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit. *Journal of Machine Learning Research*, 18(166):1–43, 2018.

[160] Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14(1):899–925, 2013.

[161] Tong Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568, 2009.

[162] Tong Zhang. Some sharp performance bounds for least squares regression with L_1 regularization. *The Annals of Statistics*, 37(5A):2109–2144, 2009.

[163] Tong Zhang. Sparse recovery with orthogonal matching pursuit under RIP. *IEEE Transactions on Information Theory*, 57(9):6215–6221, 2011.

[164] Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

[165] Zihan Zhou, Xiaodong Li, John Wright, Emmanuel J. Candès, and Yi Ma. Stable principal component pursuit. In *Proceedings of the 2010 IEEE International Symposium on Information Theory*, pages 1518–1522, 2010.